Learning Objects and Tests

Mirko Cesarini
Politecnico di Milano
Dip. Elettronica e Informazione
Piazza Leonardo da Vinci, 32
I 20133 Milano, Italy
cesarini@elet.polimi.it

Paolo Mazzoni
Politecnico di Milano
Dip. Elettronica e Informazione
Piazza Leonardo da Vinci, 32
I 20133 Milano, Italy
mazzoni@elet.polimi.it

Mattia Monga
Università di Milano
Dip. di Informatica e Comunicazione
Via Comelico 39/41
I 20135 Milano, Italy
mattia.monga@unimi.it

ABSTRACT
In this paper we discuss how an e-learning environment based on Learning Objects can be extended to cope with testing learners achievements. Provided that the learning management system is powerful enough to be able to handle complex relationships and process support, we found that the learning object framework is expressive enough to encapsulate reusable tests. By leveraging on a workflow engine, we applied our approach to our Virtual Campus e-learning platform to automate as much as possible the management of the test activities, including, as far as possible, evaluation and score notification.

KEY WORDS
Learning Objects, Test, Metadata

1 Introduction

The wide availability of the Internet communication infrastructure has opened new opportunities and challenges for innovative, multimedia, network-wide educational services. These services are often provided by a web-based Learning Management Systems (LMS), that are software applications able to organize and deploy multimedia and hypertextual contents according to the educational needs. In order to increase interoperability among different systems, a number of standard were proposed both for data and the system interface. The Learning Technology Standardization Committee of the IEEE promotes the *Learning Object Metadata* (LOM) standard [3], which specifies a data pattern for describing learning resources through a standardized vocabulary of the subject domain. According to this standard, a *learning object* is defined as any entity, digital or non-digital, that may be used for learning, education, or training. The main goal of a standard like LOM is that learning resources should be regarded as reusable multimedia content components: each learning object is tagged with its metadata and LMS can exploit these for searching or in order to establish relationships among content units. However, reuse a learning object may mean very different things, depending on the nature of the learning object itself [6]. Reusing slides or notes is certainly very different from "reusing" a classroom experiment. Sometimes one wants to reuse just the infrastructure needed to accomplish some learning objective: for example one could reuse a computer supported collaborative work application or a tutoring avatar. Moreover, one wants often reuse collections of learning objects as a whole course.

In the "Virtual Campus" project ongoing at Politecnico di Milano [11], we addressed the problem of encapsulating generic learning activities as learning objects and we built a web based infrastructure able to compose them in coherent workflows [7]. In this paper we will focus on the particular activity of testing learners achievements. A testing activity involves two roles: *learners*, who are requested to perform some task they should have learnt, and *instructors*, who prepare the description of the task and are able to assess the quality of learner performances. Thus, an examination session is composed by several activities: a test has to be prepared by the instructor, it has to be delivered to learners, and finally the "output" produced by learners has to be evaluated by the instructor. We wanted to be able to reuse not just the data instructors used to build their test, but also the delivery and evaluation infrastructure. In the examples shown in this paper we will focus on the simple case in which the evaluation process can be automated by a computer program. In general we aim at addressing all the examination activities in which at least some part of the evaluation activity, for example the marking policy, can be automated and embodied in a computer program. As an example, consider the following scenario: Alice, a computer science instructor, produces a set of problem statements for her students. An actual examination is assembled picking up a number of these statements and delivering them to students (possibly, giving a different one to each of them). Alice has produced a set of test cases for each problem statement and she will consider correct any program which is complaint with them. Moreover, she is going to inspect manually the code that is not working, in order to asses the errors. She marks each exercise e with a real number x_e such that $0 \leq x_e \leq 2$ (the higher the better) and the overall evaluation of a student is the average of the marks in individual exercises weighted on their difficulty. In order to help Alice a learning management system should be able to

1. store problem statements,
2. provide infrastructure to aggregate them in examinations
3. deliver an examination to learners
4. collect artefacts produced by learners
5. provide infrastructure to apply the Alice's evaluation procedure
6. provide infrastructure to apply the Alice's marking policy

Moreover, problem statements, aggregated examinations, evaluation procedures and marking policies should be available for different testing sessions.

This paper is organized as follows: in Section 2 we present the problems that arise in encapsulating tests as Learning Objects, in Section 3 we describe our approach in the context of Virtual Campus platform. Finally in Section 4 we discuss some related works and we draw some conclusions in Section 5.

2 Learning Objects and Tests

While Learning Objects have been well exploited for both learning and teaching activities, they were used only marginally to encapsulate testing material. Encoding a test within a Learning Object presents some peculiarities that need special care.

First of all, a test is normally not reused "as is" in order to limit the likeliness of cheating. Thus, reusing a test needs a small granularity of test material and an infrastructure that enables the composition of learning objects and possibly their adaption.

Second, and most important, the testing activity is intrinsically a *collaborative task* among instructors and learners. For this reason the learning management system has to be able to handle the exchange of data among instructors and learners, while keeping their work synchronized. Most systems simply *broadcast* the content to learners, but this is not enough in the testing context.

Third, instructors would like to store in a Learning Object not just the text of an exam, but all those repetitive tasks related to testing. Summarizing the results, evaluating closed form questions, marking student artifacts, publishing results, are all activities related to an examination that are often general enough to be reused in similar tests. For this, Learning Objects should encapsulate a description of these tasks: in some sense they should be used to store *computations* and not just data.

Our solution builds on our Virtual Campus environment. Our system leverages on the power of an underling workflow management engine in order to provide the notion of process to users [2]. Thus, it is possible to coordinate different activities in coherent collaborative tasks. Typically, activities produce several artifacts that are exchanged among all the actors involved, managed by the system infrastructure. Moreover, the rich conceptual model we developed [9] (starting from the LOM specification [3]) enables the use of Learning Objects to store computations and to relate them to Learning Objects containing data in a straightforward way. However, despite its ability to build and manage complex course with many possible learning paths, the current support for composing tests within the Virtual Campus environment is still poor: tests are modelled as black box components[1]. In the following sections we describe the approach we used.

3 The Evaluation Learning Object

In the Virtual Campus environment instructors define relationships among Learning Objects. These relationships are then used to build the learning paths that the workflow engine enacts after a customization process. As far as concern testing Learning Objects the most important relationships are `IsReqBy`, meaning that a Learning Object is required by an other, and `ReqOnFailure`, meaning that a Learning Object is required by another when the activity related to it completes in an unsuccessful state. The Fig. 1 shows how these relationships can be used to specify a *Test Learning Object* .

The `ReqOnFailure` edge is used to specify which is the next Learning Object to be studied by a student in case of test failure. Moreover, we couple every Test Learning Object with an *Evaluation Learning Object*, introducing the new relation `Evaluates`. Both Learning Objects provide all that is required to specify a testing activity, the idea is to confine within the Test Learning Object the specific content of a test (namely questions data and knowledge on how to evaluate the related answers, not necessarily by using an automated procedure), while the Evaluation Learning Object should contain all the reusable computations: the routines for the Virtual Campus environment on how to collect the learners' answers (the aforementioned collaborative tasks among instructors and learners), the algorithms to be used during the automatic evaluations, the workflow to be used to coordinate the evaluator activities (when manual assessment is required) and the marking policy. In this way a teacher that needs to author many tests about the same subject, has to change only the content (namely to produce different Test Learning Objects) while the same Evaluation Learning Object can be reused all the times. An alternate approach is still possible: the same Test Learning Object can be used for two or more classes, but using different Evaluation Learning Objects it is possible to evaluate differently the two classes. For example "Java Programming Test" can be used for both computer science and business students, but using different Evalu-

[1] The composition infrastructure of Learning Objects has to be extended to ease the reusability of tests. Currently in Virtual Campus you can just compose simple tests in a big one, and no support is provided to randomize the composition or other similar tasks.

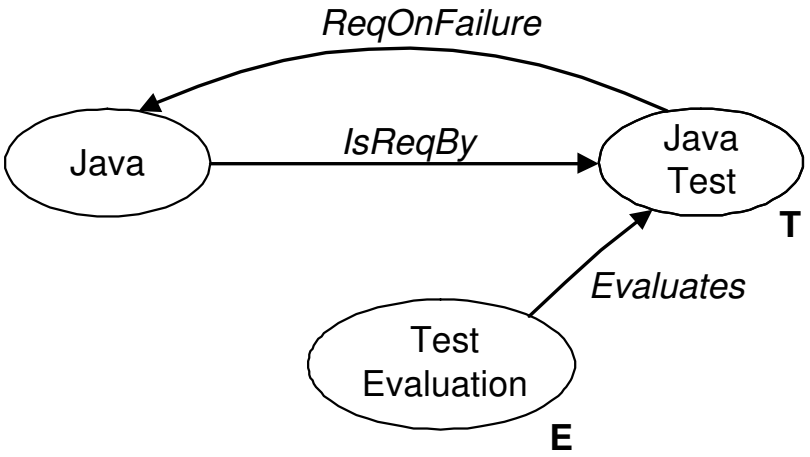

Figure 1. An example of Learning Object, Test Learning Object and Evaluation Learning Object

ation Learning Objects it is possible to evaluate business students less strictly than computer science ones.

The evaluation information could be stored into the Test Learning Object without introducing any new relationship, however in this way flexibility and reusability are greatly reduced. Different Evaluation Learning Objects can be used for the same Test Learning Object (achieving different evaluations every time), but no more than one evaluator per time can be assigned to a test. However, it is worth noting that an Evaluation Learning Object cannot be freely bound to any Test Learning Object, due to the fact that the Evaluation Learning Object should be able to interpret the data contained into the Test Learning Object. The authoring environment takes care that only type compliant evaluators are coupled to a given test by consulting the meta-data contained into the Learning Objects. For a complete details about Evaluation Learning Object meta-data see Tab. 3.

During the course customization process an instructor can be associated to an Evaluation Learning Object requiring human based evaluation. Once this information is provided, a workflow is generated and used to coordinate the tasks where people is involved. As a result the process of collecting the learners' materials, presenting it to the instructor, collecting the scores and notifying the students is performed by the workflow management engine that powers the Virtual Campus architecture. This feature is very useful when the testing activities involve more than one evaluator. In the example of Fig. 2 suppose the evaluation should be performed as follows: the first test is automatically evaluated and only succeeding students can go on with the remaining tests, which are evaluated by teachers. The workflow will take care of presenting the answers to the teachers without having them to poll the students.

The introduction of Evaluation Learning Objects enables an easy solution to the problem of aggregating into a single score the results of inner Test Learning Objects composing a complex one. Consider the example in Fig. 2, the final score could be an average of the four inner test results (even if not all are positive scores), but also it could be an average of the last three evaluations, with the constraint that the student succeeded the first test. Coupling an Evaluation Learning Object with a complex Test Learning Object allows to specify the aggregation policy to be used.

3.1 Virtual Campus Integration

Our approach integrates smoothly with the rich conceptual framework provided by the Virtual Campus platform. We are currently trying to define a taxonomy of possible test types, in order to be able to couple tests with proper generic evaluators; this taxonomy is in someway similar to IMS QTI's one [4], but we are trying to define it adequately to our needs. We already identified the following types of questions:

- Open answer questions: questions that require answers made by sentences having no fixed structure.
- Closed answer questions: questions provided with a set of answers, that student must state if they are true or false.
- Numerical based questions: questions that require a numeric answer.
- Code based questions: questions that require the students to write a piece of code.

The purpose was not only to create a test type reference framework, but also to identify which kind of tests can be applied with a particular focus on automating as much as possible the evaluation processes. (See [8] for more details). Three main evaluation methods were identified: *automated correction*, *manual correction* and *hybrid correction*. *Closed answer questions* and *numerical based ques-*

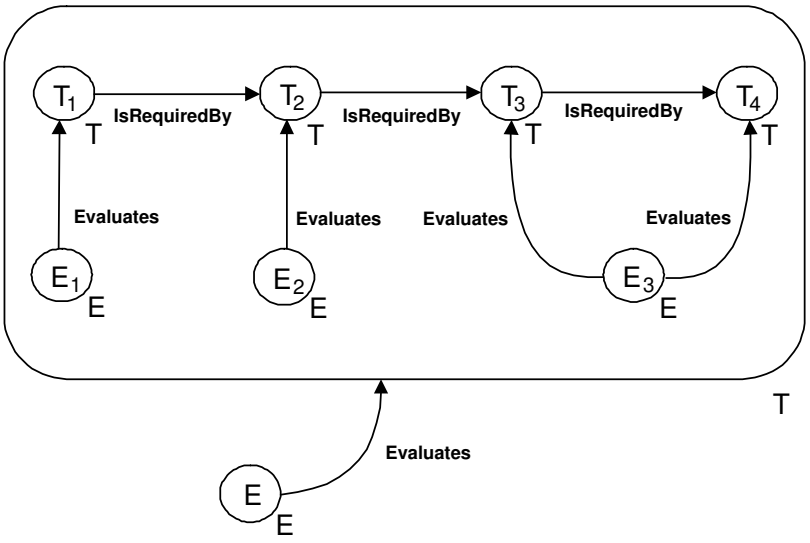

Figure 2. An example of complex Test Learning Object

Test-Type	states the Test Learning Object type that can be evaluated
Human-Intervention-Required	specifies if human intervention is required
Manage-Complex-LO	specifies if complex Learning Object can be managed.

Table 1. Some of the Evaluation Learning Object meta-data

tions can be automatically corrected, supposing that knowledge about the correct and incorrect answers is provided, while *open answers* and *code based questions* require the human intervention.

Evaluation Learning Object meta-data are used to store such a information. In this way, when generating the evaluation workflow the system will be able to recognize if it must take into account human operation.

4 Related Works

Many e-learning tools and services have been developed to manage content delivery and testing activities. The IMS Global Learning Consortium has developed a standard for testing activities, in particular the IMS Question and Test Interoperability [4] specification describes a basic structure for the representation of questions and their corresponding answers, aiming at enabling their exchange between different LMSs and authors. IMS work is very interesting and other educational tools (as Perception by QuestionMark [10] or web-based tools by Desire2Learn [5]) have been developed with reference to this work, but the information model underlying the IMS specification is not using Learning Objects, instead they propose their own model (called ASI Information Model) composed by elements that are not defined by the IEEE Learning Object standard. Within the Virtual Campus project, we preferred to use the Learning Object model to represent testing activities (test contents, fruition and evaluation) without introducing new elements in order to exploit Virtual Campus model richness and such that tests can be easily exchanged as Learning Objects between different LMSs without any substantial modification.

5 Conclusions and Future Works

The introduction of Evaluation Learning Objects allowed us to enhance testing activities in the Virtual Campus project. The Virtual Campus conceptual model, based on IEEE Learning Object standard, has proven to be a framework powerful enough to to handle the complex relationships and to supply the process support needed by tests. By leveraging on a workflow engine, our approach eases the automation of test activities, including, as far as possible, evaluation and score notification. In particular by confining within an Evaluation Learning Object all the repetitive tasks related to assessment, an improved test reusability is achieved. Evaluation Learning Objects allow also to insert information about people involved in the assessment activities during the authoring of a course learning path. In this way, the Virtual Campus environment can produce a workflow that will manage such activities, easing the coordination among learners and evaluators.

Moreover, we enriched the definition of complex Test Learning Objects that has allowed us to embody in Virtual Campus more complex testing activities, not only single tests or questionnaires, but complete exams composed by

different parts.
Another interesting problem related to test activities is the assessment of test results; we defined a taxonomy of test types (see [8]) following IMS QTI specification [4]: some of them may be corrected automatically (for example true/false questions or single/multiple choice questions), for others only a human teacher can establish if the test answers are correct. With Evaluation Learning Objects we can specify how to evaluate a test that can be automatically corrected, nevertheless the evaluation of an open question may be performed only by a human teacher. We are currently working on automatic code correction. However, if a tool of automatic code correction would be desirable, it is worth noting that it is impossible to establish automatically program correctness, so human interaction with the teacher is advised. In this scenario, the teacher configures adequately the tools that automatically produces test cases and results, reducing teacher activities, but the final evaluation is given by the teacher her/himself, in order to produce a more palatable assessment.

References

[1] D. Balzarotti, C. Ghezzi, and M. Monga. Freeing cooperation from servers tyranny. In E. Gregori, L. Cherkasova, G. Cugola, F. Panzieri, and G. P. Picco, editors, *Web Engineering and Peer-to-Peer Computing*, volume 2376 of *LNCS*, pages 235–246. Springer-Verlag, 2002.

[2] M. Cesarini, M. Monga, and R. Tedesco. Carrying on the e-learning process with a workflow management engine. Submitted for publication.

[3] I. L. T. S. Committee. Draft standard for learning object metadata. `http://ltsc.ieee.org/`.

[4] I. I. G. L. Consortium. Ims question and test interoperability: Asi information model specification. `http://www.imsglobal.org/question/index.cfm`.

[5] Desire2learn. `http://www.desire2learn.com/products/products.html`.

[6] E. Duval and W. Hodgins. A LOM research agenda. In *Proceedings of the 12th International World Wide Web Conference*, Budapest, Hungary, May 2003.

[7] L. Mainetti, M. Monga, and L. Sbattella. A Virtual Campus for tethered and untehered scenarios. In *Proceedings of Frontiers in Education*, Boston, Massachusetts, Nov. 2002. IEEE.

[8] G. D. G. Matteo Freddi, Riccardo Garofano. Progettazione e realizzazione di uno strumento per la correzione automatica degli esami. Master's thesis, Politecnico di Milano, Dipartimento di elettronica e informazione, 2003.

[9] E. D. Nitto, L. Mainetti, M. Monga, L. Sbattella, and R. Tedesco. Supporting authoring and fruition of learning objects: the virtual campus approach. In preparation.

[10] Questionmark. `http://www.questionmark.com/us/home.htm`.

[11] A Virtual Campus @ Politecnico di Milano. `http://www.elet.polimi.it/res/vcampus/`.