

Agent-Based Conceptual Framework for Collaborative Educational Resources Adaptation in Virtual Learning Environments

Vitor Bremgartner^{1,2}, José Francisco M. Netto¹, Crediné Menezes³

¹Instituto de Computação (IComp) – Universidade Federal do Amazonas (UFAM)
Av. Gen. Rodrigo Otávio, Coroado. Manaus – AM – Brasil

²Instituto Federal do Amazonas (IFAM) – Campus Manaus Distrito Industrial
Av. Danilo Areosa, Distrito Industrial. Manaus – AM – Brasil

³Faculdade de Educação – Universidade Federal do Rio Grande do Sul (UFRGS)
Av. Paulo Gama, Farroupilha. Porto Alegre – RS – Brasil

{vitorbref, jnetto}@icomp.ufam.edu.br, credine@gmail.com

Abstract. *Frequently, the existing resources in Virtual Learning Environments (VLEs), used in distance education courses and blended, are presented in the same way for all students. So, the approach adopted in this paper in order to solve this problem is based on a framework called ArCARE (Conceptual Framework of Educational Resources Adaptation in Virtual Learning Environments), which allows adaptation of resources for students in VLEs. It aims the construction of students' knowledge, using a multi-agent system technology that handles an open learner model ontology. These ArCARE resources are recommendation and adaptation of collaborative activities such as Pedagogical Architectures for the students have a more effective learning and increase in skills levels of a particular course. Results obtained in a flexible curriculum course of Computational Thinking show the feasibility of the proposal.*

1. Introduction

In the teaching-learning processes, is increasingly common the use of educational environments known as Virtual Learning Environments (VLEs). These environments support the process of communication between students, teachers, and the community, allowing everyone to participate in an interactive mode and with availability of teaching materials. In addition, there are VLEs that use Artificial Intelligence (AI) technologies, especially regarding the possibility of flexible teaching-learning processes to students, in which the learning environment is able to adapt its resources presented according to the student's needs [Bremgartner, Netto, and Menezes 2017]. In this paper, we considered the term “adaptation resources in VLEs” not only the action of changing resources (e. g., activities, school assignment, learning objects delivered), but it is how the resources will be arranged in a VLE so as to allow changes in the pedagogical organization of the learning according to the students' characteristics using technologies. The adaptation process makes use of a learner model [Bremgartner, Netto, and Menezes 2017]. This model is a record of the students' actions as well as useful information about the student profile in the VLE.

However, despite the increasing use of educational environments, they usually offer learning resources in the same way for all students (*one-size-fits-all* form), resulting that the learning cannot become effective for all because the course in the VLE does not fit according to the several characteristics that each student has and the history of his actions within the environment. This therefore creates difficulties of knowledge acquisition for some students or even lack of interest by the students in the use of learning environment. There are several

techniques for resources adaptation for students in literature as conversational systems [Latham, Crockett, and McLean 2013], group support [Oliveira and Tedesco 2009], selection of learning paths [Sarmiento *et al.* 2016] and all of them have significant technological innovations, but there is a need for improvements regarding collaborative learning that could promote the student motivation, engagement and effective learning according to his profile. A student centric approach is needed in order to retain students and according to the educational theory, learning motivation is increased by personalization and when learning motivation increases, learning effectiveness increases.

Thus, the approach adopted as a proposal to solving this problem is based on a framework called ArCARE (Conceptual Framework of Educational Resources Adaptation in VLEs), being a strategy that allows adaptation of resources for students during the course, based on Piaget Constructivism [Piaget and Inhelder 1969]. This framework uses multi-agent system technology that handles a learner model ontology which consists of students' characteristics, such as interests, competencies, skills, history of student performance in activities, frequency, and learning styles. The adaptation provided is the recommendation of adjusted resources based on collaborative learning, for example, Pedagogical Architectures (PAs) [Tavares, Menezes, and Nevado 2012] containing proposals for collaborative activities in order to student have a more effective learning of a particular course. Also, the mechanisms of resources adaptation provided by ArCARE can be used in both traditional courses (formal, defined curricula) and in flexible curricula (where the students can choose which activities they will do together with their colleagues) courses. So, traditional courses may be replaced by a more flexible curriculum through the use of Learning Units (LUs). A LU can be seen as a topic within a course. The learner model is also presented to the student, being an Open Learner Model (OLM). OLM refers to making a student's learner model explicit, externalizing the learner model contents to the learner, so as to provide an additional resource through self-awareness and possible self-regulation of the learning process that is believed to enhance learning and learner autonomy [Bull and Kay 2016].

Besides this Introduction, this paper is structured as follows: Section 2 discusses about adaptation and collaborative practices in VLEs. Section 3 presents the ArCARE architecture, also describing the multi-agent system and OLM. Section 4 reports tests of adaptations of PAs-based resources in a Computational Thinking course as a case study. Section 5 presents the conclusions.

2. Adaptive and Collaborative Practices in VLEs: Related Research

Recent theoretical underpinnings of successful Computer-Supported Collaborative Learning (CSCL) have suggested that for collaborative learning to be effective, students must explicate their thoughts, actively participate, discuss and negotiate their views with the other students in their team, coordinate and metacognitively regulate their actions between them [Järvelä and Hadwin 2013], and share responsibility for the learning process. This work deals with adaptive and collaborative resources presented to students, especially, PAs to promote collaborative learning between students, using software agents. PAs can be defined as the construction of pedagogical strategies that is based on a certain theory and its assumptions in order to assist in the effectiveness of learning mediated by digital technologies of communication and information as Virtual Learning Environments and web conferencing tools [Tavares, Menezes, and Nevado 2012].

The approach of adaptive and collaborative learning in VLEs has been an alternative used to support the educational processes mediated by technology, as we can see in related works. In [Santos, Castro, and Menezes 2012] is presented a platform designed under a paradigm called MORFEu for designing PAs-based virtual environments, with flexibility to be combined, runtime changeable without loss of data. The Oscar Conversational Intelligent Tutor System (CITS) in [Latham, Crockett, and McLean 2013] is an ITS which uses a natural

language interface to enable learners to construct their own knowledge through discussions. Oscar CITS aims to mimic a human tutor by dynamically detecting and adapting to an individual's learning styles whilst directing the conversational tutorial. As way to overcome the general feeling of isolation and consequent high dropout of students in VLEs, the i-Collaboration model in [Oliveira and Tedesco 2009] presents the results of an experiment with the model that promotes collaboration between users in a VLE. i-Collaboration is based on the use of Virtual Learning Companions (VLC) agents as collaboration monitors based on constructivist theory. The VLC agents are integrated with collaborative tools of VLE and know each student profile and his behavior in the learning environment. In [Eryilmaz *et al.* 2013], CSCL and constructivism are used. Such research has a proposal of a theoretical framework that leverages attention guidance in a social approach to facilitating the process of central domain concepts, principles, and interrelations between them based on social interactions. In [Dascalu *et al.* 2013], the authors developed an educational collaborative filtering recommender agent, with an integrated learning style finder. The agent produces two types of recommendations: suggestions and shortcuts for learning materials and learning tools, helping the learner to better navigate through educational resources. Using OLM, the work in [Hosseini *et al.* 2015] has explored the idea of combining social guidance with traditional knowledge-based guidance systems in hopes of supporting more optimal content navigation. The authors proposed a greedy sequencing approach aimed at maximizing each student's level of knowledge and implemented it in the context of an open social student modeling (OSSM) interface. Also, in [Sarmiento *et al.* 2016] is presented the design of a semi-automated Academic Tutor to support students in selecting learning paths (that consist of a set of courses which form the individual curricula) to achieve a particular professional profile, using ontologies.

Thus, we can see that there are several works in the literature that deal with resources adaptation, collaborative learning, multi-agent system, or learner model applied in VLEs. The purpose of ArCARE is to contribute when considering all these characteristics together and several data of students at the same time (skills, interests, learning styles), varying according to the history of students' interactions within the VLE and also presenting the OLM to the student. Furthermore, at the same time, ArCARE provides adaptation and recommendation of collaborative resources in the VLE for students, and allows a flexible curriculum in the courses presented. So, the main contribution of ArCARE is allowing changes in the pedagogical organization of the learning according to the students' characteristics using multi-agent and OLM-based ontology technologies. The next section describes our proposal in more details.

3. Defining ArCARE

The ArCARE architecture is shown in Figure 1. In this model, is shown the framework for adaptation and recommendation of educational resources in VLEs, depending on the characteristics of the student. This architecture is composed of three fundamental components: resources adaptation module, VLE database, and user action area.

In this framework, we believe that users (students, teachers) are always in interaction with the VLE, in the user action area in (1). The student accesses resources, updates his registration data, performs activities proposed by teacher and accesses his OLM in order to know his performance throughout the course in order to make self-reflections. The teacher can prepare courses, LUs, stored in the VLE database in (6), activities, post grades, insert resources in the repository (8) of the VLE, and perform other actions according to his assignments. In our conceptual framework we defined that the student needs to have his initial profile, i.e., data that compose his initial model, aiming the environment begin to be adapted, instead of the VLE wait for several students' interactions. For this purpose, the first interactions of students with the VLE are registered (2). In addition, his usage history of the environment is obtained (9). To achieve this, data mining techniques are used within the learner model agents (4). With this information, the OLM begins to be formed (3), which is updated in every student interaction

within the VLE by the learner model agents (4). To handle the students' data, these agents use an ontology that describes the model, the OLM ontology (5), which contains rules for message exchanging between agents, as well as definitions and rules that are part of the learner model. In addition, in the OLM, we have the *resource model*, which, in general, consists of the most relevant data regarding the resources to be used in the adaptation process in the VLE (e.g., skills associated to a resource). In turn, resource adaptation agents (7) allow the adaptation and selection of resources that are stored in the VLE database (8) that they consider most appropriate for each student in the course using student data (3) and the OLM ontology (5). Finally, these resources are presented for students (8) use them in the VLE (1). It is noticed that this resource recommendation process can be seen as a continuous process, since new resources are showed each time the learner model is updated.

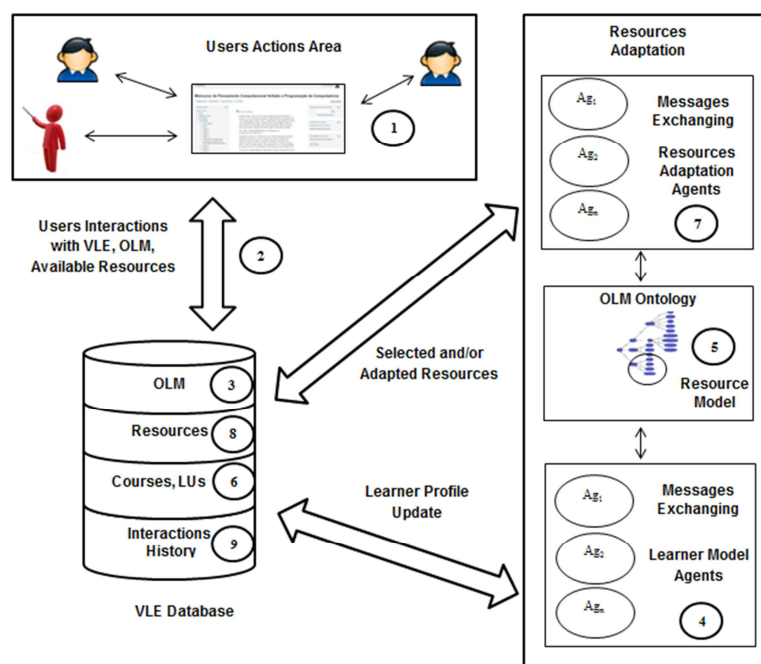


Figure 1. ArCARE Architecture [Bremgartner, Netto and Menezes 2017].

As Figure 1 shows, there are two types of agents: learner model and adaptation resources agents. The first type of agents were developed in the JADE [JADE 2017] and the latter in JADEX [JADEX 2017] frameworks. The learner model agents handle the students' data and their learner models in the VLE database. The learner model is described by its ontology, which is also useful for the correct handling of message exchanging between agents. In turn, the adaptation resource agents, with the data obtained from the learner model and using the Beliefs-Desires-Intentions (BDI) model [Georgeff *et al.* 1999], select resources contained in the repository (which is in the VLE database), to adapt the VLE. Finally, the student accesses VLE with its adapted content, and he can access his OLM. The agents who are responsible for handling the learner model data are: *Initial Profile Agent* (that sets the initial students' skills and learning styles based on the students' initial manifestations and the historical data of students' actions in the VLE); *Learning Assessment Agent* (that evaluates the activities answered by students in the VLE); *Update Profile Agent* (that updates the learner model data by students' interactions with the environment, through the information coming from Initial Profile Agent and Learning Assessment Agent. This agent also reviews past resource adaptations and whether these were really useful for improving students' skill levels. With this result, resource utility levels are updated).

The resources adaptation agents are: *Profile Situation Assessment Agent* (responsible for verifying and assessing the current situation of the student in the environment, taking decisions to aid student learning. This agent can search for students' questions and mistakes in answered activities, analyze the students' skills levels, and map these students' data into the learning resources needed to be made available for each student in VLE); *Resource Adapter Agent* (that obtains information from Profile Situation Assessment Agent and executes the process of showing in the VLE resources for each student).

Each agent has functions that they perform. The *Initial Profile Agent and Learning Assessment Agent functions* are: $get(\langle learning_styles \rangle, \langle interests \rangle)$, which collects data about learning styles and students' interests, $calculates(h_1a_1, h_1a_2, \dots, h_1a_m, h_2a_1, h_2a_2, \dots, h_p a_m, k_1, k_2, \dots, k_m, d_1, d_2, \dots, d_m, t_1, t_2, \dots, t_m, c_1, c_2, \dots, c_m, f_1, f_2, \dots, f_m, g_1, g_2, \dots, g_m)$, which calculates the levels of students' skills associated with the activities or tests they answered, in addition to their history of past actions in the VLE, where: $h_p a_m$: is an h_p skill associated with an a_m activity, where we can have p skills for each m activity; and for each student skill h_x , $1 \leq x \leq p$, is calculated the result obtained in each of the m activities associated with p skills; f_i is a normalization factor for each activity i , $1 \leq i \leq m$, that allows $0 \leq h_x \leq 10$; g_i is the student's score obtained in activity i ; c_i is a weight value for each of the m activities, obeying a criterion regarding errors and correctness of the question. For example, c_i can be +1, if the student hits the question completely or -1, if he misses; d_i : difficulty of the question, specified by the teacher; t_i : it is a weight factor according to the time interval the student answered the question; k_i : indicates the number of attempts that the student had on the question until finalizing it, with $k_i \neq 0$. If the student has not made any attempt, $h_x = 0$. In this work, we considered that the skills' levels of students, resources and activities vary on a scale of 0 to 10. So, for each h_x , $1 \leq x \leq p$, and each activity a_i , $1 \leq i \leq m$, we have in Equation 1

$$\sum_{i=1}^m (f_i \cdot g_i \cdot c_i \cdot h_i a_i \cdot d_i \cdot t_i \cdot \frac{1}{k_i}) = h_x \quad \text{Equation 1}$$

The *Update Profile Agent* stores in the VLE database the elements derived from the *Initial Profile Agent* and *Learning Assessment Agent*, and has the functions described below. After each adaptation done, it is checked whether or not the student has accessed such a resource, through the function *accessed* (*student, resource, course, time, resource_request*), where *resource_request* indicates how many times the student has tried to see the resources available to him in VLE, in which this is to indicate whether he is a student who is applying for help in the VLE or not. The function *feedback_satisfied* (*student, resource, course, level_of_satisfaction*) stores whether or not the student was satisfied with the use of the resource (through a simple question to the student in the VLE using a Likert Scale 1-very useless to 5-very useful). So, each resource will have a record of uses, and its successes or failures will be recorded when recommended for the student. A resource presented to a student who has failed after its use is not recommended again. In this case, other resources are presented to this student.

The *Profile Situation Assessment Agent function* is *checks_and_combines*($c, f, \langle learning_styles \rangle, h_1a_1, h_1a_2, \dots, h_1a_m, h_2a_1, h_2a_2, \dots, h_p a_m, hc_1, hc_2, \dots, hc_q, h_1r_1, h_1r_2, \dots, h_1r_r, h_2r_1, h_2r_2, \dots, h_s r_r, ea_1r_1, ea_1r_2, \dots, ea_1r_r, ea_2r_1, ea_2r_2, \dots, ea_n r_r, \langle interests \rangle, i_1r_1, i_1r_2, \dots, i_y r_r, ur_1, ur_2, ur_r$), in which for each candidate resource to be adapted or recommended r , its levels of skills, learning styles and interests are compared with the respective levels of skills, learning styles and interests associated with the students, where: c is the course that the student is enrolled; f is the student's frequency; $h_p a_m$: is the h_p skill associated with an a_m activity, in which we can have p skills for each of the m activities; hc_q is the q -th skill belonging to the course or discipline in which the student is enrolled; $h_s r_r$: is the h_s skill associated with a r_r resource, in which we can have s skills for each of the r resources; $ea_n r_r$: is the ea_n learning style associated with an r_r resource, in which we can have n learning styles for each of the r resources; $i_y r_r$: is the y -th interest associated with a resource r ; ur_r : is the r -th utility level associated with a resource r .

The *Resource Adapter Agent function* is *executes_strategy*($c, e, ta, tr, r_1, r_2, \dots, r_r$), where c is the course that the student is enrolled in; e is the student; ta is the pedagogical strategy applied in the resource, which can be a recommendation or an adjustment (adaptation) in the resource already presented to the student, where it depends on each resource and the purpose that it must execute in VLE; r is each resource associated with the course or discipline c ; tr is the duration of the resource that will be available to the student.

In this work, we adopted the heuristic that a resource is considered *better* (in order for the students to reach higher levels of skills) when it obeys the following criteria, in this order: 1) Resources in which occurs $\text{Min}(h_{i,r_j} - h_i)$, $1 \leq i \leq s$, $1 \leq j \leq r$, $h_{i,r_j} - h_i > 0$, that is, the smallest positive difference between the resources (h_{i,r_j}) and the students (h_i) skills levels; 2) Resources where occurs $\text{Min}(\text{Abs}(ea_{i,r_j} - ea_i))$, $1 \leq i \leq n$, $1 \leq j \leq r$, that is, the smallest module of the difference between students and resources levels of learning styles; 3) Highest level of utility according to its recommendations history, evaluated by f_{adapt} as follows: the agents select a particular resource based on the past recommendations of that resource; the Apriori algorithm is applied, taking as variables the student's feedback after using the resource, student performance results after using that resource, and utility level of that resource. With all this information, the history of actions in the VLE is set up, either by the students' interactions, or by the utilities of the resources. Patterns of performance in activities and resource utilization are detected in order to know how useful a resource was and whether it helped the student to solve his or her problems in the course.

Another important step of this work was to build the ontology that represents the OLM. The OLM ontology was built in the Protégé [Protégé 2017] editor. For the learner model we used the IMS LIP standard [LIP 2017], being a standard commonly used nowadays that allows extensions. Originally, LIP has 11 categories (or classes), but only five were used (*Activity*, *Competency*, *Identification*, *Accessibility*, and *Interest*). Furthermore, although using LIP categories, this standard has been extended to the needs of this project, by the addition of 2 categories: *Learning_Styles* and *Frequency*. The *Frequency* category describes the frequency of students in VLE. The *Learning_Styles* category contains information about the learning styles of the students. In the *Activity* category there is information about the activities proposed by the teacher. The activities in this work are based in PAs, and they have metadata to document them using the LTSC/IEEE Learning Object Metadata (LOM) standard [LOM 2017]. These metadata are useful for mapping between the fields of the LOM-based Pedagogical Architectures and students' profiles, in order to the recommended PAs for students to be more accurate. The learner model ontology in *Competency* class describes the skill level of each student, which is useful for identifying the students and their levels of competencies and skills in a discipline or course. In this work, each student's competence is composed of a set of specific skills. Other categories used are *Identification* (for the student personal data), *Accessibility* (accessibility data of user, credentials in the e-learning system) and *Interest* (containing the students' interests).

4. Applying the Framework in a Classroom

As case study of our work, we used ArCARE on a flexible curriculum course of Computational Thinking (CT) offered in blended form using the Moodle VLE [Moodle 2017]. Curricular flexibility means that every student has the option of choosing a set of Learning Units to develop a particular professional profile [Sarmiento *et al.* 2016]. According to [Wing 2006], CT is a process of solving problems that includes a number of features and arrangements. CT is essential for the development of computer applications, but can also be used to support problem solving in other subjects. Initially, it was assigned an activity in a pre-test format in order to get the diagnosed skills of the students. This CT course was held with 33 Higher education students in Mechatronics of Federal Institute of Amazonas, Campus Manaus Distrito Industrial (IFAM-CMDI). In the practice adopted in this work, the students must solve proposed problems reflecting on what are the right actions to do and which programming structures are needed.

And when an inadequacy arises, students can test and debug a new procedure through discussion with their colleagues, searching for a better result. Both in the pre-test and in the post-test the skills involved were the same ones tested, but obviously with different questions. The tests were extracted from the AP CollegeBoard Computer Science Principles – Effective Fall 2016 [AP CollegeBoard 2016]. The skills involved were those classified by Google Computational Thinking Course [Google 2017] in partnership with International Society for Technology in Education (ISTE) and Computer Science Teachers Association (CSTA) [ISTE/CSTA 2017], which are: domain in designing algorithms and programming, abstraction, problem decomposition, simulation, pattern recognition and data analysis. In addition, two other skills were evaluated so that we could have more accurate information on student performance: domain in condition and repetition structures. In the VLE were created 3 questionnaires for students answer at the beginning of the course, with the aim of obtaining the initial learner model. The questionnaires are: 1) Index of Learning Styles, seeking to know the Felder-Silverman learning styles [Soloman and Felder 2017] of students; 2) Honey-Alonso Learning Styles Questionnaire [Honey and Mumford 2017]; 3) The pre-test itself, as mentioned previously.

In this course the LUs were composed of Mandatory Questionnaires, Mandatory Learning Units, Optional LUs, and Recommended Pedagogical Architectures (PAs), which are recommended or adapted resources for students. The Mandatory Questionnaires consist of questions related to student learning styles. The Mandatory LUs are the predefined units in the course, which the student must perform, consisting of the mandatory part of the course, although the student could choose the order in which he would study them. In turn, the Optional LUs are extra units, in which the student was free to choose what he wanted to learn and what skills in the course he wanted to develop. Finally, the Recommended PAs are the recommended resources for students to interact with each other in order to develop their skills. In this work, the PA used was *thesis debate* [Santos, Castro, and Menezes 2012]. In this architecture, the intention is stimulate the participants, from their prior knowledge, extend and deepen their knowledge through interactions with peers, following a certain dynamic. In these interactions, which are performed through text production, the participants display their convictions on certain thesis proposed by mediator. Participants can evaluate the work of other peers in different ways. This activity also coordinates the collection and distribution of these assessments. After getting the initial profile of each student from the 3 initial questionnaires, PAs-based adapted activities were recommended according to their profiles. Figure 2 shows an example of a thesis debate recommended for some students, and the assessment of a student by his colleague.

Arquiteturas Pedagógicas Recomendadas

Debate de Teses 4

Determine se cada uma das sentenças seguintes é verdadeira ou falsa. Se a sentença for falsa, explique o porquê.

a) A experiência tem mostrado que a parte mais difícil de resolver um problema de computador é produzir um programa em C.

minha tese 1
por [redacted]

• **TESE de [redacted] 1.pdf**

Sua avaliação
por [redacted]
Nota: 80 de 80

Formulário de avaliação

Você concorda com a resposta do colega? No Yes

Feedback global
Questão respondida perfeitamente!

Figure 2. Example of recommended thesis debate activities for students.

The adaptation of resources on this experiment was made by Resources Adapter Agent. It occurred in two ways: 1) peer selection to correct students' answers and 2) PAs recommendations according to student profile. So, in this test scenario we used Pedagogical Architectures with peer correction. The agents made the choice of students to correct the work of their colleagues, based on their profiles. Students of different profiles were chosen to form their peers of performer-evaluator within the activities. In each record of the student in VLE, the environment continuously perceives, with the aid of agents and ontology, who are the most suitable students to correct the activities of their colleagues. For the choice of peer students, we use the K-Means algorithm in Weka software tool [Weka 2017] regarding skills levels, although the learning styles and student interests may be different between them. We start from the premise that students with different skills, learning styles, and interests could interact in collaborative activities using different points of view. Data mining can be very useful in discovering valuable information which can be used for profiling students based on their academic record. Clustering aims to partition n ($n = 33$ profiles) observations into K clusters in which each observation belongs to the cluster with the nearest mean. The use of K-means clustering algorithm used 2 clusters ($K = 2$) as assumed outcome and the following results were reported: Cluster 0 (lower skills levels): 20 students (60.61%); Cluster 1 (higher skills levels): 13 students (39.39%). Each student of Cluster 0 interacted with at least one student of Cluster 1, and vice versa.

For the recommendation of resources based on PAs, we use the f_{sit} *checks_and_combines(...)*, in which for the recommendation of useful resources (in criterion 3), we used Apriori algorithm with minimum support 10% and minimum confidence 90%. We used as the class attribute the utility level of the recommended resource (ur_r), and the other attributes used are shown in Table 1.

Table 1. Students' Attributes used in Apriori Algorithm.

<i>Attribute</i>	<i>Definition</i>	<i>Possible Values</i>
active/reflective, sensing/intuitive, visual/verbal, and sequential/global	Felder-Silverman Learning Styles	-11 to +11, odd values
Active, reflexive, pragmatist, theorist	Honey-Alonso Learning Styles (Kolb)	0% to 100%
Domain in designing algorithms and programming, abstraction, problem decomposition, simulation, pattern recognition and data analysis, domain in condition and repetition structures	Skills	0.0 to 10.0
Interests	Interests	Nominal Values
Utility_level (ur_r)	Utility Level of a Resource	0.0 to 1.0

The OLM must be easily understood by the student. So, Figure 3(a) shows an example of OLM presented to a student. The presented graph shows the current levels of skills of the student in Computational Thinking course. Figure 3(b) shows the exchange of messages between *Profile Situation Assessment Agent* and *Recommended Resources Agent* (an instance of *Resource Adapter Agent*), by using the *JADEX Communication Analyzer* tool, which allows the visualization of the exchange of messages. There is a third agent, the *Dummy*, responsible for helping dialogs between agents. Comparing the class average in pre- and post-testing, on a scale of 0 to 10, the class pre-test obtained an average of 6.63 with a standard deviation of 3.24, with 48.97% of students approved, whereas in the post-test the class average was 9.85 with a standard deviation of 0.57, an average increase of 32.20%, with 96.55% of students approved. We applied a t-test to verify if there were differences between these averages. We got $t = -3.5983$, $p\text{-value} = 0.001066$, with 95% of confidence interval, confirming that there was a

difference. The students were asked about the activities done. The questions were: Q1) Have you assessed a colleague? Q2) Have you received any recommendation of educational resource? Q3) Do you agree with your OLM shown? Q4) Were the presented resources for you useful? Table 2 shows the percentage of students' answers on the questions. As Table 2 shows, most students in the class assessed at least one colleague (83.87%), received a recommendation of educational resource (87.09%), agreed with the presented OLM (83.87%) and considered the recommended resource useful (96.42%). However, in Q3, 3.58% of the class (1 student) answered “It depends”, as this student did not agree at the beginning of the course with his OLM or disagreed with the recommended resources to him. It can be seen in these tests that successful adaptations are being obtained with the ArCARE approach, since the activities recommendation based on PAs by agents and OLM ontology is a useful technique for improving student learning and his engagement, encouraging the practice of collaborative activities in VLEs.

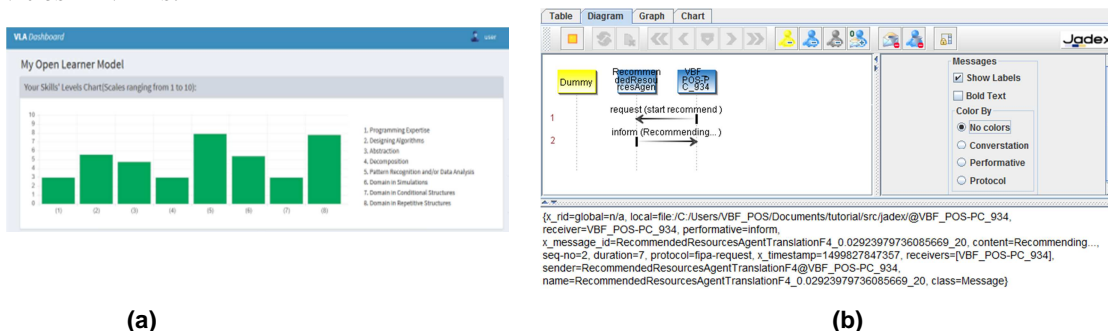


Figure 3. (a) Presenting OLM to students [Bremgartner, Netto, and Menezes 2017]. (b) Information exchanging between agents.

Table 2. Students' Responses in the Questionnaire.

Answers	Questions			
	Q1	Q2	Q3	Q4
Yes	83.87%	87.09%	83.87%	96.42%
No	16.13%	12.91%	12.91%	3.58%
It depends	0%	0%	3.22%	0%

5. Conclusions

This paper has shown the ArCARE framework, which is a model of adaptation of resources in VLEs based on collaborative learning. We developed a multi-agent system and an Open Learner Model Ontology in the IMS LIP standard able to select adapted educational resources in VLEs. This strategy allows greater customization of resources based on the characteristics of the students. The tests showed that adaptations in VLEs through recommendations and adjustments of resources presented to students based on collaborative learning is a solution for the one-size-fits-all problem in learning environments that can increase the student's knowledge and engagement in a useful and effective way. The students' feedbacks, as well as their results in the Computational Thinking course, show that there was a high acceptance of students regarding the use of framework in Moodle. In the case study presented in this paper, the resources presented to students were based on Pedagogical Architectures. The main contribution of ArCARE framework is allowing changes in the pedagogical organization of the learning according to the students' characteristics using multi-agent and OLM-based ontology technologies, allowing adaptations in courses of flexible curricula, besides presenting the OLM to the student, stimulating the self-regulated learning, at the same time that collaborative learning between students occurs. The approach of multi-agent system, plus the OLM, can be applied to other VLEs, since the agents and educational environment can share the same database.

References

- AP CollegeBoard Computer Science Principles – Effective Fall (2016) <https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-computer-science-principles-course-and-exam-description.pdf>
- Bremgartner, V., Netto, J. M., and Menezes, C. S. (2017) “Conceptual Framework of Educational Resources Adaptation for Improve Collaborative Learning in Virtual Learning Environments”. Proceedings of 47th IEEE Frontiers in Education Conference. Indianapolis, USA.
- Bull, S. and Kay, J. (2016) “SMILI[©]: a Framework for Interfaces to Learning Data in Open Learner Models, Learning Analytics and Related Fields”. International Journal of Artificial Intelligence in Education, v26 n1 p293-331.
- Dascalu, M., Bodea, C., Moldoveanu, A., Mohora, A., Lytras, M., & de Pablos, P. O. (2015) “A recommender agent based on learning styles for better virtual collaborative learning experiences”. Computers in Human Behavior, Volume 45, pp. 243-253, ISSN 0747-5632.
- Eryilmaz, E., Chiu, M. M., Thoms, B., Mary, J., & Kim, R. (2013) “Design and evaluation of instructor-based and peer-oriented attention guidance functionalities in an open source anchored discussion system”. Computers & Education 71, Elsevier.
- Georgeff, M., Pell, B., Pollack, M., Tambe, M. e Wooldridge, M. (1999) “The Belief-Desire-Intention Model of Agency”. In: Intelligent Agents V, LNAI 1555, Springer, pp. 1–10.
- Google Computational Thinking Course for Educators (2017) <https://computationalthinkingcourse.withgoogle.com/>
- Honey and Mumford's Learning Styles Questionnaire. (2017) www.nwlink.com/~donclark/hrd/styles/honey_mumford.html
- Hosseini, R., Hsiao, I-H., Guerra, J., Brusilovsky, P. (2015) “What Should I Do Next? Adaptive Sequencing in the Context of Open Social Student Modeling”. Proceedings of 10th European Conference on Technology Enhanced Learning, EC-TEL 2015, Toledo, Spain, , pp 155-168.
- ISTE – International Society for Technology in Education; CSTA Computer Science Teachers Association. (2017) Computational Thinking Leadership Toolkit First Edition, <https://www.iste.org/explore/articleDetail?articleid=152>
- JADE. (2017) Java Agent DEvelopment Framework, www.jade.tilab.com/
- JADEX. (2017) JADEX Active Components, www.activecomponents.org/bin/view/About/Features
- Järvelä, S., and Hadwin, A. F. (2013) “New frontiers: Regulating learning in CSCL”. Educational Psychologist, 48(1), 25-39.
- Latham, A., Crockett, K., and McLean, D. (2013) “An adaptation algorithm for an intelligent natural language tutoring system”. Computers & Education 71, Elsevier, pp. 97–110
- LIP. (2017) Learner Information Package Specification 1.0.1, www.imsglobal.org/profiles/lipinfo01.html#5.1/
- LOM. IEEE, LTSC. (2017). Draft Standard for Learning Technology - Learning Object Metadata. Technical report, IEEE Standards Department, New York.
- Moodle (2017) A Free, Open Source Course Management System for Online Learning, www.moodle.org/
- Oliveira, E. A. e Tedesco, P. (2009) “i-Collaboration in Practice: Results from our Investigation within the Cleverpal Environment”. Proceedings of IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), pp. 347-351.
- Piaget, J. and Inhelder, B. (1969) The Psychology Of The Child. Basic Books.
- Protégé. (2017) The Ontology Editor, www.protege.stanford.edu/
- Santos, L. N., Castro, A. N., and Menezes, C. S. (2012) “Flexible Virtual Environments for Teaching and Learning”. 42nd ASEE/IEEE Frontiers in Education Conference Proceedings (FIE), Seattle, WA, USA.
- Sarmiento, C., Duarte, O., Barrera, M., and Soto, R. (2016) “Semi-Automated Academic Tutor for the Selection of Learning Paths in a Curriculum: An Ontology-Based Approach”. Proceedings of the 8th International Conference on Engineering Education (ICEED), Selangor, Malaysia.
- Soloman, B. A., Felder, R. M. (2017) Index of Learning Styles Questionnaire, www.engr.ncsu.edu/learningstyles/ilsweb.html
- Tavares, L. O., Menezes, C. S., and Nevado, R. A. (2012) “Pedagogical architectures to support the process of teaching and learning of computer programming”. Proceedings of 42th IEEE Frontiers in Education Conference. Seattle, USA.
- Weka (2017) Data Mining Software, <http://www.cs.waikato.ac.nz/ml/weka/>
- Wing, J. (2006) “Computational Thinking”. Communications of the ACM, 3 ed.:33-35.