Conceptual Framework of Educational Resources Adaptation for Improve Collaborative Learning in Virtual Learning Environments

Vitor Bremgartner
CMDI-Federal Institute of Amazonas (IFAM)
Institute of Computing-Federal University of Amazonas (UFAM)
Manaus, Brazil
vitorbref@icomp.ufam.edu.br

J. F. de Magalhães Netto
Institute of Computing-Federal University of Amazonas (UFAM)
Manaus, Brazil
jnetto@icomp.ufam.edu.br

Crediné Silva de Menezes
Faculty of Education-Federal University of Rio Grande do Sul (UFRGS)
Porto Alegre, Brazil
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. This may complicate the effective learning process of each student. In order to solve this problem, one of the original goals of intelligent educational systems is to guide every student to the most appropriate educational contents. So, the approach adopted in this paper 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, allowing the construction of their knowledge, using 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 of particular course content. Results obtained from some tests in a flexible curriculum course of Computational Thinking show the feasibility of the proposal.**

Keywords—Collaborative Learning; Virtual Learning Environments; Adaptation of Resources; Software Agents; Open Learner Model

I. INTRODUCTION

Distance Education is a modality widely used in the teaching-learning processes. To support the distance education or blended courses there are educational environments as Learning Management Systems (LMSs) or Virtual Learning Environments (VLEs). These environments support the process of communication between students, teachers, tutors, 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), known as Intelligent Tutoring Systems (ITSs), 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 [1].

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. Tools in AI field, such as ontologies and software agents, can act integrated into these VLEs, becoming responsible for this intelligence layer and making use of a learner model [1] [2]. This model is a record of the students’ actions as well as useful information about the student in the VLE. As the student profile consists of raw data in the system, the student model consists of the most important information from student profile that will be useful for the processes of adaptation and customization in VLEs [2]. These data may be personal information, preferences, and performance in activities. Institutions and international organizations have established the standardization of student model used in VLEs, and the best known standard currently is the Instructional Management Systems Learning Information Package – IMS LIP [3].

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 (or discipline) in the VLE does not fit according to the several characteristics that each student has, e.g., skills, interests, and learning styles. 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 [4], group support [5], personalization with agents [6] 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 [7]. This framework uses multi-agent system technology that handles a learner model ontology which consists of several students’ characteristics, such as interests, competencies, skills, history of student performance in activities, frequency, and learning styles. We used the IMS LIP standard to integrate all these characteristics. The adaptation provided is the recommendation of adjusted resources based on collaborative learning, for example, Pedagogical Architectures (PAs) [13] containing proposals for collaborative activities in order to student have a more effective learning of in 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 can 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 [8]. In addition, the OLM is dynamically changed during the course, through the students' interactions with the VLE. Regarding visualization of the OLM, this can be seen as a specific type of Learning Analytics (LA), where what is seen is the learner model [8]. LA is the measurement, collection, analysis and reporting of data about students in their contexts, for purposes of understanding and optimizing the learning in the environment where it occurs [9].

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

II. 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 [10], and share responsibility for both the learning process and the common product [11]. The CSCL environments, and more specifically group awareness tools and supporting tools for teacher, have the potential to enhance students' regulation process [12].

This work deals with adaptive and collaborative resources presented to students, especially, Pedagogical Architectures to promote collaborative learning between students, using software agents. Comprehension about the term "Pedagogical Architectures" (PAs) in CSCL has brought multiple interpretations, and these are directly related to an epistemological line that gives basis for its pedagogical proposal. 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 [13]. The construction of the pedagogical strategy involves, however, the formation of an interdisciplinary group with the participation of professionals of education and computing areas [13].

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 [14] 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 [4] 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 [5] 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 [15] is presented Omega Network, a proposal to the social and adaptive e-learning areas. In this work, some identified students are candidates that act as a point of help of other students in an academic course in e-learning mode. This way, it was possible to use all the activities and features offered by VLE with technologies such as: adaptive presentation, collaborative filtering, and peer tutoring. In [16], 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 [17], 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 [18] 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 [28] 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.

III. Defining ArCARE

A. ArCARE Architecture

The ArCARE architecture [34] is shown in Fig. 1. In this model, is shown the framework that constitutes the personalization process of VLEs through adaptation and recommendation of educational resources, depending on the characteristics of the student, making use of software agents and an ontology describing the OLM. This architecture is composed of three fundamental components: *resources adaptation module, VLE database,* and *user action space*.

The resources adaptation module contains the entire layer of intelligence provided by ArCARE. It consists of two groups of agents: those who control and handle the learner model and the agents who are responsible for adapting resources of the educational environment from the student data. The resource adaptation module also has the OLM ontology. The VLE database contains all data regarding the educational environment, as information about the history of users' interactions, OLM information, resources, courses and their Learning Units (LUs). It's on the VLE database that the resources adaptation module operates. In turn, user action area refers to users' interactions with the educational environment. The students can also interact with each other, doing collaborative activities in VLE. Such actions are recorded in the VLE database.

In this framework, we believe that users (students, teachers) are always in interaction with the VLE, showed 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 (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), such as questionnaires to identify learning styles, initial tests for diagnosis of student skills and personal issues according to the course he is attending, in order to initialize his levels of skills. 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 manipulate 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) represented by ellipses, 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.

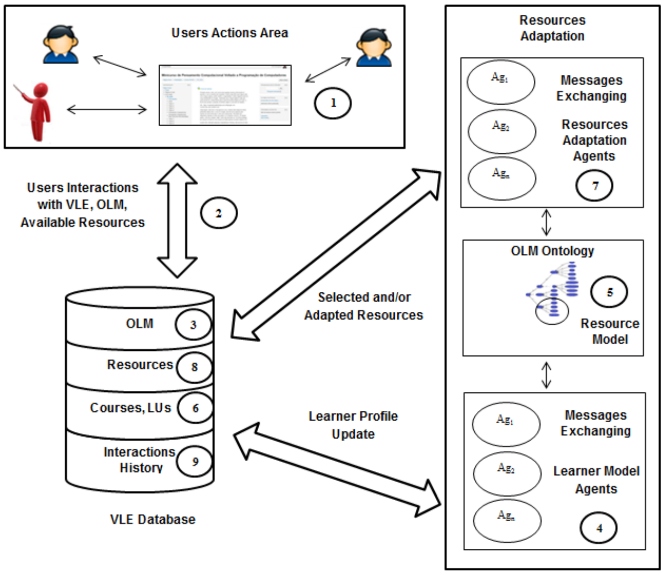

Fig. 1. ArCARE Architecture [34].

B. Agents' Main Roles in the Adaptation Process

As Fig. 1 shows, there are two types of agents: learner model and adaptation resources agents. The first type of agents were developed in the JADE [19] and the latter in JADEX [20] 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 [21], will 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:* sets the initial levels of students' skills and learning styles based on the students' initial manifestations in the environment, from their first access to VLE, such as the students' answers of the questionnaires answered at the beginning of a course. In addition, it also searches the historical data of students' actions in the VLE in order to keep the VLE increasingly updated. This agent sends this information to the Update Profile Agent. These operations are performed by *initial information functions* - f_{in};
- *Learning Assessment Agent*: responsible for evaluating the activities answered by students, besides their frequency in the VLE, and sends the scores and frequencies of the students to the Update Profile Agent. These operations are performed by *student's performance assessment functions* – f_{pa};

- *Update Profile Agent*: responsible for updating 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 (or recommendations) and whether these were really useful for improving students' skill levels. With this result, resource utility levels are updated. These operations are performed by *update student profile functions* – f_{up};

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, for example, search for students' questions and mistakes in answered activities, analyze the students' skills levels, besides to obtaining the history of student actions in the course. Also, this agent maps these students' data into the learning resources needed to be made available for each student in VLE. With this information, this agent sends a message to the Resource Adapter Agent the data of the students with their adapted resources. These operations are performed by a *student's situation verification function* – f_{sit};
- *Resource Adapter Agent*: agent that obtains information from Profile Situation Assessment Agent and executes the process of showing in the VLE resources for each student. These operations are performed by a *resource adaptation function* – f_{adapt}.

C. Agents' Functions

The *initial information functions* (f_{in}) are: *get(<learning_styles>, <interests>)*, which collects data about learning styles and students' interests, *calculates*(h_1a_1, h_1a_2, ..., h_1a_m, h_2a_1, h_2a_2, ..., h_pa_m, k_1, k_2, ..., k_m, d_1, d_2, ..., d_m, t_1,t_2,..t_m, c_1,c_2,..., c_m,f_1, f_2,..., f_m, g_1, g_2,..., 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_pa_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 (1)

$$\sum_{i=1}^{m}(f_i . g_i . c_i . h_i a_i . d_i . t_i . \frac{1}{k_i}) = h_x \qquad (1)$$

The *student's performance assessment functions* (f_{pa}) are defined in the same way as the f_{in} functions, but they consider the student's frequency in the educational environment. In addition, the *Learning Assessment Agent* identifies updates of students' skill levels based on student performances on activities. For this, the function *calculates(...)* is used, so that these skills can be identified by means of an evaluation of the student's interaction history with the VLE. When students' skills are identified, the resources adaptation agents can take appropriate actions to improve student skills, for example, through a recommendation of additional resources.

An important aspect that deserves attention is in an activity there are often several associated skills. In this case, we assume that the student has some of these skills, but others don't. For example, in a Computational Thinking course, students are faced with an activity that involves the skill of domain in repetitive structures. Suppose that a student even may have the skill of domain in conditional structures, but have difficulty dealing with repetitions. Therefore, it is believed that the student will not be able to obtain the maximum grade in an activity. Thus, there is an element in the Learning Assessment Agent, which evaluates the activity from the point of view of each skill associated with an activity *m* separately, in order to know in which skills the student is weak. Figure 2 shows an example where the Learning Assessment Agent obtains the students' activities answers in the VLE database and, at certain moment, finds an activity *m* where, in that activity, its parts are verified (the correct parts indicated by an OK! and where the student missed indicated by an ERROR!). Skills not associated with the activity *m* are not checked, so they will not reflect on upgrading of student skills' levels. In this way, all the skills associated with that activity are covered and it is verified, according to the student's response, whether or not this answer indicates that he has such a skill, and then the student's profile is updated.

The *update student profile functions* (f_{up}) are: *save* (*<frequency>*, *<learning_styles>*, *<interests>*, h_1a_1, h_1a_2, ..., h_1a_m, h_2a_1, h_2a_2, ..., h_pa_m, k_1, k_2, ..., k_m, t_1,t_2,..,t_m), which stores in the VLE database the elements derived from the f_{in} and f_{pa} functions, and the functions *accessed(student, resource, course, time, resource_request)*, *feedback_satisfied (student, resource, course, level_satisfaction)*, *evaluation_pre_pos_resource (student, resource, course, list_activities_pre, list_activities_pos)*, explained below.

After each adaptation done, it is checked whether or not the student has accessed such a resource, by the function *accessed (student, resource, course, time, resource_request)*, where r*esource_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. These successes and failures are in accordance with 2 parameters: 1) feedback from the student, as already shown in the function *feedback_satisfied (...)* and 2) if the student scored well (above average) in activities after recommendation of resources (*list_of_activities_pos*).

For this verification of success or failure of a resource, only activities whose associated skills are the same skills associated with the previously presented resources are verified. All these checks are carried out by the *evaluation_pre_pos_resource (student, resource, course, list_of_activities_pre, list_of_activities_pos)*. 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.

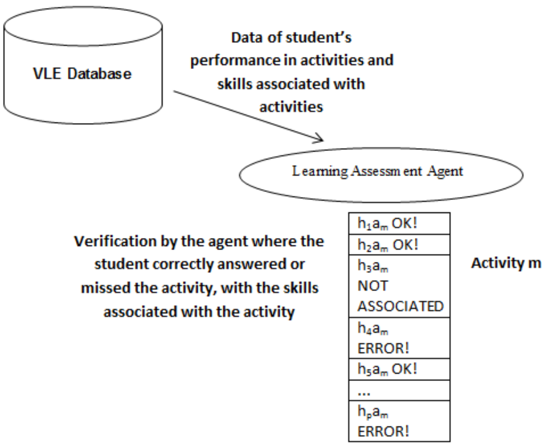

Fig. 2. Verification of students' correctness and mistakes in an activity and its associated skills.

The resource usage history by students is useful to associate several types of student profiles with more useful resources, optimizing the reasoning process in the multi-agent system when resources adaptation agents take action. With these records of past actions, for example, the use of resources by the students, these are also associated with other student profile variables, as already mentioned, such as their learning styles, frequencies in the environment and interests, in order to discover patterns of student profiles that feed the knowledge base of agents, optimizing the process of adapting resources in the VLE in the future.

The *student's situation verification function,* f_{sit}, is *checks_and_combines(c, f, <learning_styles>,* h_1a_1, h_1a_2, ..., h_1a_m, h_2a_1,h_2a_2, ..., h_pa_m, hc_1, hc_2, ..., hc_q, h_1r_1, h_1r_2, ... h_1r_r, h_2r_1, h_2r_2, ..., h_sr_r, ea_1r_1, ea_1r_2, ... ea_1r_r, ea_2r_1, ea_2r_2, ... , ea_nr_r, *<interests>,* i_1r_1, i_1r_2,...,i_yr_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_pa_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_sr_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_nr_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_yr_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 adaptation function*, f_{adapt}, is *executes_strategy(c,e,ta,tr,*r_1,r_2,...,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 will be available to the student. Figure 3 shows a detailed schema, from the ArCARE architecture shown in Fig. 1, of how to update the student profile according to their answers to the activities, as well as an example of a recommendation of resources in a VLE after the skills have been updated. When the student hits or misses an activity, respectively, the levels of *q* students' skills in a given course *(*hc_1, hc_2, ..., hc_q*)* that correspond to the same skills of each of the *m* activities *(*h_1a_1, h_1a_2, ..., h_1a_m, h_2a_1, h_2a_2, ..., h_pa_m*)*, both stored in the VLE database, are incremented or decremented, as shown in Fig. 3, by the learner model agents, especially by the *Learning Assessment Agent*.

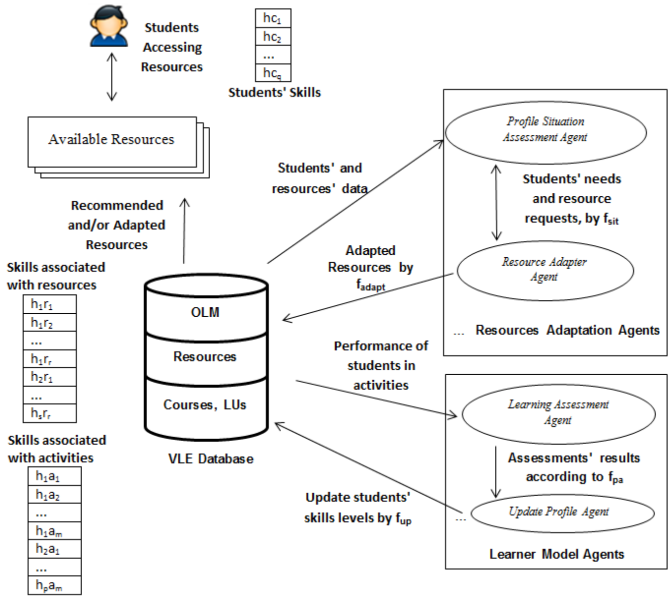

Fig. 3. Update of student's skills based on the results of activities and recommendation of resources according to his skills.

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 $\mathrm{Min}(h_ir_j - h_i)$, $1 \leq i \leq s$, $1 \leq j \leq r$, $h_ir_j - h_i > 0$, that is, the smallest positive difference between the resources *(*h_ir_j*)* and the students *(*h_i*)* skills levels; 2) Resources where occurs $\mathrm{Min}(\mathrm{Abs}(ear_ir_j\text{-}ea_i))$, $1 \leq i \leq n$, $1 \leq j \leq r$, 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 [29] is applied, taking as variables the student's feedback after using the resource, student performance results when using that resource, and frequency of recommendation of that resource. Criteria 1) and 2) are justified by the fact that a resource adapted or recommended for students should have the levels of skills and learning styles as close as possible to the students' levels (but the resources' skills levels should be higher than those of the students'), so that the students, when receiving the resources in the VLE, may learn more and increase their skills' levels.

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. In order to obtain this historical information about past actions in the VLE, the Initial Profile Agent uses f_{in} functions, as shown in Fig. 4. 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. These patterns are then mapped into skills' levels of students and resource utility, whose values are used by the Update Profile Agent's f_{up}, in order to updates the information about students and resources, in which this information will be important for resources adaptation agents use in resource adaptation processes.

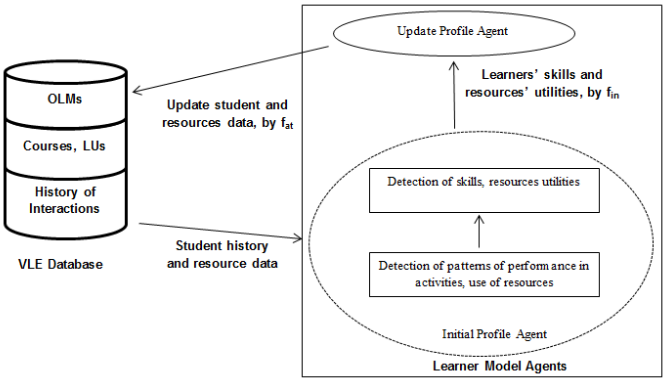

Fig. 4. Obtaining the history of VLE interactions by learner model agents.

D. The Open Learner Model Ontology

Another important step of this work was to build a domain ontology that represents the learner model. For the learner model we used the IMS LIP standard [3], 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 addiction 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 [22]. These metadata are useful for mapping between the fields of the LOM-based Pedagogical Architectures and students' profiles, in order to the PAs recommended 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).

The OLM ontology was built in the Protégé [23] editor and has two main classes: *AID* (describing the agents that constitute the multi-agent system) and *Learner_LIP* (containing the LIP categories used in this work plus the *Learning_Styles* and *Frequency*); both classes are useful allowing the ontology to be integrated into the multi-agent system. Fig. 5 shows a summarized graphical representation of the OLM ontology, only the class *Learner_LIP*, describing the extended LIP standard, presenting the slots (properties) of the classes. For example, in the *Learning_Styles* class, a property is *ls_fs_active_reflective*, indicating whether the learner tends to be more active or reflective in the Felder-Silverman learning styles [24].

The 2 AID presented in Fig. 5 are the *Profile Situation Assessment Agent and Resource Adapter Agent.* The relationships between classes describe the facets of ontology, represented by blue and black arrows (the latter indicating only the relationship "*isa*", subclass to superclass). As some examples of facets (in this case, rules) in blue arrows, the student *doesActivity* and *hasDoubts_Errors* (connecting *Identification* and *Activity* classes), *hasSkillsCompetencies* (connecting *Identification* and *Competency* classes). Furthermore, the *Resource Adapter Agent recommends_resources*, e.g., PAs according to the student profile and activities that will match a given student or groups of students profiles (*gets_best_fit_resources*, which comes from the *Activity* class).

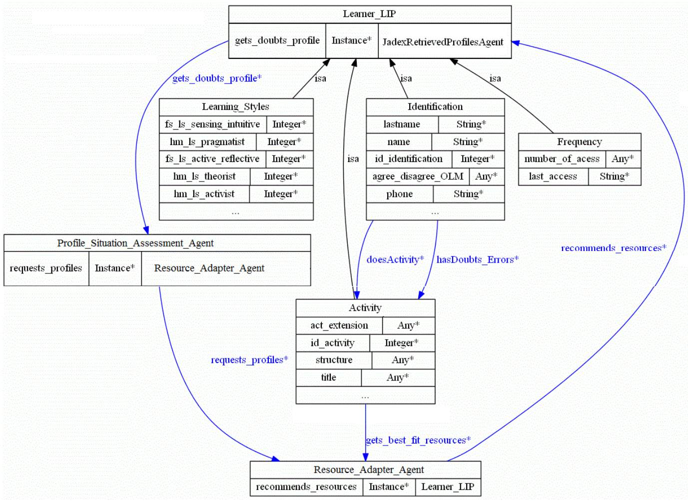

Fig. 5. Graphical representation of the Open Learner Model Ontology summarized.

IV. 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 [26]. Curricular flexibility means that every student has the option of choosing a set of Learning Units to develop a particular professional profile [28]. According to [25], 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). The educational objectives of the course, aligned with the difficulties commonly encountered by beginners in programming, are: (i) assess their own knowledge; (ii) understand and interpret problems; (iii) solve problems by applying Computational Thinking; and (iv) debugging errors and correct them. 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 [30]. The skills involved were those classified by Google Computational Thinking Course [31] in partnership with International Society for Technology in Education (ISTE) and Computer Science Teachers Association (CSTA) [32], 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 [24] of students; 2) Honey-Alonso Learning Styles Questionnaire, adapted from Honey-Mumford [27]. It was decided to choose these two questionnaires in order to have more accurate information about the students and their learning styles, obtained by the combination of two questionnaires mentioned above; 3) The pre-test itself, as mentioned previously. The links to access these questionnaires and OLM were available in VLE. Figure 6 shows an example of a pre-test question in the VLE.

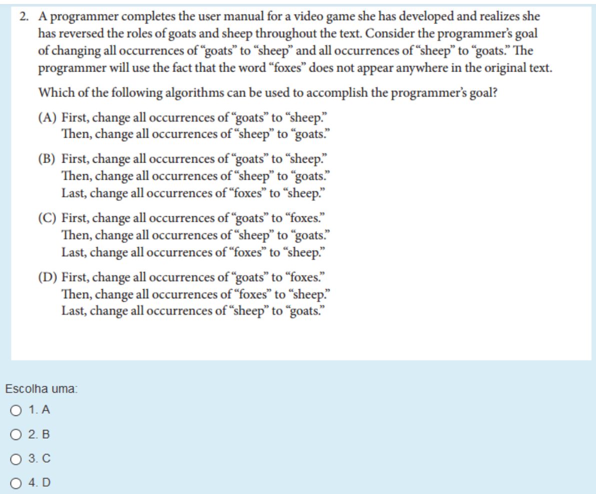

Fig. 6. A pre-test question.

In this course the LUs are 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 can choose the order in which he will study them. In turn, the Optional LUs are extra units, in which the student is free to choose what he wants to learn and what skills in the course he wants 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* [14]. 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 7 shows an example of a thesis debate recommended for some students, and the assessment of a student by his colleague.

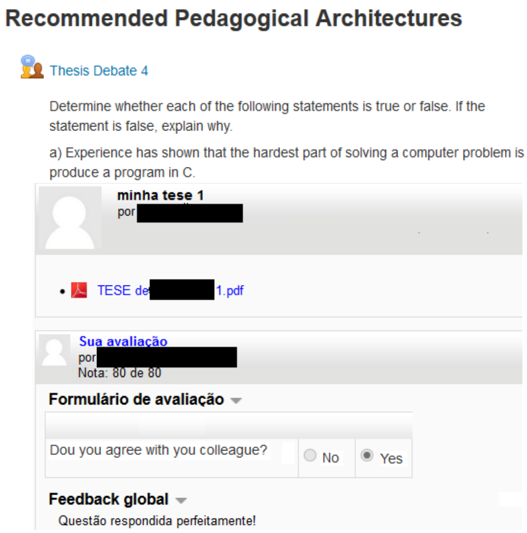

Fig. 7. Example of recommended thesis debate activities for students

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 [33] regarding skills levels, although the learning styles and student interests may be different. We start from the premise that students with different 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 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 I.

TABLE I. STUDENTS' ATTRIBUTES USED IN APRIORI ALGORITHM.

Attribute	*Definition*	*Possible Values*
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, Fig. 8 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 9 shows the exchange of messages between *Learning Assessment Agent* (d) and *Update Profile Agent* (u), by using the JADE *Sniffer* tool, which allows the visualization of the exchange of messages. There is a third agent, the *Directory Facilitator* (df), responsible for controlling requests, provision of services and dialogs between agents.

Fig. 8. Presenting OLM to students [34].

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. 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 II shows the percentage of students' answers on the questions. As Table II 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 OLMs 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.

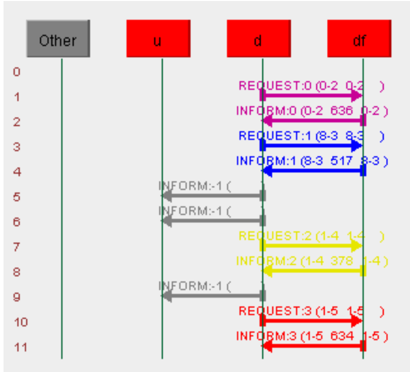

Fig. 9. Information exchanging between agents.

TABLE II. 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%

V. CONCLUSIONS AND FUTURE WORK

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 CT 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. As future work, the idea is to test the approach in mobile devices and adjust the interface of thesis debates activities, making them easier to use.

REFERENCES

[1] Bremgartner, V., Netto, J. M., & Menezes, C. S. Adaptation Resources in Virtual Learning Environments under Constructivist Approach: A Systematic Review. Proceedings of 45th IEEE Frontiers in Education Conference. El Paso, USA (2015).

[2] Fröschl, C. User Modeling and User Profiling in Adaptive E-learning Systems. Master's Thesis. Institute for Information Systems and Computer Media, Graz University of Technology, Graz, Austria (2005)

[3] LIP. Learner Information Package Specification 1.0.1, www.imsglobal.org/profiles/lipinfo01.html#5.1/

[4] Latham, A., Crockett, K., McLean, D.: An adaptation algorithm for an intelligent natural language tutoring system. Computers & Education 71, p. 97–110 (2013)

[5] Oliveira, E. A., Tedesco, P.: 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 (2009)

[6] de la Iglesia, D. G., Calderón, J. F., Weyns, D., Milrad, M., & Nussbaum. M. A Self-Adaptive Multi-Agent System Approach for Collaborative Mobile Learning. In *IEEE Transactions on Learning Technologies*, vol. 8, no. 2, pp. 158-172 (2015)

[7] Piaget, J. & Inhelder, B. The Psychology Of The Child. Basic Books (1969)

[8] Bull, S. & Kay, J. 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 Mar (2016)

[9] Siemens, G. Learning Analytics: Envisioning a Research Discipline and a Domain of Practice. ACM 2nd International Conference on Learning Analytics and Knowledge Proceedings, pp. 4-8 (2012).

[10] Järvelä, S., & Hadwin, A. F. New frontiers: Regulating learning in CSCL. Educational Psychologist, 48(1), 25-39. doi:10.1080/00461520.2012.74800 (2013)

[11] Fransen, J., Weinberger, A., & Kirschner, P. Team effectiveness and team development in CSCL. Educational Psychologist, 48(1), 9–24. (2013)

[12] Janssen, J., Erkens, G., & Kirschner, P. A. (2011). Group awareness tools: It's what you do with it that matters. Computers in Human Behavior, 27(3), 1046-1058. (2011)

[13] Tavares, L. O., Menezes, C. S., & Nevado, R. A. Pedagogical architectures to support the process of teaching and learning of computer programming. Proceedings of 42th IEEE Frontiers in Education Conference. Seattle, USA. (2012)

[14] Santos, L. N., Castro, A. N., Menezes, C. S.: Flexible Virtual Environments for Teaching and Learning. 42nd ASEE/IEEE Frontiers in Education Conference Proceedings (FIE), Seattle, WA, USA, pp. 1388-1393 (2012)

[15] Dominoni M., Riva, G., & Pinardi, S. Omega Network: an Adaptive Approach to Social Learning. Proceedings of Intelligent Systems Design and Applications (ISDA), 10th International Conference on, (pp. 953-958) (2010)

[16] Eryilmaz, E., Chiu, M. M., Thoms, B., Mary, J., & Kim, R. Design and evaluation of instructor-based and peer-oriented attention guidance functionalities in an open source anchored discussion system. Computers & Education. (2013)

[17] Dascalu, M., Bodea, C., Moldoveanu, A., Mohora, A., Lytras, M., & de Pablos, P. O. 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 (2015)

[18] Hosseini, R., Hsiao, I-H., Guerra, J., Brusilovsky, P. 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. (2015)

[19] JADE: Java Agent DEvelopment Framework, www.jade.tilab.com/

[20] JADEX Active Components, www.activecomponents.org/bin/view/About/Features

[21] Georgeff, M., Pell, B., Pollack, M., Tambe, M., Wooldridge, M. The Belief-Desire-Intention Model of Agency. In: Intelligent Agents V, LNAI 1555, Springer, pp. 1–10 (1999)

[22] LOM. IEEE, LTSC. Draft Standard for Learning Technology - Learning Object Metadata. Technical report, IEEE Standards Department, New York (2017)

[23] Protégé. The Ontology Editor, www.protege.stanford.edu/

[24] Soloman, B. A., Felder, R. M.: Index of Learning Styles Questionnaire, www.engr.ncsu.edu/learningstyles/ilsweb.html

[25] Wing, J. Computational Thinking. Communications of the ACM, 3 ed.:33-35 (2006)

[26] Moodle: A Free, Open Source Course Management System for Online Learning, www.moodle.org/

[27] Honey and Mumford's Learning Styles Questionnaire, www.nwlink.com/~donclark/hrd/styles/honey_mumford.html

[28] Sarmiento, C., Duarte, O., Barrera, M., Soto, R. Semi-Automated Academic Tutor for the Selection of Learning Paths in a Curriculum: An Ontology-Based Approach. Proceeding of the 8th International Conference on Engineering Education (ICEED), Selangor, Malaysia. (2016).

[29] Agrawal, R., Srikant, R. Fast algorithms for mining association rules in large databases. Proceedings of the International Conference on VeryLarge Databases, Santiago, Chile. (1994)

[30] 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

[31] Google Computational Thinking Course for Educators, https://computationalthinkingcourse.withgoogle.com/

[32] ISTE – International Society for Technology in Education; CSTA Computer Science Teachers Association. Computational Thinking Leadership Toolkit First Edition, https://www.iste.org/explore/articleDetail?articleid=152

[33] Weka: Data Mining Software, http://www.cs.waikato.ac.nz/ml/weka/

[34] Bremgartner, V., Netto, J. M., & Menezes, C. S. Conceptual Framework for Collaborative Educational Resources Adaptation in Virtual Learning Environments. Proceedings of the 18th International Conference on Artificial Intelligence in Education (AIED 2017), Wuhan, China. (2017)