Autonomous recommender system architecture for virtual learning environments

Julián Monsalve-Pulido
GIDITIC, Universidad EAFIT, Medellín, Colombia
Jose Aguilar
GIDITIC, Universidad EAFIT, Medellín, Colombia and CEMISID, Universidad de Los Andes, Mérida, Venezuela, and
Edwin Montoya and Camilo Salazar
GIDITIC, Universidad EAFIT, Medellín, Colombia

Received 10 December 2019
Revised 5 February 2020
Accepted 2 March 2020

Abstract

This article proposes an architecture of an intelligent and autonomous recommendation system to be applied to any virtual learning environment, with the objective of efficiently recommending digital resources. The paper presents the architectural details of the intelligent and autonomous dimensions of the recommendation system. The paper describes a hybrid recommendation model that orchestrates and manages the available information and the specific recommendation needs, in order to determine the recommendation algorithms to be used. The hybrid model allows the integration of the approaches based on collaborative filter, content or knowledge. In the architecture, information is extracted from four sources: the context, the students, the course and the digital resources, identifying variables, such as individual learning styles, socioeconomic information, connection characteristics, location, etc. Tests were carried out for the creation of an academic course, in order to analyse the intelligent and autonomous capabilities of the architecture.

Keywords Autonomous computing, Context-aware, Recommendation systems

Paper type Original Article

This work has been supported by the project 64366: “Contenidos de aprendizaje inteligentes a través del uso de herramientas de Big Data, Analítica Avanzada e IA” – Ministry of Science – Government of Antioquia – Republic of Colombia.

Declaration of Competing Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Publishers note: The publisher wishes to inform readers that the article “Autonomous recommender system architecture for virtual learning environments” was originally published by the previous publisher of *Applied Computing and Informatics* and the pagination of this article has been subsequently changed. There has been no change to the content of the article. This change was necessary for the journal to transition from the previous publisher to the new one. The publisher sincerely apologises for any inconvenience caused. To access and cite this article, please use Monsalve-Pulido, J., Aguilar, J., Montoya, W., Salazar., C. (2020), “Autonomous recommender system architecture for virtual learning environments”, *Applied Computing and Informatics*. Vol. ahead-of-print No. ahead-of-print. https://10.1016/j.aci.2020.03.001. The original publication date for this paper was 07/03/2020.

Applied Computing and Informatics
Vol. 20 No. 1/2, 2024
pp. 69-88
Emerald Publishing Limited
e-ISSN: 2210-8327
p-ISSN: 2634-1964
DOI 10.1016/j.aci.2020.03.001

1. Introduction

In general, recommendation systems are based on different domains, like the cognitive sciences, information retrieval, consumer modeling, among others [1]. Normally, the problem to be solved is to provide users with personalized suggestions (recommendations) about certain types of items. Thus, the recommendation system carries out a ranking of the items to be proposed to the users [2].

The recommendation systems have been used in several areas, with a great importance in recent years in the electronic commerce domain [2–4], with implementations of recommendation systems for the sale of products or services on digital platforms [5]. Also, the recommendation systems have appeared in education through the Learning Management Systems (LMS), in order to improve the online learning-teaching processes [6–9]. A LMS is a web-based software to manage academic courses, where tutors manage the teaching–learning process through an integration of students, digital resources and contents, according to an instructional design [6]. The recommendation systems can improve the distribution of academic material to the students, through content adapted to their needs according to their learning styles [10,11]. Recommendation systems in the educational domain are not directly transferable to conventional systems, specifically due to the student's cognitive state and learning contents, because they are very specific to this domain [12].

To improve the teaching processes in virtual learning environments (VLE), it is necessary recommendation systems that deeply analyze the student's variables, such as its learning style, preferences, its social networks, among others. Also, a recommendation system must permanently recommend digital contents, updates the information about the current academic course, etc. Based on these ideas, this paper proposes an intelligent autonomous architecture for a recommendation system to be used in a VLE, for an efficient recommendation of digital resources. The architecture presents a hybrid recommendation model that orchestrates and manages the inputs, and specific recommendation needs. The proposed architecture extracts information about the context (such as the location, the connectivity information), the students (such as their learning styles, socioeconomic information, etc.), the courses (such as their instructional designs), and the digital resources (which can be about the learning objects, patents, journals, among others), in order to give an intelligent and autonomous recommendation.

The document begins by presenting the previous works related to the investigation. Then, it describes our autonomous and intelligent recommendation architecture. In particular, this section defines its metamodels, and its intelligent and autonomous dimensions. The next section analyses the behavior of the architecture, in an example of creation of an academic course. The next section presents a comparison with previous researches, to identify the relevance of the proposed architecture. Finally, some conclusions and future work are proposed in the last section.

2. Related works

In the literature, there are several propositions of architectures of recommendation systems for different areas. In the academic area, it aims to seek the customization of the teaching–learning process, as the proposal of [13], which describes a general architecture for a hybrid recommendation system, through an effort-based learning to mature quality over time. In the architecture, they take into account the context variables (content, user and time) for the application of the recommendation algorithms. Another hybrid architecture is proposed in [7], which combines recommendations based on improved cases (driven by a quality model) with user comments (collaborative), to recommend open courses and educational resources. The architecture proposed by [8] is based on multiple hierarchical intelligent agents, in order to offer a range of main functions, including static and dynamic user modeling, generation and adjustment of learning plans, personalized recommendation, and real-time learning evaluation progress, among other things. The construction of knowledge based on collaboration is important tasks for the teaching–learning processes, [14] proposes an architecture of a multi-agent client–server application to perform recommendation of activities with different characteristics. The architecture analyzes the set of basic functionalities and operations that give access to these functionalities, for the exchange of knowledge in virtual communities. In this way, the personal agents in Android exploit them to execute their tasks while achieving their individual and collective goals. The architecture has been used in the context of Smart Cities, fulfilling the objectives of decentralization for the community management, the personalized automatic management, and the discovery of communities.

The definition of intelligent recommendation systems is a challenge. The work [15] proposes a general framework for an intelligent recommender system that extends the concept of a knowledge-based recommender system. The intelligent recommender system learns, discovers new information, infers preferences, among other things. The components of the intelligent recommender system are the following: a module to model the knowledge based on different representation paradigms, a module with learning methods, and finally, a module with the reasoning mechanisms. Additionally, it proposes five knowledge models about the users, items, domain, context and criticisms. The recommender system exploits this knowledge. They instance the intelligent recommender system using Fuzzy Cognitive Maps (FCMs) [16], and test its performance.

In e-commerce, the architectures of recommendation systems have been proposed in recent years, such as the case of [3], where they apply artificial intelligence technologies and big data to design the architecture of a C2C s-commerce recommendation system, in order to generate efficient recommendations for capability enhancement in online business arrangements. Also, in [5] describes a set of web-based data mining tools for identifying patterns of navigation, in order to support a personalized product recommendation system for online sales. It presents an architecture supported by a neural network that has been trained off-line with user actions, for the recommendation of sets of individual navigation options by products.

Anticipation of customer interest is an important value in e-commerce processes. In this context, [17] presents an intelligent recommendation system architecture, which uses an autonomous agent to track user web browsing behaviors, and tries to anticipate current interests, building user profiles dynamically. Also, consumer privacy is important for e-commerce processes. Thus, [4] presents PrivBox, a decentralized reputation system that preserves privacy and calculates the reputation of retailers or service providers, taking advantage of the user comments. The PrivBox system uses primitives from a homomorphic cryptographic system and a non-interactive zero knowledge test, to achieve the objectives of preserving privacy.

In other areas, architectures of recommendation systems have been presented. For example, in the case of music and multimedia, like [18] that presents an architecture of a music recommendation system, which has been defined under a general framework that uses a ubiquitous computer middleware for the extraction of the context since several sensors, by detecting, combining, inferring and learning the context information, to provide diverse user services in changing environments. In the case of multimedia, [19] proposes a peer-oriented multimedia content recommendation system based on collaborative filtering, to obtain peer search efficiency. The architecture presents a quality validation of the recommendations, and is faster than a centralized reference system.

In this section, we have presented several works that propose architectures of recommendation systems, but only two in the educational context. These architectures consider the context, but they are not autonomous or intelligent. Also, there are propositions of architectures of intelligent recommendation systems, but they are not autonomous. They consider the utilization of intelligent techniques in other to process the information for the recommendation. In general, there are not propositions of architectures of recommendation systems that integrate the intelligent and autonomous aspects, in the educational context, in order to proactively work during a learning process in a VLE. The Section 6 analyses these aspects between different works.

3. Architecture of the intelligent autonomous recommender system

In this section, we propose a new architecture of a Recommender System, based on two principles. The first is the principles of autonomous computing, which is based on a model presented by [20], where it is proposed an approach of self-managing computing based on the human nervous system. The second is based on the principles of an intelligent Recommender System defined in [15], which establishes the components (knowledge representation paradigm, learning methods, and reasoning mechanisms), and the knowledge models (users, items, domain, context and criticisms) that must be considered for an intelligent recommendation.

3.1 General architecture

The Figure 1 presents the general architecture proposed of an Autonomous Recommendation System (ARS) for VLE, with the aim of integrating actors and contents in the teaching–learning process, in order to offer recommendations in an autonomous and intelligent way for students and teachers.

The architecture of the autonomous recommendation system considers the 3 main phases during a teaching–learning process, to meet the recommendation objectives. In the first

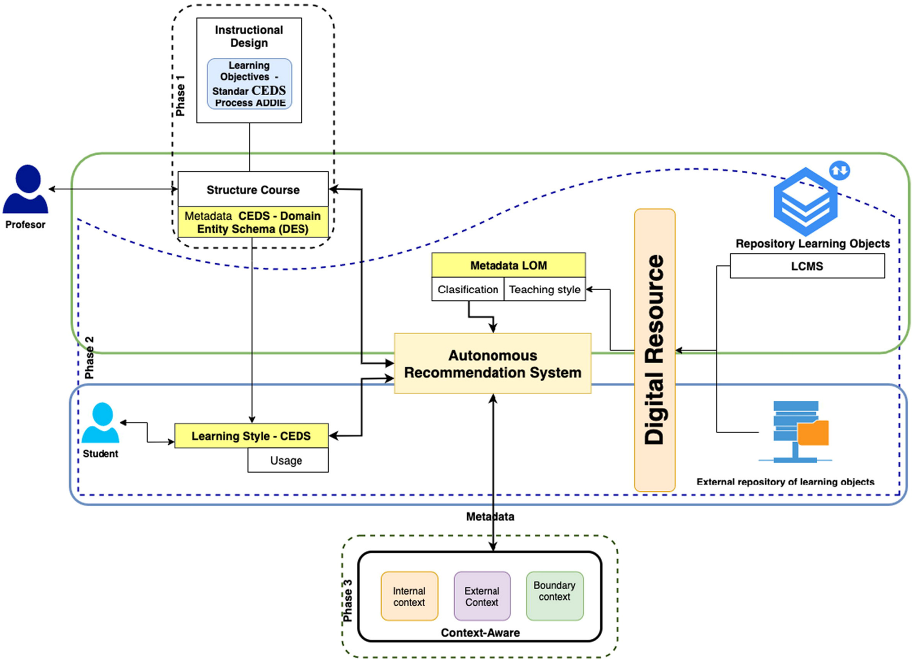

Figure 1.
Architecture of the Autonomous Recommendation System.

phase, the process of creating of academic courses is carried out; in the second phase, the usability of the digital resources by the student must be guaranteed; and finally, during the third phase, the extraction of context variables is necessary for a complete autonomous recommendation system that consider the environment. These phases are described in detail below.

- *Phase 1:* In this phase is carried out the creation of the academic course, through a structure based on the teaching objectives proposed in the academic program. In the creation of the course, an instructional design is necessary that identify the pedagogical aspects to be considered by the ARS. Thus, for the description of the instructional design of a course, the ARS uses the ADDIE (Analysis-Design-Development-Implementation-Evaluation) model standard [21], and for the learning objectives, the Common Education Data Standard (CEDS) [22].
- *Phase 2:* In addition to the information about a course, it is necessary the information of the digital resources and students. The digital resources will be extracted since two main sources, the first from the Learning Content Management Systems (LCMSs), and the second from external open learning resource repositories. Digital resources are characterized using the Learning Object Metadata (LOM) standard [23], in order to structure the specific information of the resource. Learning resource repositories contain several types of sources of academic information (scientific articles, patents, virtual learning objects, books, videos, among others). The student is a fundamental part of the ARS. The profile of a student is composed by its learning style and its personal and academic information. The learning style information can be determined of different forms, via the analysis of the VLE logs, of the behaviour in the social networks, of the academic record, among other ways. This information is stored using the CEDS [22] standard, and sent to the ARS, so that it can use it to determine the recommendations to the students.
- *Phase 3:* The context is important for any learning customization. The definition of context focuses on identifying any information that can be used to characterize any entity in the environment, considering that an entity can be a person, a place or an object [24]. In the ARS, this information allows that it is context awareness. Particularly, the context information is extracted from the student environment, and it is stored in a general context metadata model. According to this model, the information extracted is stored into three parts [25,26]: internal context, external context and Edge Context. This information is used to feed the ARS for the delivery of better adapted recommendations to the student. This model will be explained in the next section.

3.2 Metamodels

In this section, the description of the metamodels that are part of the architecture of the autonomous recommendation system is presented. Each metamodel defines the metadata of the different sources of knowledge (data dimensions) used by our recommendation system: students, courses, digital resources and context.

3.2.1 Student metamodel. About the students, several information are very important. For example, the identification of the learning style is important for the autonomous recommendation system, in order to define the academic contents to the students. It is proposed to use the learning style model proposed by Felder-Silverman [27], where the identification of the types/styles of learning are defined according to several dimensions: global vs sequential, visual vs verbal, sensitive vs intuitive, and inductive vs deductive. With the identification of the student's learning style, composed by the combinations of these dimensions, the autonomous recommendation system can search the digital resources that adapt to the specific needs of the student, such as videos, books, presentations, articles, slides, graphics, among others. The extended student metadata that is used includes the personal and social information of the students, based on the Common Education Data Standards (CEDS) [22]. This metamodel is integrated with the rest of data dimensions for the optimization of the recommendation process (see Figure 2).

The Table 1 presents the detail of the student metadata proposal based on the CDES standard, extended with variables about the learning style based on the Felder-Silverman model. The metadata is divided into seven information categories that are integrated into the proposed recommendation system architecture. The categories are: Identification, Contact, Demography, Academic history, inabilities, learning style and additional fields.

3.2.2 Course metamodel. The data about the academic courses are based on two standards, the first also related to the students is the Common Education Data (CEDS) [22], and the other is the ADDIE model [21]. The ADDIE model is used for the creation of the instructional designs of the academic courses. The instructional design can be defined as a systematic, planned and structured process to support the learning processes adapted to the student's learning needs. The ADDIE model uses five phases of the instructional design of a course: it begins with the analysis of the context, where are performed the needs assessment, the identification of the problem to be solved and the definition of the specific tasks required. The second phase is the design, where it is specified the objectives, themes and resources of the course. In the third phase, the development of the course is carried out through a production process of the educational material. The fourth phase is the implementation of the

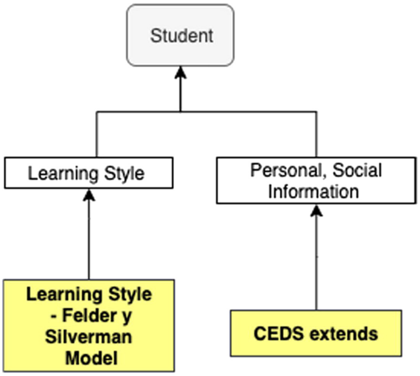
Student
Learning Style
Personal, Social Information
Learning Style - Felder y Silverman Model
CEDS extends

Figure 2.
Information extracted from the students.

Table 1.
Description of the student metadata.

Identity	First Name – Middle Name – Student ID– Surname – Student identification system
Contact	Residence address – Mobile phone number – Email
Demography	Birthdate – Sex
Academic historial	Number of credits approved – Number of credits failed
Inability	Initial Status Assessment – Disability indicator – Type of main disability – Type of disability determination source – Educational environment for the school era – IDEA
Learning style	Active/reflective – Sensory/intuitive – Visual/verbal – Sequential/global
Adicional files	Language – Ethnicity – Typical preferred age range – Preferred difficulty – Contents Viewed – Active courses

course, where it is applied in a pilot context to analyze the evaluations and the student's comments in a controlled manner. Finally, the last phase is the evaluation of the course, which is made by students and teachers, depending on the fulfillment of the teaching–learning objectives.

The proposed metadata for the course is based on the CEDS standard, which is composed of 20 description fields. Some of them are: 1. Course identifier; 2. Course code; 3. Course name; 4. Course description: it details a course; 5. Credit units of the course; 6. Type of credit unit: Unit of credit measurement in the institution; 7. Course level feature: it defines the difficulty of the course for students; 8. Main academic course: Defines whether or not it is a main academic course. 9. Language of instruction: language in which the course is taught; 10. Keywords; 11. Duration of the course; 12. Type of interactivity expected in the course; 13. Typical age range of students; 14. Cognitive process: if the course has an expected cognitive process (this field is based on the verbs of the taxonomy of Bloom [28]; 15. Type of educational resource expected; and 16. Expected semantic density: For example: a course could expect content to be of a low semantic density if it is an introductory course or of low difficulty.

3.2.3 Digital resource metamodel. For the analysis and classification of digital resources, it is necessary to use a metadata standard that adapts to any digital material, regardless of the language or structure that may contain, with enough information to be classified and reused over time. For the ARS, it is defined a Digital Resource Metamodel based on the Learning Object Metadata (LOM) IEEE standard [23]. It defines nine information categories, some of them are: 1. General information about the learning object, 2. Life cycle of the learning object, 3.Technique features of the learning object, 4. Pedagogical features of the learning object, 5. Utilisation conditions of the learning object, and 6. Relationships with other learning objects. This LOM metadata is extended to describe different types of digital materials, such as learning objects, scientific articles, books, patents, among others.

The structure of extended metadata is created based on the LOM standard, where a field is added for the characterization of the different types of digital resources. The extended metadata follows the LOM standard in its original structure with its nine categories (General, Life cycle, metadata, technical requirements, pedagogical characteristics, usage rights, relationships, annotations and classification). In addition, it has an extra category (category ten) for the description of the specific digital resource (an extension to the LOM standard), called additional fields, where variables, such as the digital resource type, the digital resource source, the digital resource descriptors, and the feature extraction method, among others, are incorporated.

3.2.4 Context metamodel. The context is very important for the learning customization. In our work, the definition of the context focuses on identifying any information that can be used to characterize any entity, taking into account that an entity can be a person, a place or any object that can be integrated into the learning process [24]. The Figure 3 presents the context model for our autonomous recommendation system, which is composed of three levels: an internal level, an external level, and finally, a boundary level.

The external level describes the context from the point of view of the learning process (information about the students and courses, based on the metamodel of these data dimensions). The internal level describes the different devices, and other specific features, in the environment. Finally, the boundary level is defined by the digital resources and other domain services outside of the environment.

With respect to the internal level, the information from the context that defines its metadata is described in detail in the Table 2. There are five groups of variables, which are important for the orchestration of the autonomous recommendation system. The first group defines the hardware devices, and specifically, the special features used in the interaction with the learning environment. Similarly, the second group defines the main features of the software devices. The third group gives information about the environment, and the last groups contain information about connectivity capabilities of the devices and information about the movement in the environment.

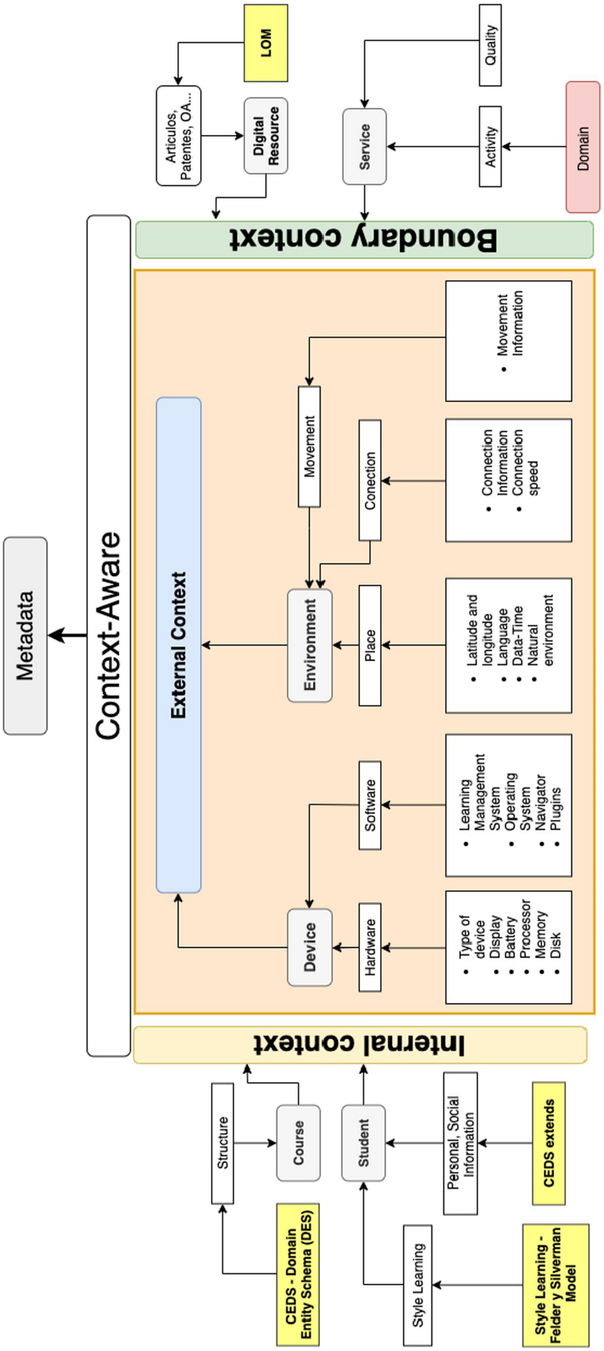

Figure 3.
Context in the autonomous recommender system.

	Device – Hardware	
Context variables	Description	Datatype
Type of device	Identify the type of device (Smartphone, Desktop, Wearable, Laptop, others.)	String
Display	Device screen size, and resolution	String
Battery	Device Battery percentage	Float
Processor	Device Processor features	String
Memory	Device memory capacity	Float
Disk	Device Free storage capacity	Float
	Device – Software	
Learning Management System	LMS name, version, type of access (installation or access by browser).	String
Operating System	Operating system type, version and update status	String
Navigator	Installed browsers, versions and update status	String
Plugins	Plugins installed on the device, versions and update status.	String
	Environment – Place	
Latitude and longitude	Location information when accessing the educational resource, by means of longitude and latitude.	Float – Float
Language	Extraction of the language of use	String
Data-time	Date and time of access to the educational resource	DateTime
Natural environment	Climate, Temperature and Light	String
	Device – Connection	
Connection information	Type of connection (Wifi, 4G, GPRS, others).	String
Connection speed	Internet speed up and down that the user has when accessing the academic resource.	Float
	Device – Movement	
Movement Information	At the time of using the academic material, the user is (walking, traveling in a means of transport or is detained in one place, others.)	String

Table 2. Description of variables extracted from context.

3.3 Intelligent dimension

Figure 4 shows the architecture of the intelligent engine for the recommendation system, based on the components that an intelligent recommender system must have: a knowledge representation paradigm, learning methods, and reasoning mechanisms. Our ARS is composed of three layers. In the acquisition layer is where the data from different sources of information are obtained, the second layer is where the information is stored in a knowledge database, and finally, the reasoning and verification layer uses this information for the recommendations. Each of the three layers of the proposed intelligent engine is described in detail below.

- *Acquisition layer*: In this phase, the information is extracted from different sources of information in two ways. The first is around the educational process, which consider information about academic courses and digital resources (scientific articles, academic books, learning objects, patents, among others). The information is extracted in an organized way, by means of the metadata defined in the previous section. The second extraction is about the student and the general context information. Particularly, the learning styles and the general contextual information are extracted permanently according to the features defined in the metadata. The information extracted of the students has as sources the logs of connection, the VLE, etc. On the other hand, all

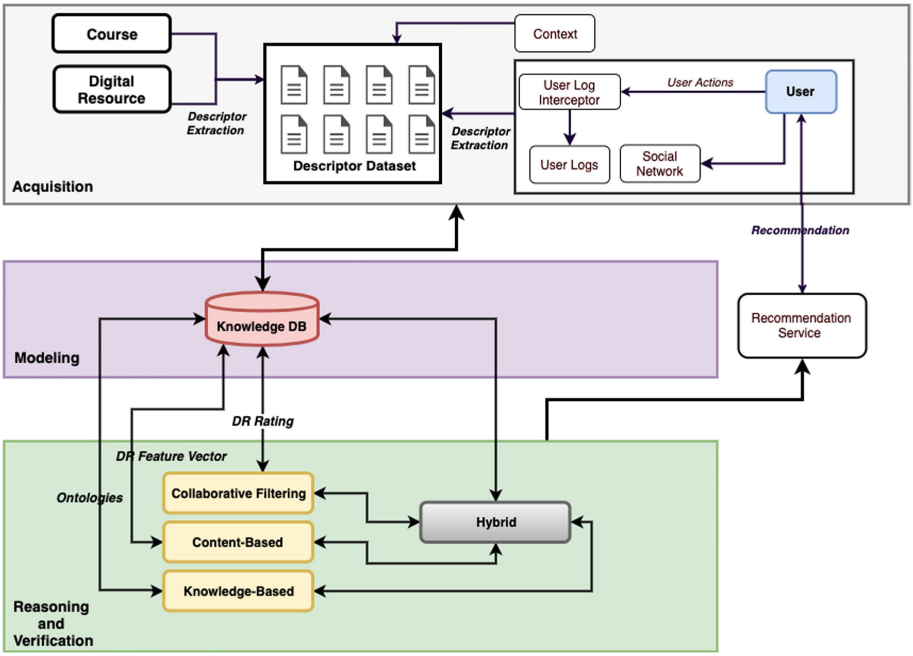

Figure 4.
Architecture of the Intelligent Dimension.

social media records are also taken into account for the extraction of features, where the information of comments, images, and in some cases affective states, are important for the identification of the student preferences. Particularly, in this layer must be used different learning methods for the extraction of features and information, based on supervised, unsupervised, reinforcement or hybrid approaches. For that, everything mining tasks will be required to discover useful information for the recommendations. For example, data mining tasks to analyse the academic datasets or the VLE logs, or semantic mining tasks for the digital resource content analysis, or graph mining tasks to study the social networks, among others.

- *Modeling*: In this layer, all the information of the recommendation system is stored through a knowledge database. The structure of the knowledge database is based on the metadata of the digital resources, academic courses, students and context. This database is feed using the learning methods defined in the acquisition layer. Also, the verification and reasoning layer stores the results of the recommendation process, like the ratings of the collaborative filter algorithms, the function-based algorithm vectors of content, the ontologies of the knowledge-based system, and the recommendation vectors of the hybrid system. In general, it can use different knowledge representation paradigms, for example, ontologies to represent concepts, relationships, facts and axioms in a domain; or graphs to model the relations among digital resources and students, etc.

- *Reasoning and Verification*: In this layer, the system performs the reasoning process for the recommendation. It aims to efficiently recommend academic content to students, according to their needs and interests [29,9]. The information used is stored in the knowledge database, which has been extracted by the acquisition layer, to build an effective teaching–learning process.

> In this layer, different reasoning mechanisms can be used, according to the data models defined in the modelling layer. For example, can be deductive, inductive or abductive mechanisms. Also, can be used hybrid recommendation techniques, combining collaborative filtering, content-based and knowledge-based approaches. The above, with the aim of exploiting the advantages of each technique, in order to recommend a more efficient content [30,31].

Specifically, the Figure 5 describes the hybrid recommendation process of our ARS. the objective is to create an orchestration algorithm that manages as optimally as possible, depending on inputs and specific recommendation needs, the combination strategies of the types of recommender systems. In the monolithic hybrid design, all algorithms are integrated (knowledge-based, collaborative filter, and content-based), so that the input data of one would be the specific data of another, according to the used combination strategy. The combination strategies can be: sequentialization, parallelization and integration, or selection of one of them. The sequentialization or parallelization can be with two or three of the types of recommender systems.

As has been explained before, the goal of the combination strategies is to exploit the advantages of each type of recommender system. The purpose of collaborative filtering is to recommend content based on the qualification of the students and other users, who have previously interacted with the content. The recommendation is made based on the similarity between students/users. This similarity is calculated using the historical record of the students, their preferences, etc. [32]. Thus, in the collaborative filtering, the quality of an academic content is based on the qualification of the students/users.

In the content-based recommendation, academic course, digital resources and student profiles contain independent information that describes the nature of each one. The goal is to learn to recommend, by using the similarity of the information contents of each one [33]. It searches to match the contents of the student profile attributes (preferences, learning styles and interests), with the contents of the digital resource attributes (academic articles, patents, books, learning objects, patents, others) and of the academic courses, in order to deliver recommendations to the students. One of the advantages of the content-based is the user independence, where the student profile can be constructed with his learning style, interaction logs, without relying on other users.

Finally, in the based on knowledge, the recommendations are based on the information stored in an ontology that describes the student (preferences, learning styles) context, educational material and courses. The above is done through a reasoning process in which are used axioms that establish the specific requirements or needs of the students. In knowledge-based can be used case-based and constraint models, to analysis the inconsistency of the information. Particularly, the case model focuses on the recovery of cases similar to a specific problem, to be used as the solution to the problem; and in the restriction model they explore the knowledge bases of predefined recommendations, where

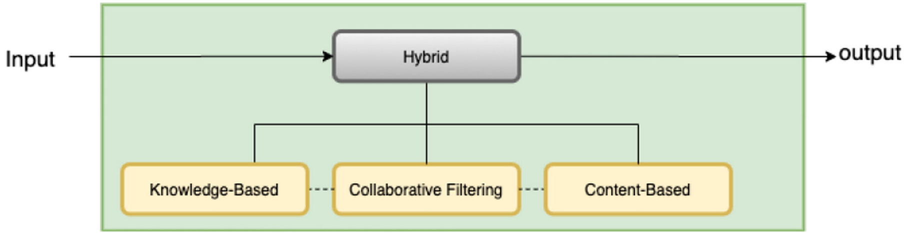

Figure 5.
Hybrid System Design.

explicit rules are stored about the relationship between the user requirements and the characteristics of the items [34].

3.4 Autonomous dimension

The autonomic architecture of the recommendation system guarantees self-management and adaptability to any context without human intervention. The Figure 6 shows the model proposed based on the MAPE-K loop model (Monitor-Analyze-Plan-Execute plus Knowledge), which aims to recommend digital academic contents to students depending on their needs. It is composed of the next components:

- *Monitor:* This component includes the extraction methods of the properties/ information of the digital resources, courses, context, social networks and student logs, among other sources of information. In the monitor component, the verification of the quality of the features/information is carried out by means of metrics. Also, all the data pre-processing mechanisms (cleaning, atypical data detection, etc.) are considered by this component.

- *Analysis – Planning:* In this component, the information extracted from the monitoring component is processed. This information mainly is composed of a dataset of descriptors/features extracted from each of the sources. This component exploits our hybrid recommendation filter, which consists of collaborative, content-based and knowledge-based filters, in order to determine the correct combination strategy to apply in order to guarantee a high quality process of recommendation of contents. Additionally, this component can estimate future recommendations. The results, like in the case of the monitoring component, are stored in the knowledge database.

- *Planning – Execution:* This component aims of deploying the recommendations (digital resources) according to user needs (learning style and context), the structure of the course and the VLE. It analyses the best characteristics to project the recommendation, which can include the adaptation of the digital resources to the context. Thus, it generates the commands to execute the recommendations through automatic applications available at runtime on any virtual learning platform.

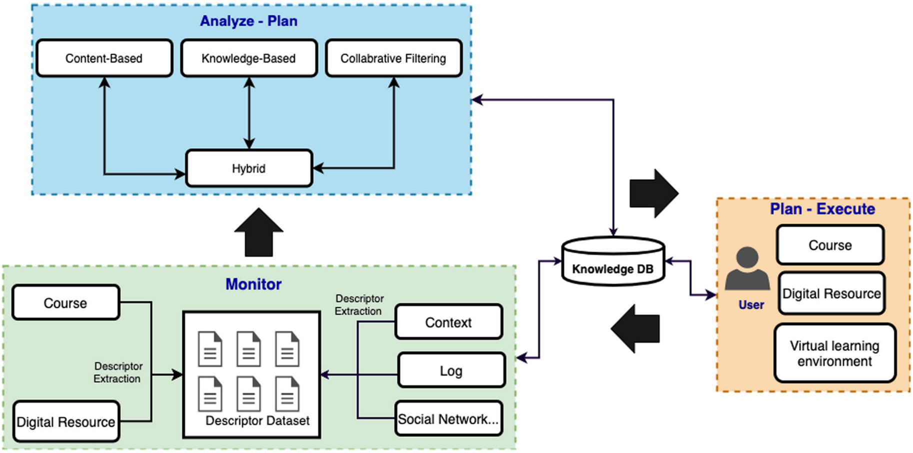

Figure 6.
Architecture of the Autonomous Dimension.

4. Analysis of the behaviour of the ARS

4.1 General context: creation of academic courses

In traditional VLEs (see Figure 7), academic courses are created by teachers depending on the learning objectives. The teacher has the autonomy to create the structure of the course according to the institutional pedagogical guidelines. Digital resources are assigned to the course according to the teacher's selection, through searches by subject or area of knowledge made to institutional repositories or external sources. When the course is created in its entirety, it is offered to the students to interact with the VLE, and to perform all the programmed tasks of learning and evaluation, using the pre-selected digital resources.

With the application of the proposed recommendation architecture, the process of allocating of digital resources changes significantly (see Figure 8). The teacher creates the academic course, according to the institutional pedagogical structure, using the metadata of the Course Metamodel (phase 1). In addition, the teacher defines a general profile of the students of this course (ages, cognitive level, etc.) (phase 2). With this information, the recommendation system searches digital resources in digital repositories

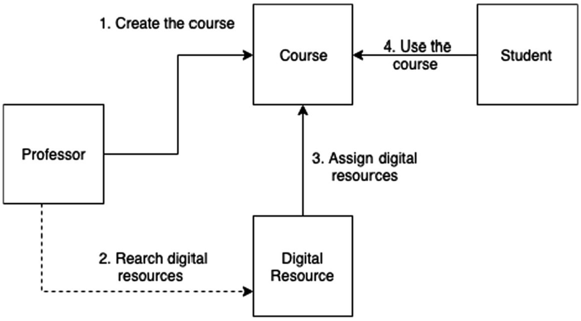

Figure 7.
Traditional way of definition of digital resources for a course.

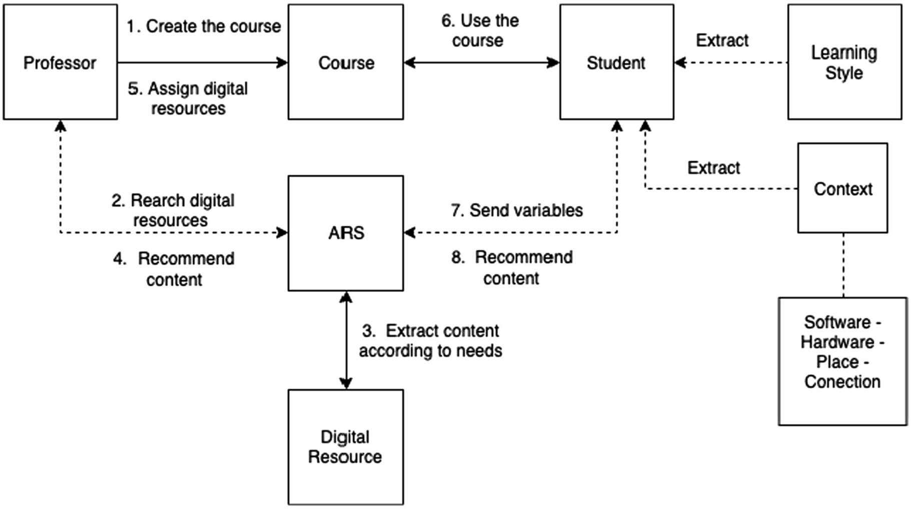

Figure 8.
New way for the management of the digital resources with the proposed recommendation architecture.

through the next criteria (phase 2): contents that fit the learning objectives, contents with good grades of the students, contents that fit the domain of knowledge of the course, contents that adapt to the preliminary profiles of the students, the context characteristics (phase 3), etc. Now, the teacher completes the course by selecting the digital resources since the recommendation system, according to its pedagogical experience in the area, ensuring the fulfillment of the teaching–learning objectives. At the moment that the academic course is in an active state, the students begin with the interaction with the VLE using the digital contents proposed since the creation of the course. But the recommendation system is also executed by the VLE, which allows the recommendation of new digital contents adjusted to the individual learning styles and context where the student is interacting (phase 3).

4.2 Behaviour of our recommendation system

In this section, it is described in detail the recommendation process in the previous context, using our architecture. Particularly, it is supposed the creation of a new academic course: Introduction to Programming. The academic objective is to teach algorithms using a programming language. The general topics of the course are: Introduction to the programming logic; introduction to the programming languages; what are the Data types, operators and logical connectors; introduction to the control structures (selection, repetitive, etc.); definition of functions, procedures, and modular programming; and introduction to the File Management.

The Figure 9 shows the behavior of the intelligent and autonomous architecture of the recommendation system, which is described below:

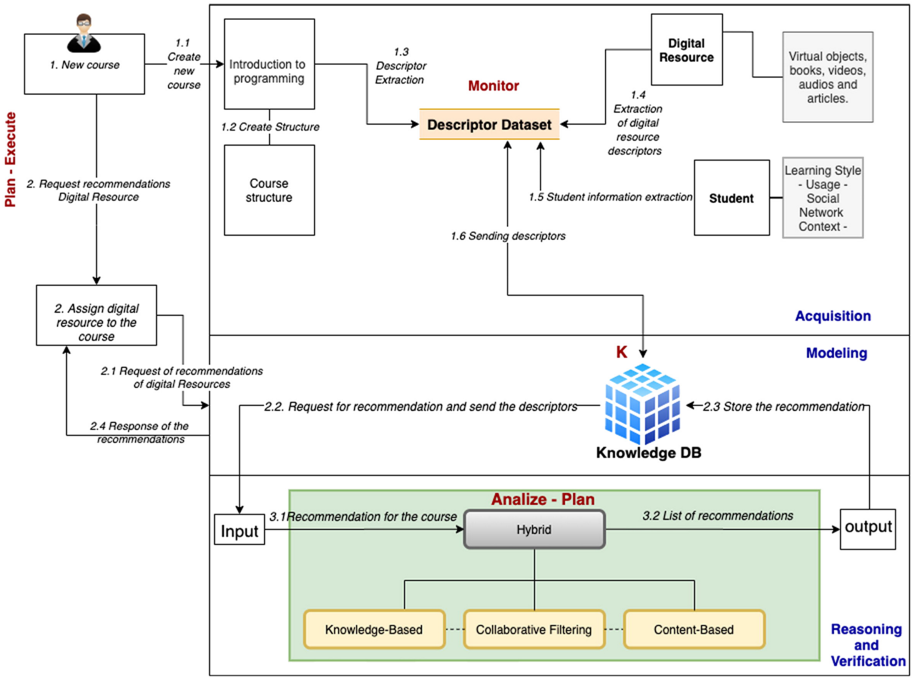
1. New course
1.1 Create new course
Introduction to programming
1.2 Create Structure
Course structure
1.3 Descriptor Extraction
Monitor
Descriptor Dataset
Digital Resource
Virtual objects, books, videos, audios and articles.
1.4 Extraction of digital resource descriptors
1.5 Student information extraction
Student
Learning Style - Usage - Social Network Context -
1.6 Sending descriptors
Plan - Execute
2. Request recommendations Digital Resource
2. Assign digital resource to the course
2.1 Request of recommendations of digital Resources
2.4 Response of the recommendations
Acquisition
Modeling
K
2.2. Request for recommendation and send the descriptors
2.3 Store the recommendation
Knowledge DB
Analize - Plan
Input
3.1Recommendation for the course
Hybrid
3.2 List of recommendations
output
Knowledge-Based
Collaborative Filtering
Content-Based
Reasoning and Verification

Figure 9.
Recommendation process with our recommendation system for the creation of the course "Introduction to programming".

- *Acquisition layer*:

1. **New Course:** At the beginning of the process, the teacher must create the new course "Introduction to programming" in the system through the following steps.

1.1 Create New Course (phase 1): The teacher fills the general information about the course using the CEDS metadata in the course metamodel.

1.2 Create structure (phase 1): The teacher builds the general content of the course (themes and sub-themes) using the ADDIE model in the course metamodel.

1.3 Descriptor Extraction: The system extracts the descriptors of the course "Introduction to Programming course" to be stored in the descriptor Dataset.

1.4 Extraction of digital resource descriptors (phase 2): The system extracts the information of the digital resources stored in the repositories that can be used in the course, according to its digital resource metamodel. For example, videos about algorithms, open programming logic books, among others. This extraction action is permanently carried out.

1.5 Student information extraction (phase 2): This layer carries out different types of extraction of information about the student. The extraction of the learning styles of the students who are going to take the programming logic course can be from a quiz defined by the Felder-Silverman model, or using information about its activities in the VLE or social networks. Previous works have proposed approaches to that based on these ideas [11,10,35]. Also, its information in the academic system is used, in order to have an accurate information about it.

1.6 Contextual information extraction (phase 3): This layer extracts the information about the educational context, in order to fill-out the general context metadata model, which is composed of three parts: internal context, external context and Edge Context.

1.7 Sending descriptors: The descriptors are sent to the modeling layer, where they are stored in the knowledge database.

- *Modeling Layer*:

2. **Assign digital resources to the course:** When the teacher makes the creation of the academic course, he must select the digital resources that best suits the pedagogical objectives, the structure of the course and the needs of the students. For that, he uses our recommendation system in the next way:

2.1 Request of recommendations of digital resources: The teacher requests recommendations of digital resources for the course of Introduction to programming. The request is received by the recommendation system.

2.2 Determine and send the descriptors: The knowledge base is used to extract the descriptors according to the request, and sends them for the recommendation to the reasoning and verification layer. The descriptors are about the course and students.

2.3 Store the recommendations: the recommendations determined by the Reasoning and verification layer are stored in the knowledge base.

2.4 Response of the recommendations: The modeling layer sends the final recommendations to the teacher, such that he can select them and integrate them into the final course.

- *Reasoning and verification layer:*

3. **Recommendation of digital resources for the course:** This layer carries out the recommendation process in our architecture. For that, it is made the following

3.1 Recommendation for the course: with the recommendation request sent from the knowledge base is started the hybrid recommendation process. The hybrid algorithm orchestrates the recommendation process using the different information and recommendation algorithms (Based on knowledge, collaborative filter, and/or content). The hybrid algorithm identifies the recommendation algorithms to be used and the way to mix them, and ranks the digital resources that can be used in the course, using all the available information of the students, the digital resources, the course "Introduction to Programming", and context. The information is sent to each of type of selected recommendation algorithm that will be used in the recommendation, and the results are analyzed by the hybrid algorithm in order to rank them.

3.2 List of recommendations: The hybrid algorithm sends the list of recommendations as the output of the reasoning and verification layer.

- *MAPE-K loop*

4. **Autonomous behavior of our Recommendation System:** To guarantee the autonomous behavior of the architecture, it must comply with the phases of the MAPE-K model. Particularly, after the initial recommendation of the digital resources during the creation of the course, it is required that our recommendation system can autonomously update the recommendations. In the example of creation of the course "Introduction to programming", when the course is activated in the VLE, the behavior would be as follows:

Monitoring: it observes the learning process in the course. In addition, it permanently extracts the current information of the context, of the students in the course, among other data dimensions, to be stored in the descriptor dataset (knowledge database).

knowledge component: it stores all the datasets of the descriptors, recommended digital resources, and specific information that needs the MAPE-K loop.

Analysis – Planning: The information extracted by the monitoring component is processed by the hybrid recommendation algorithm. Each used recommendation algorithm selects the best recommendation for the course "Introduction to Programming", according to the information available.

Planning and execution: this phase, in this case, deploys the recommend digital resources in the VLE. In this way, our system can execute new recommendations of digital resources that are adjusted to the needs of the course and the characteristics of the students, during the learning process.

5. Comparison with previous works

In the comparison with the previous works, four criteria were used: (A) recommendation architecture uses a hybrid algorithm, (B) recommendation architecture is based on intelligent paradigms, (C) it is an autonomous recommendation architecture, and (D) recommendation architecture analyzes the context of the users. The Table 3 shows the behavior of the previous works according to these criteria. Our recommendation architecture is the only one, which complies with the four comparison characteristics. It is a recommendation architecture that uses hybrid algorithms, in order to combine content-based, knowledge-based and collaborative-filter algorithms. In addition, it is a recommendation architecture based on the principles of the intelligent paradigms, with acquisition, modeling, and reasoning phases. Also, the architecture has an autonomous engine based on the MAPE-K loop model (Monitor-Analyze-Plan-Execute

Publications	A	B	C	D
[36]		X		X
[37]		X		
[38]	X	X		X
[7]	X			
[3]		X		X
[39]		X	X	X
[40]				X
[41]		X	X	X
[15]		X		
[14]		X		X
Our architecture	X	X	X	X

Table 3. Comparison with previous works.

plus Knowledge), which is an autonomous administrator of the recommendation process. Finally, the recommendation architecture analyzes the context by extracting variables from various sources, such as social networks, usage logs, devices, among others.

In general, few papers exploit the advantages of the different types of recommendation systems in their architecture, only [38,7] and our proposition. There are a lot of architectures of intelligent recommendation systems, which use artificial intelligence techniques for the processing of the information before the recommendation, among other things. But, with respect to an architecture of an autonomous recommendation system, only there are three works, [39,41], and our work. Finally, there are a lot of architectures that carry out the analysis of the context during the recommendations. In general, our proposal is the only architecture that combines these four aspects. The integration of these dimensions/aspects into a recommendation system confers great advantages. For example, it can adapt to its users through its learning mechanisms; it enables to consider context information to personalize its recommendations; it can self-organize to improve its performance without requiring external interventions; and finally, it integrates the different types of recommendation mechanisms (collaborative, content-based, etc.) to exploit their advantages during the recommendation process. The integration of these aspects introduces computational costs and implementation challenges that must be analyzed in future works.

6. Conclusions

An architecture of an intelligent and autonomous recommendation system is proposed in this work, which can be used in any VLE, with the aim of efficiently recommending digital resources. The architecture extracts information from the context, of the students, identifying variables such as individual learning styles, socioeconomic information, connection log information, location information, among others.

In the architecture, a hybrid recommendation system is proposed where it aims to orchestrate and manage effectively the recommendation process, depending on the available information and recommendation needs. A hybrid algorithm that integrates approaches collaborative filter, based on content and knowledge is defined, so that their combinations can be parallel or sequential. The implementation of the MAPE-K model for the architecture ensures that recommendation process has an autonomous behavior during the learning process, through its monitoring, analysis, planning and execution.

Another aspect very important of our architecture is its capability to use different student information, which allow it to carry out assertive decisions. It uses the learning styles of the students, the context information, the social networks, among other sources, to select the best digital resources. In this way, it considers the characteristics of the students, the devices to access the VLE, the characteristics of the environment, the way like is the content consumption, the collaborative contexts, etc. to give a more accurate recommendation.

Our architecture has been tested in the creation of an academic course. It is used during the creation of the course, and when the course is activated in the VLE. During the creation, it proposes digital resources to the teacher, for their selection. During the execution of the course in the VLE, our recommendation system deploys its intelligent and autonomous behaviour, proposing new recommendations that meet the current needs of the learning process.

As a future work, it is projected the integration of a motor of multimodal sentiment to the model of autonomous recommendation, to improve the recommendation process with the emotions and sentiment of the students. Also, for the creation of the prototype of our recommendation system, an extraction mechanism will be done that considers various sources of opinion and interaction generated by students, using a multimodal analysis. Other future work is the implementation of the hybrid recommendation algorithm, which orchestrates the integration process of the different recommendation types. A last future work is the integration of a prototype of the ARS in a VLE, and to test in a real educational context, to validate the quality of the recommended LRs in a learning process using metrics that evaluate their impacts in the students.

References

[1] A. Gediminas, T. Allexander, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749.

[2] G. Pavlidis, Recommender systems, cultural heritage applications, and the way forward, J. Cultural Heritage 35 (2019) 183–196.

[3] S. Sukrat, B. Papasratorn, An architectural framework for developing a recommendation system to enhance vendors' capability in C2C social commerce, Social Network Anal. Mining 8 (1) (2018).

[4] M.A. Azad, S. Bag, F. Hao, Privbox: Verifiable decentralized reputation system for online marketplaces, Future Gener. Computer Syst. 89 (2018) 44–57.

[5] X. Zhang, J. Edwards, J. Harding, Personalised online sales using web usage data mining, Comput. Ind. 58 (8) (2007) 772–782.

[6] T.A. Syed, S.S.K. Nair, Personalized recommendation system for advanced learning management systems, ACM Int. Conf. Proc. Ser. (2018) 90–95.

[7] M. Vladoiu, Z. Constantinescu, G. Moise, Qorect–a case-based framework for quality-based recommending open courseware and open educational resources, in: International Conference on Computational Collective Intelligence, Springer, 2013, pp. 681–690.

[8] F. Zhu, H.H.S. Ip, A.W.P. Fok, J. Cao, PeRES: A personalized recommendation education system based on multi-agents and SCORM, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 4823, LNCS, 2008, pp. 31–42.

[9] M.K. Khribi, M. Jemni, O. Nasraoui, Recommendation systems for personalized technology-enhanced learning, Commun. ACM 41 (1) (2015) 159–180.

[10] Y. Delgado, D. Hernández, J. Aguilar, J. Altamiranda, Reconfiguration of the learning style in the computer learning platform based on cloud paradigm, in: 2016 35th International Conference of the Chilean Computer Science Society (SCCC), 2016, pp. 1–12.

[11] J. Aguilar, O. Buendia, A. Pinto, J. Gutiérrez, Social learning analytics for determining learning styles in a smart classroom, Interactive Learn. Environ. (2019) 1–17.

[12] E.R. Núñez-Valdéz, J.M.C. Lovelle, O.S. Martínez, V. García-Díaz, P.O. De Pablos, C.E.M. Marín, Implicit feedback techniques on recommender systems applied to electronic books, Comput. Hum. Behav. 28 (4) (2012) 1186–1193.

[13] N. Golovin, E. Rahm, Reinforcement learning architecture for web recommendations, International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004, Vol. 1, 2004, pp. 398–402.

[14] L.A.D. Knob, R.P. Esteves, L.Z. Granville and L.M.R. Tarouco, Mitigating elephant flows in sdn-based ixp networks, in 2017 IEEE Symposium on Computers and Communications (ISCC), July 2017, pp. 1352–1359.

[15] J. Aguilar, P. Valdiviezo-Díaz, G. Riofrio, A general framework for intelligent recommender systems, Appl. Comput. Inform. 13 (2) (2017) 147–160.

[16] J. Aguilar, Different dynamic causal relationship approaches for cognitive maps, Appl. Soft Comput. 13 (1) (2013) 271–282 Available: http://www.sciencedirect.com/science/article/pii/S1568494612003948.

[17] Min Xu, Yu-Hui Qiu, Jing Qiu, Xiao-Fang Wang, Using autonomous agents to make recommendation in a modeling market, Proceedings. International Conference on Machine Learning and Cybernetics, Vol. 3, 2002, pp. 1322–1325.

[18] H. Park, J. Lee, A framework of context-awareness for ubiquitous computing middlewares, Proceedings – Fourth Annual ACIS International Conference on Computer and Information Science, Vol. 2005, ICIS, 2005, pp. 369–374.

[19] J.K. Kim, H.K. Kim, Y.H. Cho, A user-oriented contents recommendation system in peer-to-peer architecture, Expert Syst. Appl. 34 (1) (2008) 300–312.

[20] J.R. McBride, A.R. Lupini, M.A. Schreuder, N.J. Smith, S.J. Pennycook, S.J. Rosenthal, Few-layer graphene as a support film for transmission electron microscopy imaging of nanoparticles, ACS Appl. Mater. Interfaces 1 (12) (2009) 2886–2892.

[21] K. Kruse, Introduction to instructional design and the addie model, Retrieved January, vol. 26, p. 2005, 2002.

[22] NCES, Common Education Data Standards, pp. 37–39, 2014. [Online]. Available: http://ceds.ed.gov.

[23] L.T.S.C. of the IEEE P1484.12.2/D1, Final Standard for Learning Technology - Learning Object Metadata, pp. i–32, 2002.

[24] G.D. Dey, K. Anind, Abowd Providing architectural support for building context-aware applications, Georgia Institute of Technology, 2000. Ph.D. dissertation.

[25] J. Aguilar, M. Jerez, T. Rodríguez, Cameonto: context awareness meta ontology modeling, Appl. Comput. Inform. 14 (2) (2018) 202–213.

[26] L. Zhong-Jun, L. Guan-Yu, P. Ying, A method of meta-context ontology modeling and uncertainty reasoning in swot, in: 2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2016, pp. 128–135.

[27] R.M. Felder, L.K. Silverman, et al., Learning and teaching styles in engineering education, Eng. Educ. 78 (7) (1988) 674–681.

[28] B.S. Bloom, M.D. Engelhart, E.J. Furst, W.H. Hill, D.R. Krathwohl et al., Taxonomy of educational objectives: the classification of educational goals. handbook i: cognitive domain. new york: David mckay company, Inc. (7th Edition 1972), 1956.

[29] J. Bozo, R. Alarcón, S. Iribarra, Recommending learning objects according to a teachers' contex model, European Conference on Technology Enhanced Learning, Springer, 2010, pp. 470–475.

[30] R. Burke, Hybrid recommender systems: survey and experiments, User Model. User-adapted Inter. 12 (4) (2002) 331–370.

[31] R. Burke, Hybrid web recommender systems, The adaptive web, Springer, 2007, pp. 377–408.

[32] C. Desrosiers, G. Karypis, A comprehensive survey of neighborhood-based recommendation methods, Recommender systems handbook, Springer, 2011, pp. 107–144.

[33] X. Ning, C. Desrosiers, G. Karypis, A comprehensive survey of neighborhood-based recommendation methods, Recommender systems handbook, Springer, 2015, pp. 37–76.

[34] D. Jannach, M. Zanker, A. Felfernig, G. Friedrich, Recommender Systems: An Introduction, Cambridge University Press, 2010.

[35] J. Aguilar, J. Fuentes, Management system of learning paradigms using oda, 2014 XL Latin American Computing Conference (CLEI), 2014, pp. 1–12.

[36] B. Bouihi, M. Bahaj, Ontology and rule-based recommender system for e-learning applications, Int. J. Emerg. Tech. Learn. (iJET) 14 (15) (2019) 4–13.

[37] N.S. Raj, V. Renumol, Architecture of an adaptive personalized learning environment (aple) for content recommendation, in: Proceedings of the 2nd International Conference on Digital Technology in Education, 2018, pp. 17–22.

[38] A. Baldominos, E. Albacete, Y. Saez, P. Isasi, A scalable machine learning online service for big data real-time analysis, 2014 IEEE Symposium on Computational Intelligence in Big Data (CIBD), IEEE, 2014, pp. 1–8.

[39] S. Wan, Z. Niu, An e-learning recommendation approach based on the self-organization of learning resource, Knowl.-Based Syst. 160 (2018) 71–87.

[40] Q. Xie, K. Wu, J. Xu, P. He, M. Chen, Personalized context-aware QoS prediction for web services based on collaborative filtering, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 6441, LNAI, 2010, pp. 368–375. no. PART 2.

[41] A. Klašnja-Milićević, B. Vesin, M. Ivanović, Social tagging strategy for enhancing e-learning experience, Comput. Educ. 118 (2018) 166–181.

Corresponding author
Jose Aguilar can be contacted at: aguilar@ula.ve