IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 13, 2019, accepted January 3, 2020, date of publication January 14, 2020, date of current version January 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2966595

CASA: An Architecture to Support Complex Assessment Scenarios

MIKEL VILLAMAÑE[1], AINHOA ÁLVAREZ[2], AND MIKEL LARRAÑAGA[3], (Member, IEEE)
[1]Faculty of Engineering in Bilbao, University of the Basque Country (UPV/EHU), 48013 Bilbao, Spain
[2]Faculty of Informatics, University of the Basque Country (UPV/EHU), 20018 Donostia-San Sebastián, Spain
[3]Faculty of Engineering in Vitoria-Gasteiz, University of the Basque Country (UPV/EHU), 01006 Vitoria-Gasteiz, Spain

Corresponding author: Mikel Villamañe (mikel.v@ehu.eus)

This work was supported in part by the Basque Government under Grant IT980-16, and in part by the Office of the Vice-Chancellor for Innovation, Social Engagement, and Cultural action of the UPV/EHU through the SAE-HELAZ under Grant HBT-Adituak 2018-19/6.

ABSTRACT The evaluation or assessment of student performance and knowledge is a central element of most instructional design models, as it provides the information required to take remediation actions and improve the learning process. However, the assessment contexts present such a diverse range of cases due to course, teacher and student idiosyncrasies that it is difficult to support all the possibilities via software. To deal with this problem, this paper presents an architecture that satisfies the requirements to support complex assessment scenarios, those that potentially involve different evaluators using diverse assessment methods, tools and techniques at the same time. An analysis of the literature and the assessment scenarios of courses from Computer Engineering grades was carried out to identify the requirements for systems that support such complex assessment scenarios and to infer the information needed to support them. This information was modeled and formalized in the Complex Assessment Scenarios Architecture (CASA) that comprises functional modules, knowledge bases and the relationships among them. Following this architecture, an assessment system (AdESMuS) was developed using agile methodologies. AdESMuS has been satisfactorily tested under different complex assessment scenarios proving to be able to support those scenarios' requirements. This paper presents CASA, the architecture for developing a system to support complex assessment scenarios along with the satisfactory case study of AdESMuS.

INDEX TERMS Software architecture, data models, knowledge management, learning management systems, requirements management, software tools.

I. INTRODUCTION

Instructional design refers to the practice of creating "instructional experiences which make the acquisition of knowledge and skills more efficient, effective and appealing" [1]. Many instructional design models such as ADDIE [2] have been defined, and a key aspect of all of them is assessment. In these models, assessment allows instructors and students to identify how successful the experience has been for students, to measure the performance of the students and to take remediation actions when needed. Therefore, planning the assessment is very important in any instructional design process [3].

Any instructional design can establish many different evaluation or assessment approaches [4] by presenting miscellaneous characteristics such as different types of evaluation committees, various assessment instruments, or several combinations of individual assessable elements to obtain a student's grade for a course. For example, lecturers often consider final or continuous assessment approaches in their courses.

However, any assessment approach should consider both summative and formative assessment [5]. Summative assessment refers to the final evaluation of a unit with the aim of giving students a grade, whereas formative assessment focuses on monitoring students with the aim of providing feedback and improving the learning process. Taras [5] points out that every formative assessment begins with a summative assessment, i.e. a judgment of the work, which is complemented with feedback that might help students improve their learning. Moreover, the information elicited from the assessment can also be used by teachers to improve their teaching.

The associate editor coordinating the review of this manuscript and approving it for publication was Ghassem Mokhtari.

Assessment systems must be flexible and allow the idiosyncrasies of each assessment approach to be incorporated and the definition and use of both summative and formative assessment tools to be supported.

Summative tools are easy to work with, as they only need to manage the different numerical grades and their combination to get the final course grade.

For formative assessment, although multiple tools are available, rubrics are extensively used [6]. Rubrics are scoring tools that lay out the specific expectations for an assignment. Their definition entails determining performance levels and their corresponding descriptions for each dimension or aspect of the assignment being evaluated. Different kinds of rubrics have been identified with regard to the level of detail they have [7]. Although studies have compared different kinds of rubrics, there is not sound evidence for one kind outperforming the others. Therefore, users usually select the type of rubric to use according to their preferences.

Rubrics are helpful tools for feedback processes because students receive all the necessary details about the goals not achieved in the assignment, and they are even given suggestions for attaining better performance (in the form of descriptions of the higher performance levels) [8].

All these aspects are even more important in complex assessment scenarios, i.e. those that involve several evaluators potentially using a combination of assessment methods, tools and techniques. With these concerns in mind, this paper presents CASA (Complex Assessment Scenarios Architecture), a flexible software architecture that supports complex continuous assessment processes and can compile information from different sources using different assessment tools. This architecture has been defined taking into account the needs of blended learning environments which are one of the main current trends in higher education [9].

A system developed complying with the CASA architecture could either work autonomously or be integrated with other systems as a provider of all the services required to carry out and monitor an assessment process.

This paper is structured as follows. First, a set of requirements for an assessment system that supports complex assessment processes is presented. Then, CASA, the architecture to support those requirements and its knowledge models' design, is depicted. Next, a case study carried out to evaluate CASA is detailed. In the case study, a system was developed according to CASA and it was tested in different complex assessment scenarios analyzing the fulfillment of those scenarios' requirements. Finally, some conclusions and future work lines are outlined.

II. REQUIREMENTS TO SUPPORT COMPLEX CONTINUOUS ASSESSMENT PROCESSES

From the analysis of the literature and from an in-depth analysis of the evaluation or assessment approaches in undergraduate Computer Engineering courses at the University of the Basque Country, a set of requirements for a system that supports complex assessment processes has been derived and is summarized as follows:

1) *SelectTools:* Allow instructors to select the assessment tool for each assessable item. Instructors should be able to determine which assessment tools will be used for each assessable item, as not all assessment tools are appropriate for every situation. Moreover, diverse assessment tools can represent the same information in different formats (for example, rubrics with different detail levels can be used to assess the same element). When many instructors are involved in assessing a piece of work, it can sometimes be difficult to agree on which tool should be used [10]. Therefore, allowing each instructor to select the assessment tool to be used should be guaranteed 11].
2) *MultipleSources:* Aggregate and integrate multiple sources of information to obtain the final grade [12]. Sometimes the assessment of an item can come from combining different sources. For example, the assessment of a software program can be partly carried out by the lecturer, who takes into consideration the correctness of the design, and partly by an external system like Web-CAT,[1] which uses unit-testing techniques to automatically grade programming assignments. Similarly, in blended learning scenarios, several sources of information are usually combined to obtain the final grade [13].
3) *Rubrics:* Management and use of rubrics. Currently, there is a growing interest in formative assessment. Rubrics are one of the main tools used for this type of assessment [6], [14]–[16]. Therefore, any software aimed at supporting complex assessment processes should include the possibility of managing and using rubrics.
4) *CondAssessment:* Conditional assessment. The way the final grade of a course is calculated can depend on a condition related to the performance of students on certain assessable items [17], [18]. For example, in the University of the Basque Country many courses obtain the final grade considering an exam and a final project. In these cases, the lecturer may require the students to achieve a grade above a certain threshold on the exam in order to consider the final project.
5) *DiffApproaches:* Different assessment approaches for the same course. Providing different approaches can promote students' engagement [19] and can help to deal with the diversity of students enrolled in courses 20]. Moreover, providing different assessment approaches for the same course is also encouraged in some universities. [2,3] For example, our university recommends continuous assessment approaches to be used. However, according to the assessment regulations of our

[1] http://web-cat.org

[2] https://www.cmu.edu/teaching/designteach/design/syllabus/samples-gradingpolicies/index.html

[3] https://www.tcd.ie/CAPSL/TIC/guidelines/assessments/

university, students can apply for a final assessment approach if they work or cannot attend class at the scheduled times due to illnesses or to some other accepted reasons. Each of these approaches can be composed of different assessable items, each having a different weight in the final grade.

6) *Change:* Allow the assessment approach for a student to be changed during the course. Sometimes students cannot continue with the assessment approach they began the course with due to external factors such as starting a job or becoming ill. When this happens, they should have the option of changing the assessment approach. For example, at our university [4] students can change the assessment approach from the continuous to the final one by asking for the change in a specific time period.
7) *Committee:* Variable or customized evaluation committees of different sizes or compositions should be allowed for each assessable item. Depending on the context of the assessment, sometimes different evaluation committees assess the assessable items from each course [21]–[23]. For example, the course's assessable items might be distributed among different lecturers. Moreover, sometimes the same item must be assessed by more than one evaluator and then the results must be combined. This also occurs in peer assessment, where the evaluators are the students 24]. In those cases, the number of students that assess each item can vary. In some cases an item can be evaluated by the whole class, whereas in other cases the evaluators can be a subset of students.
8) *Privacy:* The privacy and confidentiality of students' grades should be assured following the principles of data protection regulation in the European Union [25], i.e. evaluators should not be able to see or modify the assessments carried out by the remaining course evaluators and the students should not be able to consult other students' grades.

III. ANALYSIS OF CURRENT SYSTEMS' SUPPORT FOR ASSESSMENT PROCESSES

Taking into account the previously stated requirements for an assessment system, a study of existing systems that include complex grading options was conducted in order to see whether any of them covered all the requisites. This analysis included the main Learning Management Systems (LMSs) currently used, such as Moodle,[5] Blackboard,[6] Sakai,[7] Canvas[8] or Edmodo.[9]

[4]https://www.ehu.eus/es/web/estudiosdegrado-gradukoikasketak/ebaluaziorako-arautegia

[5] https://moodle.com/

[6] https://www.blackboard.com/

[7] https://www.sakailms.org/

[8] https://www.canvaslms.com/

[9] https://www.edmodo.com

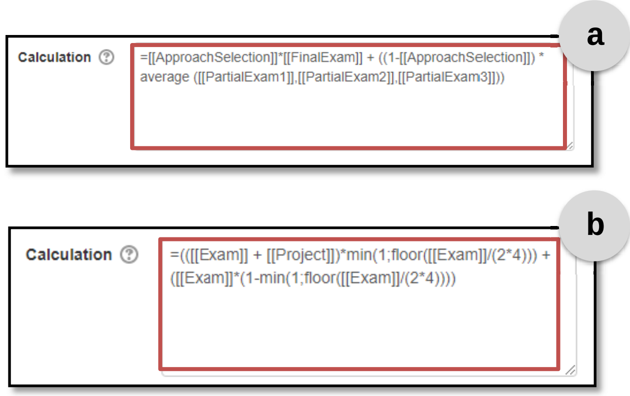

FIGURE 1. Simulating different assessment approaches (a) and a conditional assessment (b) in moodle.

All the LMS analyzed include the possibility of using different tools for grading each assessable item. Many systems allow the use of Rubrics (*Rubrics* requirement). However, they only allow the evaluator to select the tool to be used for each assessable item at definition time (*SelectTools* requirement). With regard to using multiple sources to obtain a grade for a student on a specific assessment (*MultipleSources* requirement), this could be simulated in most systems by dividing the assessment into several assessable items.

In terms of being able to include different assessment approaches in a course or change students from one assessment approach to another during the course (*DiffApproaches* and *Change* requirements), of the analyzed systems only Moodle and Edmodo meet those requirements in one way or another. Edmodo allows different grading periods to be defined, each one with a different approach, or the assessment approach to be changed for the whole class. However, it does not enable changing the approach for a particular student. Therefore, it does not completely fulfill the *Change* requirement.

Moodle fulfills both requirements completely. To enable switching from one assessment approach to another, an extra assessable item has to be defined in order to be used as the assessment approach selector. For example, the value of the *ApproachSelection* assessable item in the formula in Fig. 1-a determines how the student's grade is obtained. In this case, when *ApproachSelection* is 1, the grade is given by the assessable item *FinalExam*, whereas when *ApproachSelection* is 0, the final grade is determined by calculating the average of three exams (the assessable items *PartialExam1*, *PartialExam2* and *PartialExam3*).

Among the tools analyzed, the only one that supports conditional assessment (*CondAssessment* requirement) is also Moodle. Fig. 1-b shows the formula that allows an approach to be defined in which the student has to get at least 4 points out of 10 on two exams in order for the mean of both exams to be computed. Otherwise, the final grade will be the grade obtained on the *Exam* item.

Writing this kind of formula in order to simulate different assessment approaches or establish conditions to compute the final grade is challenging even for expert users. When the number of assessable items grows, as is usual in continuous assessment approaches, it is very difficult to define the different conditions and calculations due to the lack of an editor or assistant.

Finally, regarding the definition of different evaluation committees (*Committee* requirement), the systems analyzed allow several users to assess the course items. However, they do not allow the person in charge of each item to be indicated, and there is no privacy in the assessment.

The analysis also considered pure evaluation systems such as Surpass[10] or QuestionMark.[11] Those systems center on creating very rich assessment tasks but do not address higher level functionalities such as the management of Committees (*Committee* requirement), or changing the assessment approach during the course (*Change* requirement). They are therefore not powerful enough to fulfill all the mentioned requirements, but very interesting to integrate them in other systems, such as LMSs, to provide rich evaluation tasks.

In summary, none of the tools analyzed covered all the defined requirements, which led us to propose a system architecture, CASA, that would fulfill this gap.

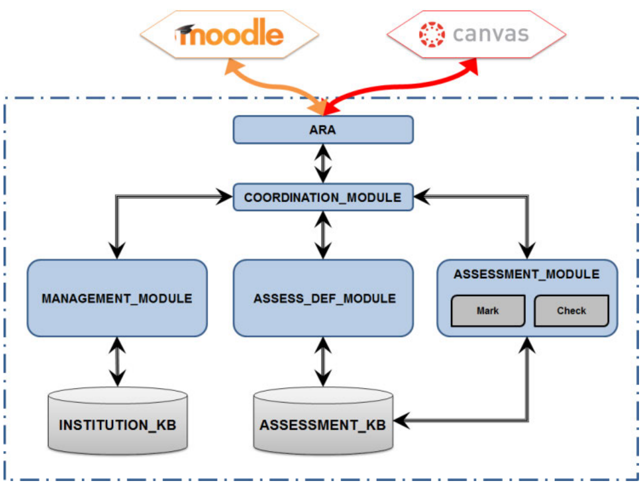

FIGURE 2. CASA architecture.

IV. THE CASA ARCHITECTURE

The systems developed following CASA should have the capacity for being integrated in blended learning environments. Therefore, the architecture should be scalable and interoperable with other systems. To this end, as shown in Fig. 2, the proposed architecture is structured in five modules (*ARA*, *Coordination_Module*, *Management_Module*, *Assess_Def_Module* and *Assessment_Module*) and two knowledge bases (*Institution_KB* and *Assessment_KB*) that store all the information. The conceptual model of these knowledge bases is defined by the three knowledge models explained in the next section.

The *ARA* (Assessment REST API) module is conceived as a REST (Representational State Transfer) Web service [26] that interacts with the *Coordination_Module* to provide other systems, such as Moodle or similar, with the functionality that is needed for complex continuous assessment (course definition, assessment definition and performing and consulting assessments). *ARA* is the only module in the architecture that is accessible from outside the system, which will make the system more secure.

The coordination module (*Coordination_Module*) processes each request from *ARA* by forwarding it to the module that is capable of dealing with it. In addition, *Coordination_Module* manages the communication between the internal modules of the architecture, minimizing their interdependencies and providing the data and functionalities that will be offered by *ARA*. Furthermore, the inclusion of this mediating module fosters the scalability and facilitates the integration of new modules with additional functionality, as it avoids dependencies between the remaining modules.

Management_Module provides all the functionality required to define, consult and manage the information related to the academic administration of an educational institution including degrees, courses, lecturers and enrolment. All this administrative information is stored in the *Institution_KB* knowledge base.

The module that defines the assessment approaches (*Assess_Def_Module*) allows lecturers to manage the assessment definition for a specific course. Through this module, the lecturers in charge of the groups can define the assessment approaches for the courses as well as the tools that will be used for this purpose. All the information about the assessment procedures is stored into the *Assessment_KB* knowledge base.

Finally, the assessment module (*Assessment_Module*) provides the means to carry out and consult the assessments through two submodules. The *Mark* submodule allows the assessment to be carried out and updates the information in the *Assessment_KB* using the assessment results. The *Check* submodule deals with the queries about assessments and provides information as a function of the user role, and it ensures security and privacy. For example, management staff may only be provided with aggregated data about the courses, whereas students can consult their performance and compare it to the performance of the group they belong to, but they cannot access the grades of other students.

V. KNOWLEDGE MODELS

The knowledge is represented in CASA using three models: *Institution_Mod*, *Definition_Mod* and *Assessment_Mod*.

Institution_mod represents all the academic information needed for assessment purposes including degrees, university colleges, enrolled students, lecturers in charge of the assessment courses and student groups in the course. This model establishes the structure of the *Institution_KB*.

[10] https://www.btl.com/surpass/
[11] https://www.questionmark.com

Definition_Mod and *Assessment_Mod* are used to represent the required aspects of assessment. The former is used to define the assessment process for each course, while the latter allows the definition of the assessment approach to be linked with the people involved in the assessment and its results. Those models define the structure of the *Assessment_KB*.

As the architecture must be able to incorporate information coming from different sources (from online platforms, from face to face interactions and so on), those models re-use existing ontologies with similar purposes and adapts them to integrate information from different sources.

In what follows, we discuss how these models can be used to describe the assessment of different courses. For the sake of making easier to understand these models, a course in which two assessment approaches coexist (see Fig. 3) will be used as example.

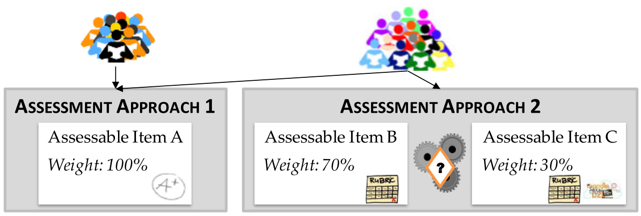

FIGURE 3. **Assessment approaches.**

The first assessment approach relies on a final exam that the lecturer evaluates by hand. The second approach is composed of two assessable items, each with a different weight in the final grade and a condition that determines the way the final grade is computed. The first assessable item is assessed by two evaluators following a rubric. For the second item, two sources are used to compute the final grade; the first source is the assessment carried out by the lecturers following a rubric, whereas the second source is the grade given by an external system. Both grades must be combined to obtain the student's final grade. However, in this approach, the grades from both items are only combined when the grade for the first item is higher than 3.5 on a 10-point scale.

The students in the example course are divided in two groups. The first group follows Assessment Approach 1, which entails a unique assessable element, whereas the students in group two can follow that approach or Assessment Approach 2, which consists of two assessable elements.

For reasons of clarity, some explanations of the models are accompanied with mockups of feasible graphical interfaces and schemas to show the information a system based in the CASA architecture should work with.

A. DEFINING THE ASSESSMENT APPROACHES

Defining the assessment approaches involves using elements defined in *Definition_Mod* and *Institution_Mod* (see Fig. 4). The general definition of assessment includes defining different assessment approaches (*AssessmentApproach*), which entails diverse assessable items (*AssessableItem*) along with their weight (*ItemWeight*) and the competences (*Competence*) considered in the assessment. In addition, information about where the assessment of the assessable item comes from can be described (*Source*). *Definition_Mod* also offers the possibility of describing the different conditions (*Condition*) to be considered for grade calculation through the definition of expressions (*BooleanExpression* and *ArithmeticExpression*, among others).

1) GENERAL DEFINITION OF THE ASSESSMENT

Fig. 5 shows a feasible graphical interface for the definition of the first assessment approach (up) which is represented in the model (down) by an *AssessmentApproach* (Fig. 5-a) in which a unique grade will be given to *AssessableItem* A (Fig. 5-c). This means that the percentage of influence of this *AssessableItem* will be 100% (*ItemWeight*, Fig. 5-b).

In this *AssessmentApproach*, the assessment is provided by a unique source (Fig. 5-d) without storing any evidence about it. The *SourceType* used is *MANUAL* (Fig. 5-e) and the evaluators of that *AssessableItem* directly input the numeric grade they believe the student has earned.

The definition of the second *AssessmentApproach* is more complex, as it involves different assessable items, each of which is assessed using different sources (see Fig. 6 up). In order to completely represent the second *AssessmentApproach* (Fig. 6 down), an *AssessableItem* instance must be defined for each of the assessable elements and related to the *AssessmentApproach* through *ItemWeight* instances. In this approach, the first *ItemWeight* object has a weight of 70% in the final grade and the second has a weight of 30%. Fig. 6-a shows the two assessable items involved in this approach.

In addition, the first assessable element has a rubric as the unique source for the assessment. For the second assessable item, the information from two sources must be combined (Fig. 6-b): a rubric filled in by the evaluator (with a weight of 40%) and the grade from the external system (whose weight is 60%).

In general, the grade sources can be rubrics, manual evaluation by a lecturer or an external LMS or a similar system. When combining the grades from different sources, they can be combined in a percentage-based way, as in this example, or on an average, maximum or minimum basis.

The second assessment approach requires a student to get at least 3.5 out of 10 on Assessable Item B in order to allow the grades on all the assessable elements to be combined in the final grade. Otherwise, the final score will be the grade on Assessable Item B (see Fig. 7).

2) ESTABLISHING THE ASSESSMENT APPROACHES FOR STUDENTS

In the example used throughout this paper, students are divided into two groups. The first group has only one assessment approach available (see Fig. 8-a), whereas the second group has both assessment approaches available (see Fig. 8-b). This assessment approach can be assigned individually for each student (see Fig. 8-c) and changed during the course.

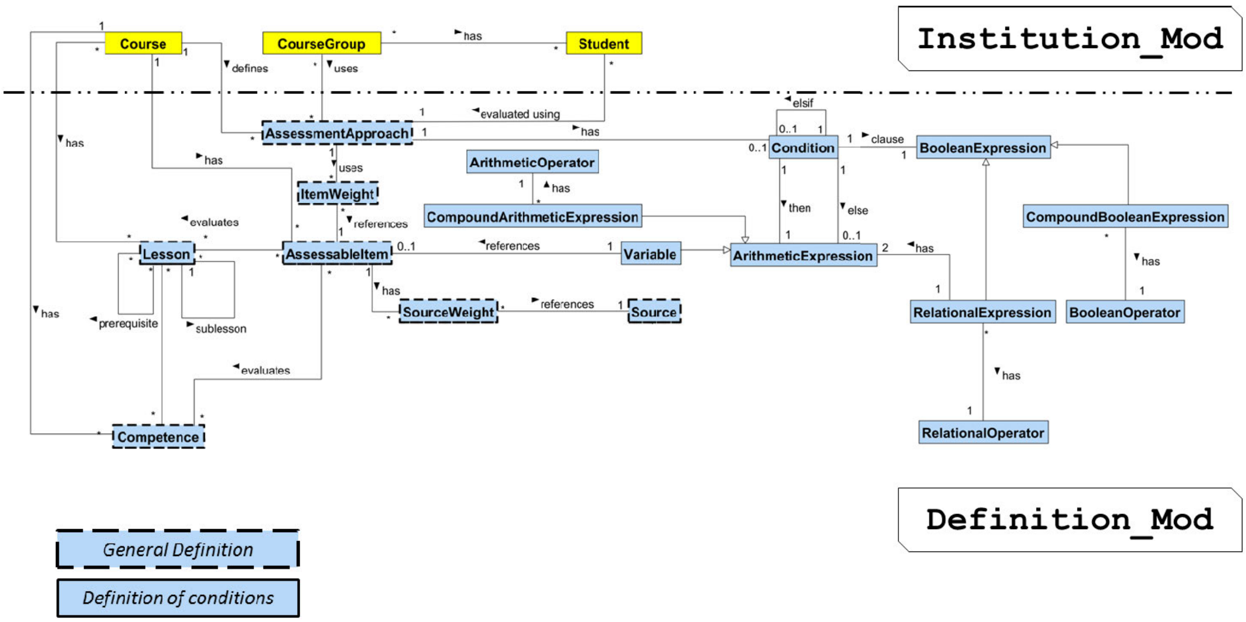

FIGURE 4. Definition_Mod represented in UML.

B. CARRYING OUT THE ASSESSMENT PROCESS

In what follows, the assessment process is described, along with how it is stored (see Fig. 9). For any *AssessmentApproach*, *Assessment_Mod* allows the person (lecturer or student) in charge of assessing each *AssessableItem* to be defined for each student (*AssessmentTask* class).

For instance, the second assessment approach of the example entails two assessable items, where each one is evaluated by different lecturers using their corresponding assessment tools (see Fig. 10).

The result of the assessment for each *AssessableItem*, carried out by each evaluator using a specific *Source* in an *AssessmentApproach* is stored in an *AssessmentTask* instance. Fig. 11 shows a feasible interface to be used by an evaluator to evaluate Assessable Item B in the second assessment approach using a rubric.

Fig. 12 shows an example of two Assessment Tasks (one has a computed grade of 8 points, and the other one has a computed grade of 7 points) for the Assessable Item B of Assessment Approach 2.

The grade for an assessable item is obtained by compiling the grades of each related assessment, and the grade for the assessable item is stored in an *AssessableItemGrade* instance.

Fig. 12 (down) shows the instantiation of the classes required to represent the example from Fig. 12 (up). In this case, the grade for Assessable Item B (*AssessableItemGrade*) is obtained by averaging the grades provided by two lecturers to that item, so two *AssessmentTask* elements are needed. Lecturers can include personalized feedback not only at the rubric or dimension levels, but also in the assessment task level regardless of the way the task has been assessed.

The student's final grade is calculated by taking into account the grades obtained for each assessable item (*AssessableItemGrade* in Fig. 12 down) and their weights in the final grade according to the conditions established for the assessment approach (see Fig. 13).

Fig. 13 (down) shows the instantiation of the model in order to calculate the final grade of the student in the task (*AssessmentResult*) from the *AssessableItemGrade* objects corresponding to Assessable Item B and Assessable Item C. Lecturers can also include feedback at this level to summarize their advices to the student regarding the whole assessment process.

VI. CASE STUDY: AdESMuS

To evaluate the proposed architecture and verify its usefulness to fulfill the requirements described in Section II, a web system called AdESMuS was developed following CASA.

The evaluation was carried out considering two different complex assessment contexts and analyzing the degree to which the system fulfilled the stablished requirements. To that end, a requirement validation was carried out with real users.

The complex assessment contexts used for the evaluation were selected among the assessment contexts in the University of the Basque Country. This section describes the assessment approaches of both assessment contexts. Then, it presents the results of the evaluation process related to the coverage of the requirements identified in Section II.

A. ASSESSMENT CONTEXTS FOR THE CASE STUDY

The evaluation was carried out in two assessment contexts, each related to a different course. In order to select the courses

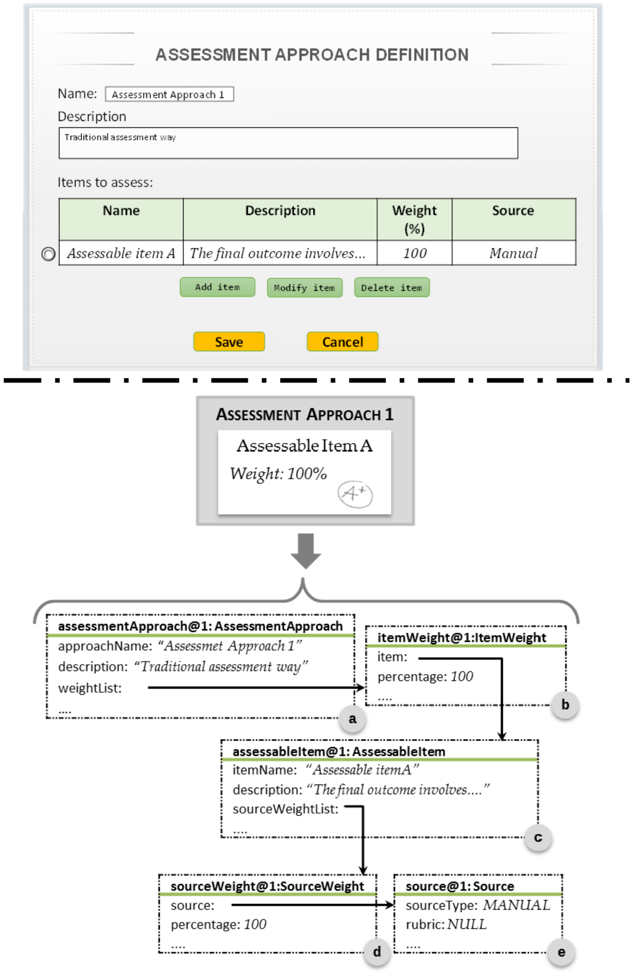

FIGURE 5. **Definition of the first assessment approach: interface (up) and model (down).**

for the evaluation, the authors analyzed the different assessment approaches used in the Computer Engineering degree program at the University of the Basque Country. After this analysis, the course Software Engineering and the Final Year Project assessment context were used as the basis for the evaluation presented in this section. These two contexts incorporate all the assessment aspects that appear in the different courses of this degree program. Moreover, these contexts are so complex that a system that meets their requirements would fulfill all the requirements stablished in Section II.

1) SOFTWARE ENGINEERING

Software Engineering is a required second-year course in Computer Engineering. Currently, the University of the Basque Country is involved in the implementation of new educational methodologies involving formative assessment. Therefore, three assessment approaches are implemented in this course (see Fig. 14). By default, all the enrolled students must follow the continuous assessment approach. This assessment approach comprises four assessable elements: three exams and the development of a game. Each of the

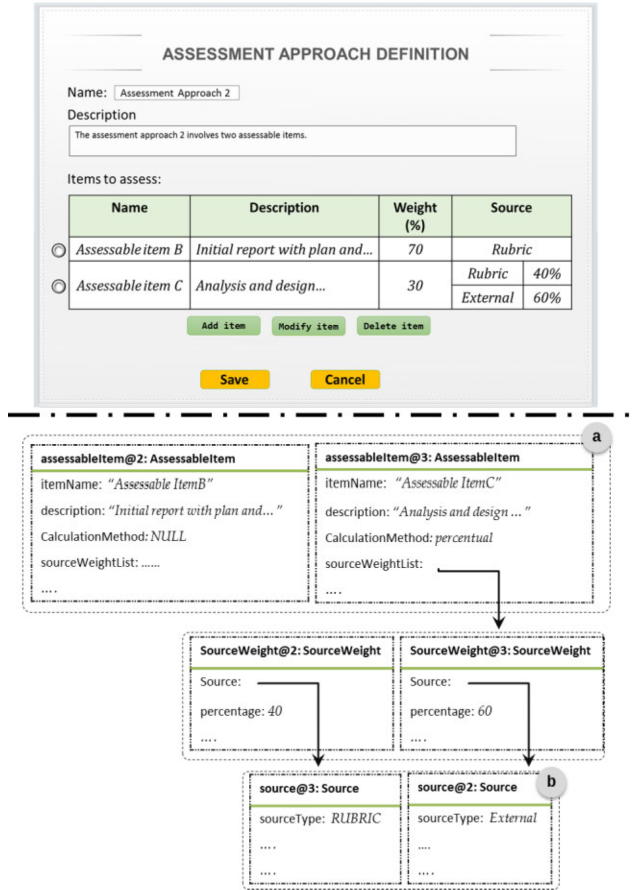

FIGURE 6. **Definition of the second assessment approach: interface (up) and model (down).**

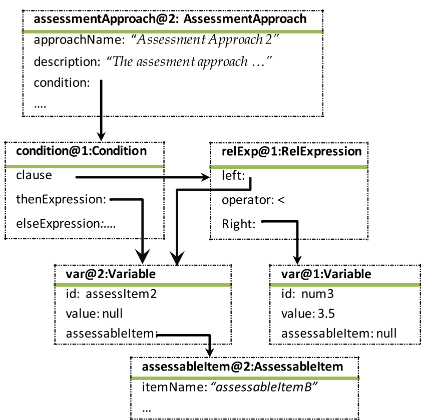

FIGURE 7. **Example of conditions for the assessment approaches.**

exams evaluates different parts of the course (design patterns, software design and development methodologies), each having a different weight in the final grade. In the game development project students apply all the concepts and techniques learned throughout the course. However, some students cannot follow this assessment approach (for instance, some students are working or have been ill during the semester, which prevented them from doing the project). Students in those

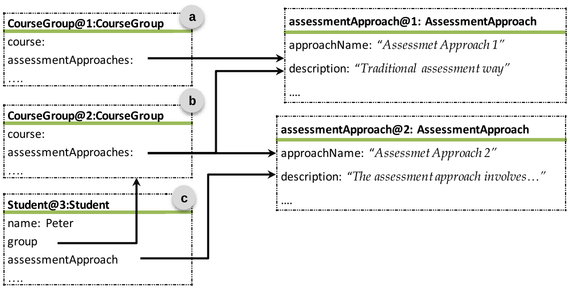

FIGURE 8. **Establishment of assessment approaches.**

situations can apply for an alternative assessment approach that entails just a final exam (Final assessment approach).

Students that do not pass the course following the regular assessment approaches described above have the option of requesting an extra assessment at the end of the term. This extra assessment approach comprises a final exam that evaluates all the theoretical aspects of the course and the game development.

In general, most exams are manually assessed, as shown in Fig. 14. However, the assessable elements that involve the assessment of design aspects are assessed using rubrics. This way, the Design Exam, the Extra Exam and the Game implementation contain a rubric as a grade source. The Extra Exam, which comprises all the theoretical aspects assessed, contains two grade sources that are combined to obtain the final grade. The Game Development assessable evaluates the design aspects according to a Rubric, but it also considers other aspects such as team work and the capability of using a version management system or the implementation of a development methodology. To assess those aspects, information gathered from other two sources, GitHub and IceScrum, is used. In order to compute the Game Development score for the final grade, in those approaches where it is considered, students have to achieve a minimum performance level (3.5 in a 10-point scale) on the exams.

The assessment process in this context, requires the system to fulfill some of the requirements identified in Section II. The assessment plan in the Software Engineering course entails different assessment approaches (*DiffApproaches* requirement), and the students have the chance to move from one approach to another at any time (*Change* requirement). This course also includes some assessable items that are evaluated using rubrics (*Rubrics* requirement). The grade of one of the assessable items, the Game Development, is calculated by combining data from two different sources (*MultipleSources* requirement). Moreover, in those approaches where the Game Development is considered, a minimum grade on the exams is required to compute the final grade (*CondAssessment* requirement).

2) FINAL YEAR PROJECT

The Final Year Project (FYP) is required piece of work that is done at the end of the fourth year of the Computer Engineering degree program at the University of the Basque Country. Currently, two assessment approaches coexist in this context. In the first approach, called Overall Impression, the assessment of FYPs is carried out by an evaluation committee composed of three lecturers that give a single grade, taking into account the final outcome of the work. This final outcome is composed of a written report and an oral presentation. The second approach, the Continuous Assessment Approach, is rooted in the methodology defined in [27] and includes five assessable items. Each of the assessable items in the Continuous Assessment Approach has a different weight in the final grade. Furthermore, the assessment of the assessable items is carried out by different people. Some of the items are evaluated by the project supervisor while others are evaluated by an evaluation committee. To assess the elements, the use of analytic and atomistic rubrics [28] has been adopted; the evaluators can select the kind of rubric they want to use for each element according to their preferences. Fig. 15 shows a summary of the two possible assessment approaches for FYPs.

Therefore, the needs identified in the FYP course are related to 5 of the requirements identified in Section II. The assessment plan of the Final Year Projects implies two assessment approaches (*DiffApproaches* requirement) that remain fixed during the course. The marks can come from two different sources: manually or rubrics (*Rubrics* requirement). In this case, two types of rubrics have been defined, and at execution time the evaluators can select the rubric type to be used (*SelectTools* requirement). Finally, not all the items are assessed by the same people. Some of the items are assessed by the supervisor of the work, whereas the remaining elements are assessed by an evaluation committee (*Committee* requirement). Even if more than one person is involved in the assessment of a Final Year Project, each evaluator can only access his or her own grades, maintaining the privacy of the assessments (*Privacy* requirement).

B. RESULTS

Next the results of the evaluation are depicted.

The validation of the requirement fulfillment was carried out in the two assessment contexts previously presented. As it has previously been depicted, those contexts comprise all the requirements stablished in Section II and allow us to analyze their coverage.

In the first assessment context, the Software Engineering course, each user was provided with a form containing a set of questions related to the fulfillment of the requirements involved in this context (see Section II). Users had to indicate whether or not AdESMuS fulfilled each of them while defining the course assessment approaches. An excerpt of the validation form is shown in TABLE 1 together with the relation among its questions and the defined requirements.

Three lecturers that teach the course took part in the evaluation, and all of them indicated that AdESMuS fulfilled the defined validation criteria. Therefore, AdESMuS received a 100% fulfillment rate for its use in the first context.

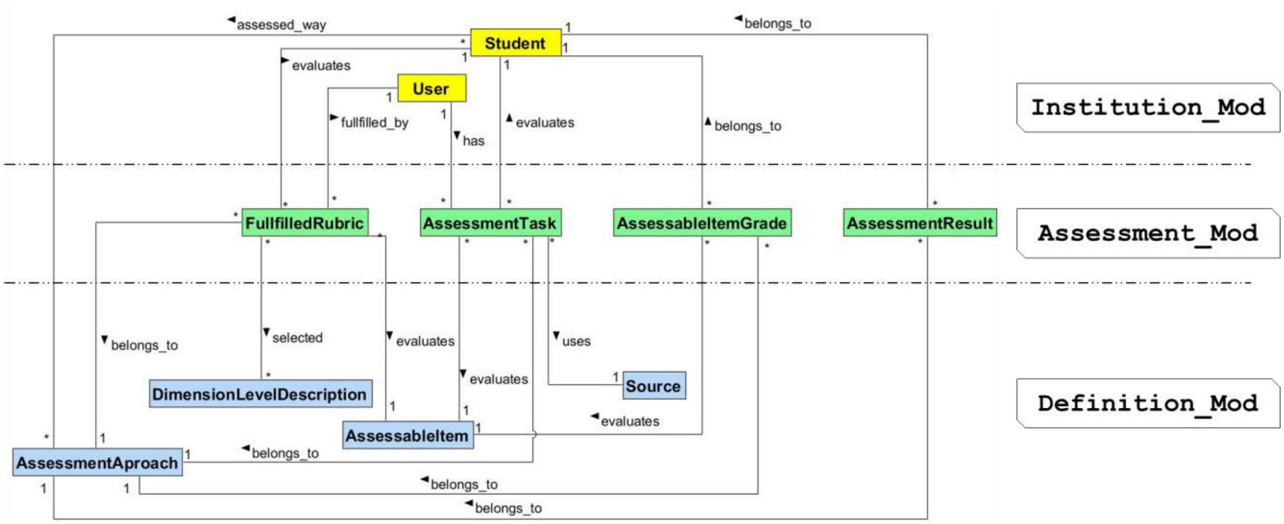

FIGURE 9. UML representation of Assessment_Mod and its relationships.

TABLE 1. Form for the requirement fulfillment validation in the sw engineering course, excerpt.

Requirement	Criteria	Fulfillment	
		YES	NO
MultipleSources	The system allows me to define a final grade that combines different sources of information		
Rubrics	The system allows me to use rubrics to assess assessable items		
CondAssessment	The system allows me to define conditions to calculate the final grade of the course		
DiffApproaches	The system allows me to define the different assessment approaches of the course.		
	The system allows me to define the different assessable items of each assessment approach.		
	The system allows different assessable items to be defined.		
Change	The systems allows me to change the assessment approach of the students at any time		

TABLE 2. Requirement validation form for the assessment of FYPs, excerpt.

Requirement	Criteria	Fulfillment	
		YES	NO
SelectTools	The system allows me to select the assessment tool for each item at execution time		
Rubrics	The system allows me to manage and use rubrics for the assessment		
DiffApproaches	The system allows to select different assessment approaches in the course for each student		
Committee	The system allows to define different people sets to evaluate each item for each student		
Privacy	The system provides confidentiality of grades		

The evaluation in the second context, the assessment of Final Year Projects, analyzed the fulfillment of the requirements required for this context (see Section II and TABLE 2).

Fifteen different lecturers that had used AdESMuS to perform the evaluation of more than 80 FYPs carried out the requirement fulfillment analysis. The fifteen lecturers participating in the assessment tasks agreed that the system

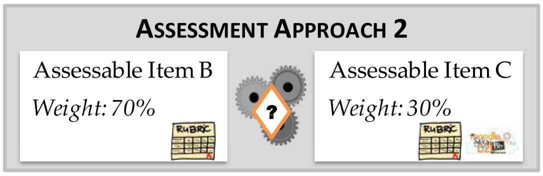

FIGURE 10. Detail of the second assessment approach.

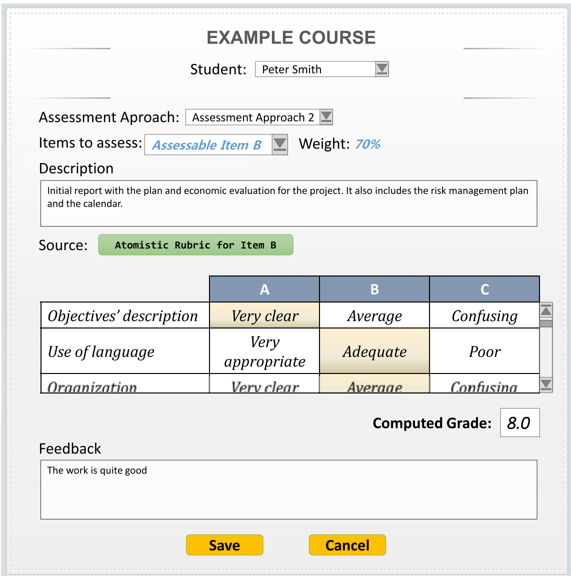

FIGURE 11. Interface related to the AssessmentTask to evaluate AssessableItem B.

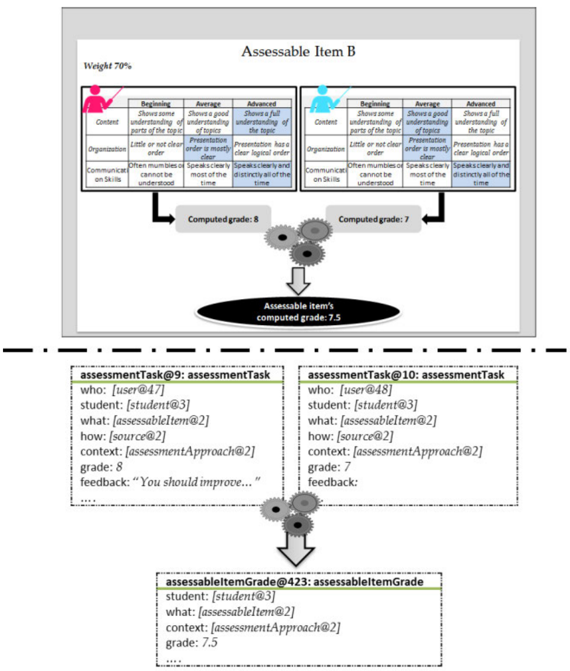

FIGURE 12. Example of assessable item grade: schema (up) and model (down).

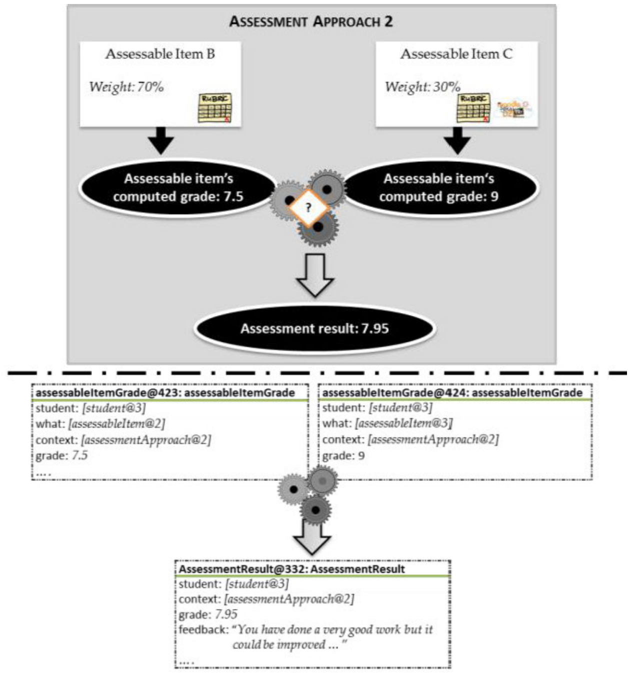

FIGURE 13. Example of assessment result: schema (up) and model (down).

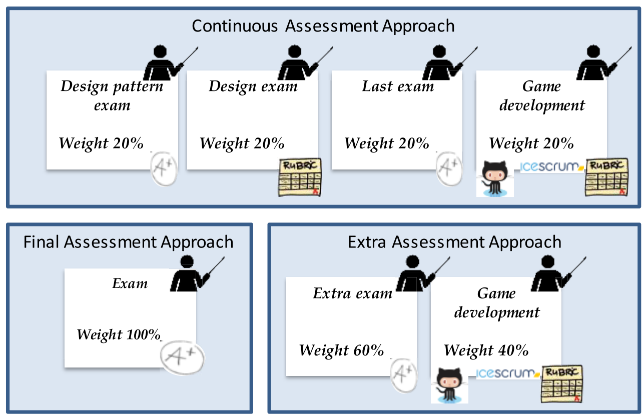

FIGURE 14. Available assessment approaches for the course software engineering.

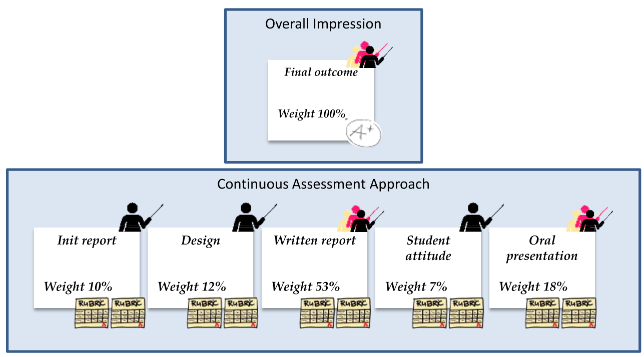

FIGURE 15. Available assessment approaches for final year projects.

fulfilled the established requirements; therefore, the fulfillment rate for using the system to perform evaluations also achieved 100%.

As all the requirements identified in Section II are covered in these two assessment contexts, and the evaluation results have been very positive, it can be said that AdESMuS supports all the defined requirements. Moreover, none of the teachers identified new requirements or needs.

Therefore, the architecture proposed in this paper (CASA) is appropriate to deploy systems that support complex assessment processes.

VII. CONCLUSION

Designing a good assessment policy is critical for any instructional design. Many assessment scenarios exist, each with its own peculiarities. In our work, after studying the needs of many different assessment contexts, a set of requirements that any system that supports complex assessment processes should fulfill was defined and presented.

As none of the existing and analyzed systems was able to give answer to all the defined requirements, the CASA architecture was specified to develop systems to fulfill all the identified requirements. This paper presented this architecture and the three knowledge models defined to feed it: *Institution_Mod*, *Definition_Mod* and *Assessment_Mod*. To evaluate the architecture, a system was developed following the guidelines of the CASA architecture using Node.js and Angular technologies.

This system was used at the University of the Basque Country in two different assessment contexts that were selected due to their complexity: the Software Engineering and the Final Year Project courses. After using the system to define each course's assessment scenarios and to evaluate the students, the involved lecturers were asked to evaluate the system's usefulness. All the results were very positive and allowed us to demonstrate that the CASA architecture is valid to define systems that cover all the defined requirements.

Taking into account these positive results, we plan to extend the use of the system to other courses and to other degree programs broadening the evaluation of the CASA architecture's validity to support complex assessment scenarios.

Future work entails integrating new modules in the architecture to extend its functionalities. One of our priorities is to add a module to incorporate Learning Analytic techniques [13] in CASA. This kind of techniques can provide users with a deeper insight in student performance and allow lecturers to identify learners that are at risk of failure in order to take remediation actions. We also plan to develop new systems using the CASA architecture and different technologies such as mobile apps.

REFERENCES

[1] M. D. Merrill, L. Drake, M. J. Lacy, J. Pratt, and I. R. Group, "Reclaiming instructional design," *Educ. Technol.*, vol. 36, no. 5, pp. 5–7, 1996.

[2] E. Forest. Educational Technology. Accessed: Jul. 4, 2017. *ADDIE Model: Instructional Design*. [Online]. Available: http://educationaltechnology.net/the-addie-model-instructional-design/

[3] W. Dick, L. Carey, and J. O. Carey, *The Systematic Design of Instruction*, 7th ed. Upper Saddle River, NJ, USA: Merrill/Pearson, 2009.

[4] M. Vujošević-Janičić, M. Nikolić, D. Tošić, and V. Kuncak, "Software verification and graph similarity for automated evaluation of students–assignments," *Inf. Softw. Technol.*, vol. 55, no. 6, pp. 1004–1016, Jun. 2013, doi: 10.1016/j.infsof.2012.12.005.

[5] M. Taras, "Assessmen-summative and formative-some theoretical reflections," *Brit. J. Educ. Stud.*, vol. 53, no. 4, pp. 466–478, Dec. 2005, doi: 10.1111/j.1467-8527.2005.00307.x.

[6] M. A. Dandis, "Benefits and criticism. Should we support or neglect using rubrics? Evidences from a literature review," *Proc. EDMETIC*, vol. 3, no. 2, pp. 91–113, Jul. 2014.

[7] D. M. Hunter, R. M. Jones, and B. S. Randhawa, "The use of holistic versus analytic scoring for large-scale assessment of writting," *Can. J. Program. Eval.*, vol. 11, no. 2, pp. 61–85, 1996.

[8] D. D. Stevens, A. J. Levi, and B. E. Walvoord, *Introduction to Rubrics: An Assessment Tool to Save Grading Time, Convey Effective Feedback, and Promote Student Learning*, 2nd ed. Washington, DC, USA: Stylus, 2012.

[9] N. Vaughan, "Student engagement and blended learning: Making the assessment connection," *Educ. Sci.*, vol. 4, no. 4, pp. 247–264, Nov. 2014, doi: 10.3390/educsci4040247.

[10] J. R. Quevedo and E. Montanes, "Obtaining rubric weights for assessments by more than one lecturer using a pairwise learning model," in *Proc. Educ. Data Mining (EDM), Cordoba, Spain*, 2009, pp. 289–298.

[11] M. Villamañe, A. Álvarez, M. Larrañaga, and B. Ferrero, "Desarrollo y validación de un conjunto de rúbricas para la evaluación de Trabajos Fin de Grado," *ReVisión*, vol. 10, no. 1, pp. –27, 2017.

[12] J. Samuelsen, W. Chen, and B. Wasson, "Integrating multiple data sources for learning analytics-review of literature," *Res. Pract. Technol. Enhanc. Learn.*, vol. 14, no. 1, p. 11, Dec. 2019, doi: 10.1186/s41039-019-0105-4.

[13] A. Pardo and S. Dawson, "Learning analytics: how can data be used to improve learning practice," in *Measuring and Visualizing Learning in the Information-Rich Classroom*, P. Reimann, S. Bull, M. Kickmeier-Rust, R. Vatrapu, B. Wasson, Eds. Evanston, IL, USA: Routledge, 2016, pp. 41–55.

[14] E. Panadero and A. Jonsson, "The use of scoring rubrics for formative assessment purposes revisited: A review," *Educ. Res. Rev.*, vol. 9, pp. 129–144, Jun. 2013.

[15] S. M. Brookhart, "Appropriate criteria: Key to effective rubrics," *Front. Educ.*, vol. 3, p. 22, Apr. 2018, doi: 10.3389/feduc.2018.00022.

[16] S. M. Brookhart, *How to Create and Use Rubrics for Formative Assessment and Grading*. Alexandria, VA, USA: Assn for supervision & curricu, 2013.

[17] *Senate Handbook: Assessment Rules*, Cranfield Univ., Vellore, India, 2019.

[18] *General Examination and Assessment Regulations for Taught Courses (GEAR)*, Univ. Brighton, Hastings, England, 2017.

[19] C. A. Rideout, "Students–choices and achievement in large undergraduate classes using a novel flexible assessment approach," *Assessment Eval. Higher Edu.*, vol. 43, no. 1, pp. 68–78, Jan. 2018, doi: 10.1080/02602938.2017.1294144.

[20] E. Montenegro and N. A. Jankowski, "Equity and assessment: Moving towards culturally responsive assessment," Nat. Inst. Learn. Outcomes Assessment, Urbana, IL, USA, Occasional Paper 29, 2017.

[21] M. Villamañe, A. Álvarez, and M. Larrañaga, "Supporting complex assessment processes," in *Proc. 5th Int. Conf. Technol. Ecosystems Enhancing Multiculturality*, Spain, 2017, pp. 1–6, doi: 10.1145/3144826.3145368.

[22] E. Valderrama, M. Rullan, F. Sanchez, J. Pons, C. Mans, F. Gine, L. Jimenez, and E. Peig, "Guidelines for the final year project assessment in engineering," in *Proc. 39th IEEE Frontiers Edu. Conf.*, Oct. 2009, pp. 1–5, doi: 10.1109/fie.2009.5350767.

[23] F. Sanchez, J. Climent, J. Corbalan, P. Fonseca, J. Garcia, J. R. Herrero, X. Llinas, H. Rodriguez, M.-R. Sancho, M. Alier, J. Cabre, and D. Lopez, "Evaluation and assessment of professional skills in the Final Year Project," in *Proc. IEEE Frontiers Edu. Conf. (FIE) Proc.*, Oct. 2014, pp. 2352–2359, doi: 10.1109/fie.2014.7044378.

[24] P. M. Papadopoulos, T. D. Lagkas, and S. N. Demetriadis, "How to improve the peer review method: Free-selection vs assigned-pair protocol evaluated in a computer networking course," *Comput. Edu.*, vol. 59, no. 2, pp. 182–195, Sep. 2012, doi: 10.1016/j.compedu.2012.01.005.

[25] *Regulation (EU) 2016/79 General Data Protection Regulation (GDPR)*, Eur. Parliament, Brussels, Belgium, 2016.

[26] R. T. Fielding, "Architectural styles and the design of network-based software architectures," M.S. thesis, Univ. California, Oakland, CA, USA, 2000.

[27] M. Villamane, B. Ferrero, A. Alvarez, M. Larranaga, A. Arruarte, and J. A. Elorriaga, "Dealing with common problems in engineering degrees' final year projects," in *Proc. IEEE Frontiers Edu. Conf. (FIE)*, Oct. 2014, pp. 2663–2670, doi: 10.1109/fie.2014.7044431.,

[28] B. M. Moskal, "Scoring rubrics: What, when and how?" *Pract. Assess. Res. Eval.*, vol. 7, no. 3, pp. 1–5, 2000.

MIKEL VILLAMAÑE received the B.S., M.S. and Ph.D. degrees from the University of the Basque Country (UPV/EHU), in 2002, 2004, and 2017, respectively, all in computer science. Since 2003, he has been an Assistant Professor with the Computer Languages and Systems' Department, Faculty of Engineering in Bilbao, University of the Basque Country (UPV/EHU). He develops research activities within the GaLan Research Group (http://galan.ehu.eus). His research interests include computer aided education, learning analytics, and mobile technologies.

AINHOA ÁLVAREZ received the B.S. and Ph.D. degrees in computer science from the University of the Basque Country (UPV/EHU), in 1999 and 2010, respectively.

Since 2001, she has been an Assistant Professor with the Computer Languages and Systems' Department, Faculty of Informatics, University of the Basque Country (UPV/EHU). She develops research activities within the GaLan Research Group (http://galan.ehu.eus). She is the author of more than 50 publications in the areas of computer aided education. Her research interests include computer aided education and learning analytics.

MIKEL LARRAÑAGA (Member, IEEE) received the Ph.D. degree in computer science from the University of the Basque Country (UPV/EHU), in 2012.

He is currently a Faculty Member with the Department of Computer Languages and Systems, Faculty of Engineering in Vitoria-Gasteiz, University of the Basque Country (UPV/EHU). He has been working in computer-based education area for the last 15 years. He develops research activities within the GaLan Research Group (http://galan.ehu.eus). His current research interests include knowledge acquisition, concept mapping, learning analytics, recommender systems, and intelligent tutoring systems.