2018 IEEE International Conference on Information Reuse and Integration for Data Science

Web-Based Recommendation System Architecture for Knowledge Reuse in MOOCs Ecosystems

Rodrigo Campos
PPGI – UFRJ
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
rodrigo.campos@ufrj.br

Rodrigo Pereira dos Santos
PPGI – UNIRIO
Federal University of the State of Rio de Janeiro
Rio de Janeiro, Brazil
rps@uniriotec.br

Jonice Oliveira
PPGI - UFRJ
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
jonice@dcc.ufrj.br

Abstract— **With the success of MOOCs (Massive Open Online Courses) in recent years and the emergence of several courses in these environments, students face difficulties in choosing the best materials, modules or courses. Some recommendation systems have come up to address such problem. However, they provide recommendation limited to a specific MOOC provider. This work proposes an architecture of a web-based recommendation system to support students in finding suitable courses or materials to reach their interests. In addition, this work will raise the fundamental actors and roles of an underlying software ecosystem (SECO) and relate them to those involved in MOOCs ecosystems. The main motivation is that several elements of such ecosystems derive from virtual learning community ecosystems, failing to reproduce particularities of MOOCs learning community ecosystem. Therefore, the system resulting from this architecture also aims to achieve the knowledge reuse in these MOOCs ecosystems, i.e., demands, improvements and software sharing over the platforms.**

Keywords- Recommendation Systems, MOOCs Ecosystems, Software Architecture, Knowledge Reuse, Software Ecosystems

I. INTRODUCTION

With the frequent evolution of technological resources, several segments have been adapted and improved. One of these segments is education. Among the existing options that changes the way students learn, we can find Massive Open Online Course (MOOC). It is defined as a new educational philosophy with new values based on openness, ethics of participation and collaboration, i.e., more than a simple open education [1].

MOOCs are still in the growth phase since their inception in 2008, being considered by some authors as one of the most advanced online learning systems, affecting and reconsidering the concepts of even the most traditional educational institutions [2]. In order to ensure a sustainable development of MOOCs as well as their entire learning community, some authors have proposed approaches based on the field of software ecosystem (SECO), conceptualizing the MOOC learning community ecosystem [3], or simply MOOCs ecosystems. This sustainable learning ecosystem can bring benefits, such as work partnership formation or alliances between companies for all ecosystems' stakeholders.

However, these ecosystems absorb some students' problems already known by MOOC providers, such as the difficulty in choosing suitable content through several MOOCs at the market (Section 6 lists works that have applied recommendation systems to address this problem). These assignments only consider course recommendations altogether, and often the most appropriate content for students is only part of the course (e.g., modules), or multiple modules at different providers.

This paper proposes an architecture of a web-based recommendation system to support students in finding suitable courses. Moreover, it should be able to recommend not only courses, but also parts of courses in MOOCs ecosystems, considering the existence of more than one provider in the recommendation process. Thus, a specific system resulting from this architecture also aims to achieve knowledge reuse in these MOOCs ecosystems, i.e., demands, improvements and sharing of software over the platforms.

This paper is organized as follows: Section 2 explores the importance of treating MOOCs with an approach inspired by SECO, explaining existing roles and types of knowledge shared among stakeholders; Section 3 presents an overview of what can be extracted from each MOOC provider, as well as limitations; in Section 4, we propose the web-based recommendation system architecture, with an

978-1-5386-2659-7/18/$31.00 ©2018 IEEE
DOI 10.1109/IRI.2018.00036

exemplification in Section 5; Section 6 presents related work, including some researches that contributed to the construction of our proposal; finally, Section 7 concludes the paper with final discussions and future work.

II. MOOCs Ecosystems

In addition to the development of sustainable MOOCs, other benefits of a SECO perspective can be pointed [4]:

- It provides software evolution and innovation in organizations, increasing the attractiveness (new players) and thus promoting the platform success;
- It helps to analyze and understand software architecture to decide which platform can be used (the largest benefit exploited in this study);
- It helps knowledge sharing through multiple and independent entities, strengthening cooperation;
- It supports business identification tasks, product architecture design, or risk identification.

SECO concept is defined as a set of businesses functioning as a unit and interacting with a shared market for software and services, together with the relationships among them, frequently underpinned by a common technological platform or market, and operating through the exchange of information, resources and artifacts [5]. To be able to define MOOCs ecosystems, it is necessary to delimit the actors' roles and their relationships. The problem faced by other authors in such task is that several elements belonging to these ecosystems derive from virtual learning community ecosystems, failing to reproduce the promising single features of MOOCs learning community ecosystem [3].

To be able to evolve the MOOC learning community ecosystem space suggested in [3], we need to understand the roles in the SECO and what each one represents, at first. To do so, Table I addresses an adaptation of the categorizations presented in [6], which in turn covers and reconciles different concepts regarding the roles of actors in these ecosystems.

The roles in MOOCs ecosystems can be analyzed from their platforms. The main function is to make interactions between several groups of agents easier. According to Belleflamme and Jacqmin [7], the most present groups are: students, teachers, higher education institutions, and other private actors (advertisers or employers). However, these definitions are not accurate. There is no consensus on the categorization of these agents. For example, Daradoumis et al. [8] adds course designers and managers in the agent list in addition to referring to students as learners (and teachers as tutors), thus requiring a more complete infrastructure for supporting the processes of a MOOC provider (not only considering the main actors). Once again, course designers are called as "those who make MOOC" [3].

It is possible to correlate some of these groups in order to create MOOCs ecosystems where such groups are given pre-defined roles according to the SECO approach. These groups circulate through the MOOCs platforms and interact with each other, offering something in exchange.

Table II shows the correlation of these groups, thus demonstrating the MOOCs ecosystems creation. It must be added that there are three stages, detailed in [3].

- In the first stage, students register their account in the MOOC provider, linking an email and registering some important data. They login for the first time with the new account; if successfully, it can already choose the course they want to enroll. At that moment, these students play as consumers, since they can already acquire products (courses), but do not have enough knowledge to interact with other participants in the network [3];
- In the second stage, students are already able to interact in existing provider's discussions. At the same time, they seek out external knowledge, downloading content from the internet, processing and creating their own learning materials, and sharing this external content with other members in the learning network. They share resources, notes and personal learning process in forums, wikis and other means of interaction. This process of getting data from the ecosystem and inserting external data according to their perception makes these students considered as decomposers [3]. In the SECO context, decomposers would be similar to dominator because they extract as much value from the ecosystem as possible, destroying it [9];
- In the third stage, students begin to collaborate with the learning community, being able to help new students in the learning process, such as dealing with eventual difficulties and doubts, i.e., increasing the community's strength. This analysis makes such students as suppliers [3].

TABLE I. DESCRIPTION OF SECO ACTORS' ROLES. SOURCE: [6]

Hub	***Keystone***	Adds value to SECO and is primarily responsible for maintaining health, i.e., longevity and growth. It can represent the dominant entity of influence.
	Dominator	Extracts value from SECO, putting its health and sustainability at risk.
Niche Player	***Customer***	Represents the customer, who generated the need for the SECO software products.
	Competitor	It tries to extract value from the ecosystem but does not threaten the SECO's health.
	Supplier	Actor providing one or more products or services required by the ecosystem.
	Vendor	Sells SECO software products. Can be classified as Reseller, Independent Software Vendor (ISV), or Value-added Reseller (VAR).
	Developer	Internal developer linked to SECO formative entities, being classified as Influencer, Hedger or Disciple.
External Actor	***3rd-party developers***	Promotes SECO and its products, and can propose improvements; similar to Influencer, but external to SECO, having no formal bond with Keystone.
	End-user	Product's final user, but differs from Customer for not hiring Keystone service.
	External Partner	Contributes to the SECO well-being through attitudes, such as the promotion of SECO and its products, also proposing improvements.

TABLE II. RELATIONS BETWEEN SECO'S ROLES AND MOOCS ECOSYSTEMS' ROLES

	1st stage	2nd stage	3rd stage
Keystone	Higher Education Institutions		
Dominator	-	Students	Students
Customer	Students		
Competitor	Advertisers		
Supplier	Teachers, Course Designers	Teachers, Course Designers	Teachers, Course Designers, Students
Vendor	MOOCs Providers (ISV)		
Developer	Course Designers		
3rd-party developers	-	-	-
End-user	-	-	-
External Partner	Employers		

To understand the dynamics of knowledge exchange from these interactions, at first, we need to reinforce the differences amongst data, information and knowledge. In this case, data are simple facts that become information if combined into an understandable structure, whereas information becomes knowledge if it is placed in a context as the result of cognitive processing/validation, besides being able to serve to make predictions [10].

In MOOC learning community ecosystem, learners transfer information though WIKI, forums, chats and other communication tools provided by MOOC. Moreover, they share information via logs [3]. This knowledge exchange takes place based on the links between the ecosystem's roles. To identify the possible interactions between the roles of the MOOCs ecosystems' actors, we investigated works that explain the main processes used by these providers in the technical literature, which resulted in the categorization presented in Table III.

Fig. 1 shows a graph where a node represents each actor and edges represent their interactions. A MOOC provider's node is the ecosystem's actor that has connection with all other actors. The direction of the graphs edges is also equivalent to the direction of the arrows arranged in the first column of Table III.

The interaction that mostly generates knowledge for the ecosystem is Student → MOOCs Providers. As the entire student learning process is recorded in the providers' platforms, students extract most of the knowledge needed to fill gaps from this interaction. In other words, knowledge will be extracted from audio, video, games, animation, blog, chat, forum, email, or virtual communities.

A student can also provide knowledge to the ecosystem, since their actions are useful to other actors, such as their registration information or the use of providers' platforms to interact with courses, assisting other students in forums or even answering proposed activities, for example.

TABLE III. INTERACTION BETWEEN DIFFERENT GROUPS OF STAKEHOLDERS

Interaction	This interaction exists…
Students --> MOOCs Providers	to help students follow courses taught by teachers.
Students --> Higher Education Institutions	to help students improve their employability, looking for information on the course quality.
Students <--> Employers	because students may exercise their abilities with employers from the ecosystem who, in turn, have access, via a MOOC platform, to a large pool of students as well as to detailed data about their skills.
Students --> Advertisers	if the advertiser's presence and their payments allow platforms to offer courses to students for free.
Students --> Students	because students might be influenced; as a result, student learning outcomes will depend on interactions with fellow students.
Teachers --> MOOCs Providers	because teachers seek to disseminate their teaching materials and experiment new pedagogies.
Teachers --> Higher Education Institutions	because even if teachers can offer a course in their own name, they usually still depend on their respective university.
Teachers --> Employers	because teachers value employers' presence indirectly if they contributes to attract more students.
Teachers <--> Students	because they can interact with each other via the MOOCs' platform, by social media, or by telephone, meeting and answering activities in the real life. Currently, students have organized offline meetings.
Higher Education Institutions --> MOOCs Providers	because institutions can decide to invest money and time in a MOOC platform.
Higher Education Institutions --> Teachers	since institutions pay teachers and encourage them via other non-monetary rewards.
Higher Education Institutions --> Employers	because institutions only value the participation of private actors to the platform indirectly.
Employers --> MOOCs Providers	because employers see MOOCs as a flexible and cheap tool to train their staff.
Advertisers --> MOOCs Providers	since advertisers are ready to pay before having access to the platforms' visitors, as well as information about them.
Course Designers --> MOOCs Providers	because courses are designed and published in the providers' platforms.

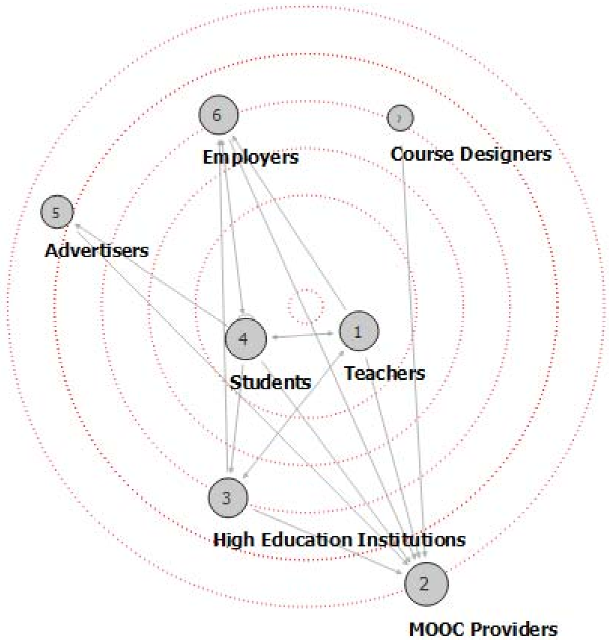

Figure 1. Interactions from the network of actors in the ecosystem

This information is stored in the providers' databases and support teachers with explicit knowledge (querying, student assessment etc.), or educational institutions that can extract knowledge for statistics purposes and for decision making regarding the courses' pedagogical plans.

III. MOOCs Data Extraction

One of the stages of our study is to map the availability and opening of users' data in each MOOCs' provider. An initial search was performed in the technical literature. The most commonly used providers are listed in Table IV.

In order to choose providers used in recommendation processes, it was verified which API had 'total availability' since all of them had an API. From Table IV, only the edX API did not have it. As highlighted in [11], this API requires an OAuth authentication to use. As such, the selected providers were: Coursera, Udacity, Khan Academy, and OCW.

The other columns in Table IV show what kind of information can be extracted from each provider. It is also possible to see that the format of all files extracted from the APIs is JSON, guaranteeing uniqueness in this regard.

IV. Proposed Architecture

This section presents a web-based recommendation system architecture and its stages, including the data extraction step described in Section 3. The architecture model is presented in Fig. 2. There is the use of Linked Data to build open, collaborative recommendation system in this proposed context, among other components. This architecture is also based on MOOCLink [11], a system that integrates data from several MOOCs providers and has algorithms for incrementally updating linked data to maintain their quality.

TABLE IV. Information about provider extractions

	How to obtain data?	It is possible to extract…	Data Format
Coursera	Coursera API	all of Coursera's courses, instructors, and partnering universities	JSON
edX	Crawler	limited information	Several
	edX API	Courses API, Data Analytics API, Discussion API, Enrollment API, Grades API, User API, Discovery API	JSON
	RSS Feed	a list of edX course list	XML
Udacity	Crawler	limited information	Several
	Udacity API	course catalog information and nanodegree courses	JSON
Khan	Khan API	"topic tree" which gives the entire hierarchy of Khan Academy's course offerings. It can also obtain the list of all badges, badge categories, details of a particular course etc.	JSON
OCW	OCW API	indexes of all these courses (e.g., links, hash, provider, language, tags, author, title, description, published, indexed, modified, categories)	JSON
	Excel Dump	all the courses (e.g., links, hash, provider, language, tags, author, title, description, published, indexed, modified, categories)	Excel

The choice of using an open recommendation system (i.e., linked open data assisting in the recommendation process) is due to the fact that there are several MOOCs providers and each provider has a different data source. Exploring multiple sources can enrich recommendation as well as alleviate problems related to recommendation systems that use Collaborative Filtering: cold-start (to make recommendation if no data about an item or user are available) and low user ratings problem [12].

Although this architecture proposes a real involvement of almost all the ecosystem's actors, the recommendation process mainly aims to benefit the customer (student). It will positively affect other roles, since students will be more satisfied with their learned content after making a good recommendation, so it can reduce dropout rates and the knowledge gap.

The directional arrows in Fig. 2 determine the flow of the recommendation process. Dashed lines indicate actions that are performed outside the main stream. The entire process is divided into some layers and steps:

- **User** submits search to the **Web-Based Recommender System**. Then, to make the recommendation, the system will query a **Knowledge Base** that holds information from the ecosystem's **Input Data** (information to support users make a recommendation). Input data contain information inferred from the recommendation system (e.g., user's competencies), curriculum and other data from MOOCs providers. Such information represents possible interactions that these users have in the ecosystem. Therefore, the architecture presented in Fig. 2 contemplates in that box the actors that who interact with students (the only exception is the course designer);

- Next, it starts **SPARQL Endpoint**, a semantic web technology that allows querying data formatted in the **RDF Data** (Resource Description Framework) model between different schemas;
- SPARQL queries data that is in the **Background Data** layer (i.e., system information before starting the recommendation process). This information was previously handled through the **Apache Jena Fuseki Server** to perform the RDF file hosting stage. In our work, Apache Jena Fuseki groups RDFs from four different MOOC providers, thus allowing a broader recommendation. In open recommendation systems, such process is done by an integration service [12]. Section 3 discusses which providers were selected for the scope of this work as well as their extraction formats;
- Data is still in the RDF model. Before being properly sent to the recommendation layer, it is converted into a **user-item matrix**;

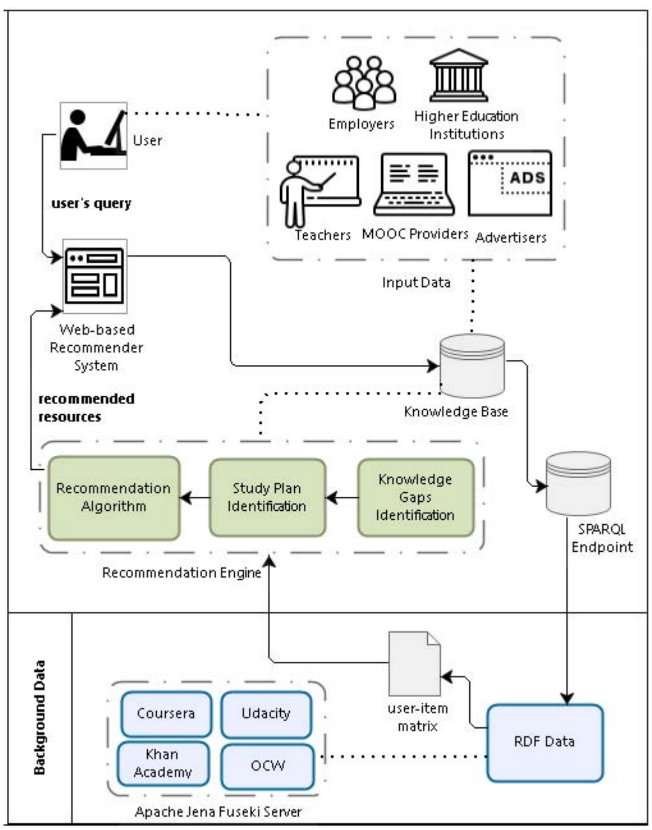

Figure 2. Proposed architecture model

- The recommendation engine layer applies three procedures in the matrix user-item. It also uses the knowledge base to enrich this process. First, **Knowledge Gaps Identification** will collect user's existing and desired skills and will identify the existing knowledge gap. Then, **Study Plan Identification** starts getting information from the courses through the matrix and will decide what courses or modules will be suitable for the study plan, filling the user gap. Then, **Recommendation Algorithm** is applied in order to rank these results and send recommendations back to the system.

The architecture model proposed in this paper aims to support the student in their decision process, but also aims to present sufficient resources to reach the four stages of knowledge reuse defined by Markus [13]:

- Capturing or documenting knowledge;
- Packaging knowledge for reuse;
- Distributing or disseminating knowledge (providing people with access to it);
- Reusing knowledge.

Since software asset management is decentralized in a SECO, the MOOCs ecosystem has shared management in order to strength stakeholder collaborations, as shown in Table III. For example, in Coursera, any user in the community can create a course and make it available over the platform. Universities also engage in development partnerships with providers. In other words, the ecosystem perspective treats stakeholders with a stake in managing this reusable knowledge by consuming, supplying or doing any other participation explained in Table II.

With this proposal, we have inputs for benefiting other stakeholders through grouped, stored and reusable knowledge, in addition to user feedback. These benefits include checking new course demands, enhancements to existing content, or even signaling software improvements from providers, as well as possible partnerships or alliances – if some content or demand for common interest software is identified.

V. EXAMPLE OF USE

In order to better visualize how the recommendation process works and to provide a preliminary evaluation of the proposed steps, this section presents an example of use. It simulates a user interested in receiving recommendations on a specific knowledge area in order to check which modules/courses would better fill their knowledge gap.

In this example, a user with a dummy name (Tom) enters the recommendation system after taking the MOOC course "Intro to HTML/CSS: Making webpages"[1] through the Khan Academy provider. Tom's goal is to study the Structured Query Language (SQL) used in webpages, since the course referred to it do not provide explanations on this topic. Tom has a knowledge gap. By searching for the "sql web", he realized that there were several options, but perhaps none was interesting to his line of learning. When accessing the recommendation system, Tom performs the same search performed in the Khan Academy ("sql web"). The system

[1]https://pt.khanacademy.org/computing/computer-programming/html-css

also has support for some search filters. Tom has selected some filters as shown in Fig. 3: Start Soon (for availability), Introductory (in level), English (as the language), and Free (for value).

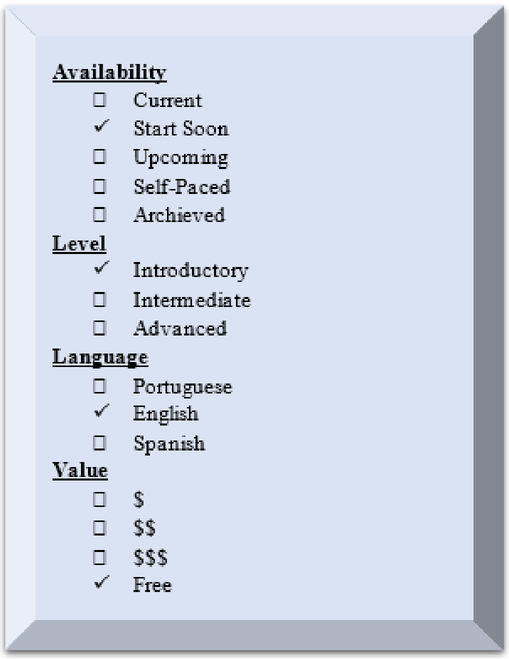

Figure 3. Examples of selected filters in the main search of the recommendation system

The recommendation process begins with the query in the knowledge base, retrieving input data from that user in the ecosystem. Next, the SPARQL search the Background Data (that has been manually generated), grouping information from several courses and including the Khan Academy courses that Tom has searched for integrating data. This is necessary because different data sources will use different vocabulary to denote the connection between students and items. Using the Apache Jena Fuseki Server, the raw data is captured from MOOCs providers into an acceptable RDF, creating and maintaining a SPARQL endpoint and finally executing SPARQL queries.

With the courses, part of courses, modules and contents gathered according to the student's query, information is sent to the recommendation engine. In addition, the recommender needs to look at all the student's knowledge base, their ecosystem interaction information, history and other data to predict which course (or parts of courses) would best fit the student' needs.

Since the process does not analyze only complete courses, it allows to recommend a book that is part of OCW's Software Engineering for Web Applications course. The book's name is "SQL for Web Nerds"[2]. With the loaded knowledge base, it is possible to visualize which courses Tom had already done, avoiding that the recommender suggests the same course from Khan Academy. Considering that this was the most well ranked content based on the algorithms, the result will instantly return to the recommendation system's screen.

This entire recommendation process is recorded in the knowledge base for future recommendations. In the next recommendations, if a previously recommended course does not appear as a finished course (or course being taken by the student), this may mean that the previously recommendation was unsuccessful. Therefore, saving recommendation records can improve the system process itself.

VI. RELATED WORK

Some works have already been proposed to make recommendations through MOOCs. MOOCRec.com [14] recommends courses to users so that they can acquire new skills that are expected from their ideal job. Another work that recommends courses which best fit students' personal interests is the MOOC-Rec [15]. Both systems do not recommend parts of such courses, such as modules.

On the other hand, OERecommender project [16] recommends not only simply courses but also Open Educational Resources (OER). This concept is very similar to the MOOCs. However, OERs consist of 'any kind of educational material in the public domain or associated with an open license' [1, p. 8]. Meanwhile, MOOCs are defined as 'online courses accessible to anyone on the web' where 'institutions have joined in an effort to make education more accessible by teaming up with MOOC providers' [11]. According to Aires [1], OERs, MOOCs, free software and other concepts are bases for one of the most important educational movements of the 21st century: Online Open Education.

As it was not implemented in the real environment, OERecommender project [16] also presents limitations in OERs classification method for MOOC environment as well as in the methods to get similarity between users. Moreover, it does not verify if the algorithms are effective. Some recommendation methods such as the one presented in [17] have been more effective than OERecommender project [16] for a recommendation process.

Another proposal related to our study is the MOOCLink [11], which integrates different MOOCs courses' providers to aggregate courses based on Linked Data and use data in the web application to discover and compare open courseware. As shown in Table V, no recommendation approach is checked. This is because the gap of MOOCLink: it does not contemplate any system or recommendation module, resembling only in the multi-provider aspect, being described as an aggregation system.

Based on the projects described in this section (MOOCLink [11], MOOCRec.Com [14], MOOC-Rec [15], and OERecommender [16]), Table V presents the recommendation systems' general characteristics to compare related work to our architecture proposed in Section 4. We referred to our proposal as 'RS' in Table V (acronym for 'Recommendation System'). Regarding the recommendation approach, related work uses CBR (Case-Base Reasoning),

[2] https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-171-software-engineering-for-web-applications-fall-2003/readings/

CF (Collaborative Filtering), MF (Matrix Factorization), or Slop-One. Our work uses a combination of CF and content-based filtering, called ‘hybrid recommendation’.

TABLE V. COMPARISON BETWEEN RELATED WORK AND THE WEB-BASED RECOMMENDATION SYSTEM (RS) ARCHITECTURE PROPOSED IN OUR RESEARCH

		[11]	**[14]**	**[16]**	**[15]**	**RS**
Interface	Web	X	X		X	X
	Web Widget			X		
Background Gathering	CAM			X		
	Crawler		X		X	
	Linked Data	X				X
Recommendation Approach	RS related with CBR				X	
	RS designed with CF			X		
	RS uses MF		X			
	RS based on Slop One algorithm			X		
	Hybrid					X
Recommends	Courses		X		X	
	OER			X		
	Courses and its parts (module, relevant content)					X
Status	Not Implemented			X	X	X
	Finished but not running	X				
	Running		X			

Although all projects have a multi-provider approach, i.e., recommendation is carried out based on data from several providers, the proposed recommendation aims to consider courses, part of courses, or modules of the MOOCs, instead of only dealing with the recommendation of MOOCs. In addition, it applies Linked Open Data to collect background data, building an open recommendation system with greater advantages than Crawler or CAM (Contextualized Attention Metadata).

VII. CONCLUSION AND FUTURE WORK

The use of recommendation systems has been growing in recent years, while challenges arise on how to optimize this process, ensuring more (and better) precise algorithms. This work explored the use of Linked Open Data, modeling an architecture for a web-based recommendation system capable of integrating different MOOCs platforms to recommend courses, modules or parts of courses, and relevant materials of students’ interests. Such different platforms, their users and other actors create MOOCs ecosystems and this approach is also explored in our work. Identifying the roles of ecosystem’s actors and how they play in the learning process can bring several benefits to all stakeholders, as highlighted in Section 2.

The main motivation of this work is that several elements of those ecosystems derive from virtual learning community ecosystems, failing to reproduce the promising single features of MOOCs learning community ecosystem. The system resulting from this architecture also aims to achieve the knowledge reuse in these MOOCs ecosystems. The next stage of this work is related to implementing the proposed architecture, selecting and extending the most appropriate algorithms for the presented ecosystem’s context as well as the technologies to support data quality, i.e., to ensure that the database from all providers is always updated.

As future work, we intend to plan a quali-quantitative method to evaluate our proposed approach. In the first stage (quantitative), a controlled experiment will be conduct through analysis of efficiency and effectiveness of recommendation algorithms, comparing results with those obtained from related work. In the second stage (qualitative), a feasibility study will be conducted to evaluate our approach in a real-world situation with two groups of people: one will perform a set of tasks using the proposed solution and another will not. Results will be compared at the end. The sample is formed by technical high school and undergraduate students of a Brazilian higher education institution. Some students have already been employed and they are mostly concluding their courses. After executing the feasibility study, we will apply an evaluation questionnaire to get feedback from these participants. As such, we will collect data from repositories and applied documents to analyze elements of our proposed solution in details.

ACKNOWLEDGMENT

The first author thanks the Federal Institute of Education, Science and Technology of Rio de Janeiro for supporting this research. The second author thanks DPq/UNIRIO for partially supporting this research. Also, the researchers thank CAPES, CNPq and FAPERJ (Brazil) for their financial support to the research group.

REFERENCES

[1] L. Aires, “E-Learning, Online Education and Open Education: A Contribution to a Theoretical Approach,” *RIED. Rev. Iberoam. Educ. a Distancia*, vol. 19, no. 1, Sep. 2015.

[2] P. Bond and F. Leibowitz, “MOOCs and Serials,” *Ser. Rev.*, vol. 39, no. 4, pp. 258–260, Dec. 2013.

[3] K. Shanyun, S. Qin, and Z. Guolin, “Research on the Construction of MOOC Learning Community Ecosystem Circle,” *Proc. - Int. Conf. Educ. Innov. Through Technol. EITT 2015*, vol. 2, pp. 199–203, 2016.

[4] O. A. L. P. Barbosa, R. P. dos Santos, C. F. Alves, C. M. L. Werner, and S. Jansen, “A Systematic Mapping Study on Software Ecosystems from a Three-Dimensional Perspective,” in *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry.*, 1ed., Northampton/USA: Edward Elgar Publishing, 2013, pp. 59–81.

[5] R. P. dos Santos, “Managing and monitoring software ecosystem to support demand and solution analysis,” Ph.D. dissertation, Federal University of Rio de Janeiro, 2016.

[6] T. Lima, R. P. dos Santos, J. Oliveira, and C. Werner, “The importance of socio-technical resources for software ecosystems management,” *J. Innov. Digit. Ecosyst.*, vol. 3, no. 2, pp. 98–113,

Dec. 2016.

[7] P. Belleflamme and J. Jacqmin, “An Economic Appraisal of MOOC Platforms: Business Models and Impacts on Higher Education,” *CESifo Econ. Stud.*, vol. 62, no. 1, pp. 148–169, Mar. 2016.

[8] T. Daradoumis, R. Bassi, F. Xhafa, and S. Caballe, “A Review on Massive E-Learning (MOOC) Design, Delivery and Assessment,” in *2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, 2013, pp. 208–213.

[9] I. van den Berk, S. Jansen, and L. Luinenburg, “Software ecosystems: a software ecosystem strategy assessment model,” in *Proceedings of the Fourth European Conference on Software Architecture Companion Volume - ECSA ’10*, 2010, p. 127.

[10] P. Cooper, “Data, information, knowledge and wisdom,” *Anaesth. Intensive Care Med.*, vol. 18, no. 1, pp. 55–56, Jan. 2017.

[11] C. Dhekne, “MOOCLink: Linking and Maintaining Quality of Data Provided by Various MOOC Providers,” M.S. thesis, Arizona State University, 2016.

[12] B. Heitmann and C. Hayes, “Using Linked Data to Build Open, Collaborative Recommender Systems.,” in *AAAI spring symposium: linked data meets artificial intelligence*, 2010, vol. 2010.

[13] M. L. Markus, “Toward a Theory of Knowledge Reuse Types of Knowledge Reuse Situations and Factors in Reuse Success,” *J. Manag. Inf. Syst.*, vol. 18, no. 1, pp. 57–93, 2001.

[14] P. Symeonidis and D. Malakoudis, “MoocRec.com : Massive Open Online Courses Recommender System,” *Soc. Behav. Sci.*, 2015.

[15] F. Bousbahi and H. Chorfi, “MOOC-Rec: A Case Based Recommender System for MOOCs,” *Procedia - Soc. Behav. Sci.*, vol. 195, pp. 1813–1822, 2015.

[16] A. G. Zuquello, “OERecommender : an OER recommendation system for MOOC,” M.S. thesis, State University of Maringá, 2015 (in Portuguese).

[17] Y. Pang, Y. Jin, Y. Zhang, and T. Zhu, “Collaborative filtering recommendation for MOOC application,” *Comput. Appl. Eng. Educ.*, vol. 25, no. 1, pp. 120–128, 2017.