Linked Educational Online Courses to Provide Personalized Learning

Heitor Barros[1(✉)], Jonathas Magalhães[2], Társis Marinho[2], Marlos Silva[2], Michel Miranda[3], and Evandro Costa[3]

[1] Instituto Federal de Brasília - IFB, Via L2 Norte, SGAN 610 (610 Norte), Módulo D, E, F e G, Brasília, DF 70830-450, Brazil
heitor.barros@ifb.edu.br

[2] Federal University of Campina Grande - UFCG, Rua Aprígio Veloso, 885, Bairro Universitário, Campina Grande, Brazil
{jonathas,tarsis,marlos}@copin.ufcg.edu.br

[3] Federal Univeristy of Alagoas - UFAL, Av. Lourival de Melo Mota, Tabuleiro do Martins, Maceió, Brazil
{michel.miranda,evandro}@ic.ufal.br

Abstract. The emergence of MOOCs enabled students from around the world engage in courses taught by professors from leading universities. However, the relatively low completion rates of MOOC participants has been a central criticism in the popular discourse. Some studies point to up to 90% evasion in some courses. The lack of knowledge in relation to course prerequisites (background gaps) is one of the reasons that reduce the completion rate. To alleviate this problem, this paper proposes the use of a Linked Courses structure to provide support to students. In this proposal, before starting a course, the background gaps of each student are identified and a personalized set of support courses is recommended to help him. Results obtained so far indicate the effectiveness of this approach.

1 Introduction

Massive Open Online Courses or MOOCs are online courses accessible to anyone on the web. Hundreds of institutions have joined in an effort to make education more accessible by teaming up with MOOC providers [9]. With this new approach, inquisitive learners from all over the world can participate in the lectures of proven experts [5].

The relatively low completion rates of MOOC participants has been a central criticism in the popular discourse [6]. Some studies point to up to 90% evasion in some courses [5]. One of the reasons for these low completion rates is the static nature of the courses that do not fit the needs of students. Often a student starts a course without having the proper theoretical background. These **background gaps** make he did not perform well and not finish the course [5].

This problem is common in e-Learning domain and happens not only in MOOCs but in any Web-based Learning System. In this context, several

F. Castro et al. (Eds.): MICAI 2017, LNAI 10633, pp. 198–207, 2018.
https://doi.org/10.1007/978-3-030-02840-4_16

approaches have emerged in order to adapt the content of online courses according to user needs, creating techniques to provide personalized learning [2].

However, the construction of mechanisms that provide personalized learning involves activities such as predict possible needs of students during the execution of course and create educational resources for the different possibilities. Depending on the course, these activities can become very costly and make unfeasible the construction of the course.

Given this context, this paper presents an approach that links educational online courses aiming to use related courses as support resources for students. In other words, this approach identifies knowledge gaps of students and recommends courses that can help them overcome these gaps.

To evaluate our proposal, we conducted a preliminary study with a group of students trying to learn a course. This study showed that the proposed model is effective in detecting background gaps and recommending courses to help the students.

2 Related Work

The development of techniques and technologies to provide personalized learning has been one of the outstanding tasks in the Technology-Enhanced Learning (TEL) field. These work aims to identify the needs of students and guide them through the learning process.

Several techniques have been proposed, Chen [2] presents an approach for building personalized learning path guidance based on a **pre-test** that uses the incorrect responses of the learner to identify knowledge gaps. Lin et al. [7] proposes an approach based on the **data mining** technique of decision trees to provide personalized learning paths in a learning system focused on creativity skill.

Özpolat and Akar [8] proposed an automatic student modelling method that is based on a **clustering method** and **keyword mapping**. Fabio and Antonietti [4] presents an **intelligent agents** system with **machine learning** techniques to predict learners preferences or needs of students with attention deficit/hyperactivity disorder (ADHD).

Brinton et al. [1] proposes the Mobile Integrated and Individualized Course (MIIC). MIIC is a platform for personalized course delivery which integrates lecture videos, text, assessments, and social learning into a mobile native app. This approach collects **behavioural measurements** to update the learner model, which can in turn be used to determine the resources adaptation.

Henning et al. [5] presents an approach based on learning pathways and observation of learner behaviour to recommend the best resources for each step of the pathway.

Although there are techniques to enable personalized learning, one of the factors that hinder the widespread adoption of these techniques is the need to provide different paths and resources based on the characteristics of the students. In this context, our proposal has the advantage of the reuse of educational resources present in other online courses, facilitating the creation of personalized learning structure.

3 Linked Courses

Linked Courses structure aims to organize educational resources to facilitate their reuse. It takes into consideration the context in which these resources can be used within a course/discipline. In addition, Linked Courses structure allows to identify the required knowledge for a course, and also which courses can be used to support that required knowledge.

In the Sect. 3.1, we define the representation of a course structure through an Integration Ontology. In the Sect. 3.2, we present how different courses are linked, focusing on the precedence issues among courses. To present our approach we use some examples based on two online courses: Introduction to Programming in Java[1] and Java Web Programming[2].

3.1 Course Structure

As said before, the Integration Ontology is responsible for linking different courses structures. We define the course structure based on a model for knowledge representation. The Integration Ontology uses a strategy of **Course Models**. Course Models are used in steps of classification, structuring and alignment of educational resources. Besides, a Course Model specify the pedagogical units of a discipline in a structured way (*Hierarchical* and *Sequencing* structures). Hierarchical structure defines the notion of topics and subtopics and Sequencing structure determines the order in which the resources should be used in a course.

In the following, we present the definitions of the integration approach based on Course Models:

D1 $\rightarrow$ A Course Model M_a is composed by a set of Pedagogical Units (Topics) U_a and it is related to a *Knowledge Domain* D_x.

D2 $\rightarrow$ The set U_a is given by a **Domain Specialist** in the step construction of the model M_a. He determines the pedagogical units according to his vision of how to structure an educational course in the domain D_x.

D3: Sequencing Structure $\rightarrow \forall u_j \in U_a$ there is a set $N_j \subset U_a \setminus \{u_j\}$ that contains the *sequent* pedagogical units with respect to u_j. In the same way, u_j is a *previous* unit to a unit $u_k \in N_j$.

D4: Hierarchical Structure $\rightarrow \forall u_i \in U_a$ there is a set $S_i \subset U_a \setminus \{u_i\}$ that contains the pedagogical units that are *specializations* or *sub-topics* of u_i. In the same way, u_i is a *generalization* or *super topic* of a unit $u_k \in S_i$.

[1] Available at: http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-092-introduction-to-programming-in-java-january-iap-2010/.

[2] Available at: https://www.virtualpairprogrammers.com/training-courses/Java-Web-Development-training.html.

D5 → To insert an **Educational Resource** to the model, the resource should be related to one or more Pedagogical Units. This way, it is defined the property $l(R, U)$, where R is a set of Educational Resources and U is set of Pedagogical Units. Given $r_e \in R$ and $u_i \in U$, so $l(r_e, u_i)$ indicates that the resource r_e is related to the pedagogical unit u_i.

Following the presented definitions, the Listing 1.1 presents an OWL code snippet showing an instance of a Model Course related to the Java programming domain.

Listing 1.1. Course Model based on a Java programming presented in the Manchester OWL pattern..

```
1 Individual: <onto.owl#model_Java>
2
3   Types:
4     <onto.owl#CourseModel>
5
6   Facts:
7 <onto.owl#belongsTo> <onto.owl#Java_Programming>,
8 <onto.owl#hasTopic> <onto.owl#Variables_types>,
9 <onto.owl#hasTopic> <onto.owl#Operators>,
10 <onto.owl#hasTopic> <onto.owl#Methods>,
11 <onto.owl#hasTopic> <onto.owl#Loops_and_arrays>,
12 <onto.owl#hasTopic> <onto.owl#Access_control>,
13 <onto.owl#hasTopic> <onto.owl#Class_scope>,
14 <onto.owl#hasTopic> <onto.owl#Packages_and_Java_API>,
15 <onto.owl#hasTopic> <onto.owl#Inheritance>,
16 <onto.owl#hasTopic> <onto.owl#Interfaces>,
17 <onto.owl#hasTopic> <onto.owl#Exceptions>,
18 <onto.owl#hasTopic> <onto.owl#File_IO>
```

In Listing 1.1, line 1 defines the individual *model_Java*, line 4 defines that this individual is Course Model, line 7 uses the property *belongsTo* to relate this Course Model to the *Java_Programming* **Knowledge Domain**. Finally, lines 8 to 18 have the property *hasTopic* that defines the topics of this Course Model.

In Listing 1.2 the code of Topic *Exceptions* is presented. Line 4 defines the type of this individual. In line 7 this topic is linked with the *model_java* Course Model using the property *belongsToModel*. Line 8 uses the property *nextTopic* to define that the topic **File_IO** comes after the topic *Exceptions* in this Course Model.

Listing 1.2. OWL individual of Topic *Exceptions* that belongs to Course Model *model_Java*.

```
1 Individual: <onto.owl#Exceptions>
2
3     Types:
4         <onto.owl#Topic>
```

```

    Facts:
     <onto.owl#belongsToModel> <onto.owl#model_Java>,
     <onto.owl#nextTopic> <onto.owl#File_IO>
```

3.2 Course Mapping

Our model also provides mapping properties to align common points in two distinct Course Models.

D6 $\rightarrow$ Given U_a and U_b the sets of pedagogical unit of Models M_a and M_b, respectively. A Course Model M_a can be aligned to model M_b, using the following properties:

1. **Equivalence:** An unit $u_i \in U_a$ can be *equivalent* to a unit $u_m \in U_b$. It is defined the property $e(U, U)$, where $e(u_i, u_m)$ are equivalent units. This property is reflexive.
2. **Generalization:** An unit $u_i \in U_a$ can be *generalization* of a unit $u_n \in U_b$. It is defined the property $g(U, U)$, where $g(u_i, u_n)$ indicates that u_i is a *generalization* of u_n and u_n is a *specialization* of u_i.
3. **Precedence:** An unit $u_j \in U_a$ can be a *precedent unit* of $u_o \in U_b$. It is defined the property $s(U, U)$, where $s(u_j, u_o)$ indicates that u_j is a precedent unit of u_o and u_o is a sequent unit of u_j.

For example, the Listing 1.3 shows a fragment of a individual of Course Model that represents a course of Java Web Development. This Course Model belongs to *Java_Web_Development* domain.

Listing 1.3. Course Model of a Java Web Development course presented in Manchester OWL format.

```
Individual: <onto.owl#model_Java_Web>

  Types:
    <onto.owl#CourseModel>

  Facts:
<onto.owl#belongsTo> <onto.owl#Java_Web_Development>,
<onto.owl#hasTopic> <onto.owl#Servlets>,
<onto.owl#hasTopic> <onto.owl#Web_Application_Deployment>,
<onto.owl#hasTopic> <onto.owl#Handling_Forms>,
<onto.owl#hasTopic> <onto.owl#GET_and_POST>,
<onto.owl#hasTopic> <onto.owl#Session_Management>,
<onto.owl#hasTopic> <onto.owl#Java_Server_Pages>,
<onto.owl#hasTopic> <onto.owl#Model_View_Control>,
<onto.owl#hasTopic> <onto.owl#Java_Standard_Tag_Library>,
<onto.owl#hasTopic> <onto.owl#Frameworks_and_Struts>
```

In this context, the student must have knowledge of Java language so that he can start the Java Web course. For this, it is necessary to use the properties **precedes** to link these courses.

Listing 1.4 shows the Topic *File_IO*. This topic is the last of Course Model *model_Java* and it uses the property **precedes** to link this model to the topic *Servlets*, first topic of *model_Java_Web*. Thus, the two course models are connected via *precedes* property.

Listing 1.4. OWL individual of Topic *File_IO* that belongs to Course Model *model_Java*.

```
Individual: <onto.owl#File_IO>

    Types:
        <onto.owl#Topic>

    Facts:
     <onto.owl#belongsToModel> <onto.owl#model_Java>,
     <onto.owl#precedes> <onto.owl#Servlets>
```

4 Personalized Course Planning

Figure 1 illustrates the overview of personalized course planning. This process receive as inputs a learner, a linked courses graph and a goal course. The linked courses graph contains structured courses based on model defined in Sect. 3 and the goal course is the course that the student wants to study. The steps of this process are discussed below:

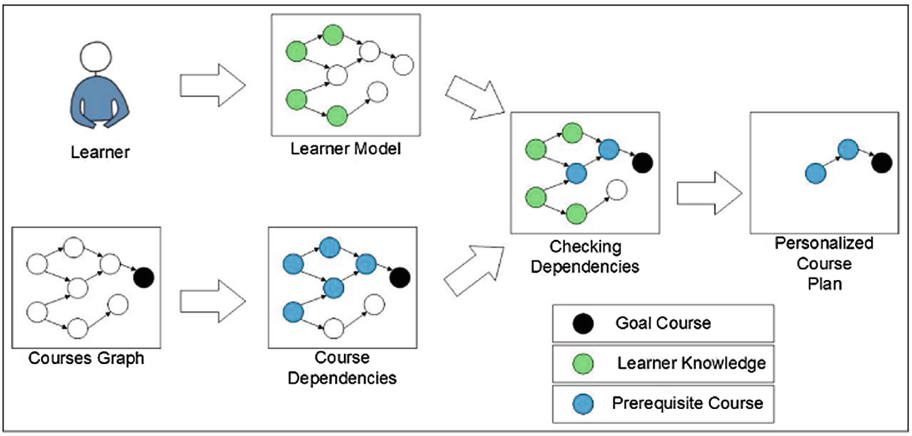

Fig. 1. Overview of personalized course planning

Create Learner Model - this step is responsible for create a Learner Model to describe the knowledge of the leaner. This process is described in Sect. 4.1.

Define Course Dependencies - this sub-process identifies the *goal course* prerequisites based on precedence links of Courses Graph.

Check Learner Dependencies - This step requires the Leaner Model and the prerequisites of *goal course*. In this sub-process the previous knowledge of the learner is compared with the *goal courses* dependencies. The purpose is to determine the dependencies that the learner has not learned.

Create Course Plan - the personalized course plan is based on the goal course dependencies that are not in the Learner Model.

4.1 Learner Model

We define a student model to represent the student's knowledge, using this model we can identify the student's background gaps in a determined course. We use Bayesian Networks (BN) to represent the student model, we follow a similar approach as proposed in [3]. The BN is constructed taking into account the hierarchical structure of Pedagogical Units presented in Subsect. 3.1. So, the BN contains just one type of node, a *Pedagogical Unit (PU)* that represents a skill of the learner. This node can be parent of a set of other *Pedagogical Unit* nodes. Every node has two values, mastered or not mastered, that measure the learner's knowledge.

The weights of the Bayesian Network are given by the Knowledge Engineer (responsible for modeling the domain knowledge). To set the prior probabilities, we established five degrees of knowledge: *No idea* = 0.05, *Basic* = 0.25, *Good* = 0.50, *Very good* = 0.75 and *Expert* = 0.95. To set the table of conditional probabilities, we follow the approach proposed by Zapata-Rivera and Greer [10].

We define two scenarios in which are collected evidence about the student to update the student model. The first scenario is through questionnaire to identify if the student has the required background to start the course. For example, suppose that the student pretends to learn the Spring framework, then the student would have to respond to a questionnaire containing questions about the required background, i.e., Java language and Java Web. So, the node can be calibrated using the following equation: $\#correct\_questions/\#questions$. The second scenario to update the student model occurs when he finishes a course. Her final course grade is used as calibration to the correspondent node in her student model. For example, if the student takes a grade of 70/100 in the Java course, this node weight will be calibrated to 0.7.

5 Evaluation

To evaluate the proposal of this paper, we developed a running example to demonstrate the effectiveness of our approach. Initially, this Section describes the scenario of the running example, i.e. the courses and learners. Then, how our approach can provide recommendations to the students to help them in the process of learning. In this running example, we use the following courses:

C1 - Java Programming (presented in Sect. 3).
C2 - Java Web Development (presented in Sect. 3).
C3 - Spring MVC Framework[3] (http://spring.io).

The Spring Framework is a widely-used MVC framework that provides a comprehensive programming and configuration model for modern Java-based enterprise applications. Listing 1.5 shows the Course Model code of Spring course. The Course Models of the other courses were presented in Sect. 3 (Listings 1.1 and 1.3).

Listing 1.5. Course Model of a Spring MVC Framework course presented in Manchester Owl format.

```
Individual: <onto.owl#model_Spring>

  Types:
    <onto.owl#CourseModel>

  Facts:
<onto.owl#belongsTo> <onto.owl#Spring_Framework>,
<onto.owl#hasTopic> <onto.owl#Getting_Started>,
<onto.owl#hasTopic> <onto.owl#Autowiring>,
<onto.owl#hasTopic> <onto.owl#Wiring_with_Annotations>,
<onto.owl#hasTopic> <onto.owl#SPEL_Expression_Language>,
<onto.owl#hasTopic> <onto.owl#Web_Application_Basics>,
<onto.owl#hasTopic> <onto.owl#Web_Forms>,
<onto.owl#hasTopic> <onto.owl#Spring_Security>,
<onto.owl#hasTopic> <onto.owl#Logging_and_Testing>,
<onto.owl#hasTopic> <onto.owl#Hibernate>,
<onto.owl#hasTopic> <onto.owl#Spring_Webflow>,
<onto.owl#hasTopic> <onto.owl#JSON_and_AJAX>
```

In this scenario, a group of 14 students are involved in a project that uses the Spring Framework for application development. In this context, the chosen *goal course* is the Spring course (C3).

This is a heterogeneous group of students, some are undergraduate students and others are master's students, so they have different knowledge background. For this reason, we need to evaluate the individual knowledge of each student to identify potential knowledge gaps. Students that already known Spring Framework did not participate of this study.

First, it is necessary to identify Knowledge prerequisites of Spring course(C3). The development of applications with Spring Framework (C3) requires knowledge as Servlets and Java Server Pages that are topics of Java Web course (C2). Similarly, as shown in Sect. 3, Java Web course (C2) is dependent on the content of Java programming course (C1). This way, there is a dependence chain among these courses.

In this scenario, the student must have sufficient knowledge of the Java language and Java Web development to start the Spring course. Otherwise, the

[3] Available at: https://www.udemy.com/javaspring/.

student will start the course with background gaps that can result in poor performance in the course.

In other words, we need to to create the Learner Model of these students, as specified in Sect. 4.1. Thus, each student answered a questionnaire aiming to measure his knowledge related to background courses (C1 and C2). It is not possible that a student has knowledge of Java Web (C2) and he does not know Java (C1), whereas C1 is a prerequisite of C2.

The questionnaire classified the students into three groups as shown in Table 1. As expected, no group was empty. This proves the heterogeneity of the group.

Table 1. Students knowledge background and courses recommendation

Group	Student knows Java programming?	Student knows Java web development?	Courses recommendation	Number of students
1	No	No	$C1 \rightarrow C2 \rightarrow C3$	3
2	Yes	No	$C2 \rightarrow C3$	5
3	Yes	Yes	$C3$	6

After running this case study we got the following conclusions:

– Students who participated in the case study showed the characteristic heterogeneity MOOCs students, even though a scenario on a much smaller scale.
– Given the goal of learning Spring, students were informed about the existence of knowledge gaps and what courses they should do. Without the use of this approach, students would start the course with knowledge gaps or would have to seek information and/or courses on their own.
– The proposed model was able to represent and connect the three courses. In addition this model identified the precedence order between them.
– The construction of this scenario not demanded much effort since the courses and educational resources were available on the web. The work came down to select these courses and use the Integration Ontology to annotate and connect these courses. This demonstrates that this approach encourages the reuse of educational resources.

6 Conclusion and Future Work

The emergence of MOOCs enabled students from around the world engage in courses taught by professors from leading universities in the world. This scenario increases the need for techniques that adapt these courses to students, providing personalized learning.

In this paper, we propose the use of a linked courses structure to identify knowledge gaps of students and use these courses to help students overcome these gaps. The main advantages of this approach are: (i) prevent students starting courses with background gaps, (ii) guide students so that they overcome these gaps and (iii) facilitate the reuse of educational resources.

To evaluate our proposal, we conducted a preliminary study that proved the effectiveness of the proposal in connecting related courses and use them to provide a personalized course plan for each student, based on his knowledge background.

As future work, we plan to evaluate the proposed model in a scenario with a higher number of students. Also, we will enhance the Learner Model and the course structure to enable the recommendation of a set of topics of a course for a student, not just a full course, in order to provide more accurate recommendations.

References

1. Brinton, C., Rill, R., Ha, S., Chiang, M., Smith, R., Ju, W.: Individualization for education at scale: MIIC design and preliminaryevaluation. IEEE Trans. Learn. Technol. **PP**(99), 1 (2014)
2. Chen, C.-M.: Intelligent web-based learning system with personalized learning path guidance. Comput. Educ. **51**(2), 787–814 (2008)
3. Costa, E., Silva, P., Magalhães, J., Silva, M.: An open and inspectable learner modelingwith a negotiation mechanism to solve cognitive conflicts in an intelligent tutoring system. In: Proceedings of the 2nd Workshop on Personalization Approaches for Learning Environments (PALE 2012). CEUR Workshop Proceedings, vol. 872. CEUR-WS.org (2012)
4. Fabio, R.A., Antonietti, A.: Effects of hypermedia instruction on declarative, conditional and procedural knowledge in ADHD students. Res. Dev. Disabil. **33**(6), 2028–2039 (2012)
5. Henning, P.A., et al.: Personalized web learning: merging open educational resources into adaptive courses for higher education. Personal. Approach. Learn. Environ. **55**, 55–62 (2014). ISSN: 1613-0073
6. Kizilcec, R.F., Piech, C., Schneider, E.: Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In: Proceedings of the Third International Conference on Learning Analytics and Knowledge, pp. 170–179. ACM (2013)
7. Lin, C.F., Yeh, Y.-C., Hung, Y.H., Chang, R.I.: Data mining for providing a personalized learning path in creativity: an application of decision trees. Comput. Educ. **68**, 199–210 (2013)
8. Ozpolat, E., Akar, G.B.: Automatic detection of learning styles for an e-learning system. Comput. Educ. **53**(2), 355–367 (2009)
9. Pappano, L.: The Rise of MOOCs. The New York Times Magazine, September 2013
10. Zapata-Rivera, J.-D., Greer, J.E.: Interacting with inspectable Bayesian student models. Int. J. Artif. Intell. Educ. **14**(2), 127–163 (2004)