

# A programming exercise evaluation service for Mooshak

José Paulo Leal <zp@dcc.fc.up.pt> and Ricardo Queirós <ricardo.queiros@eu.ipp.pt>

In this paper we describe the modelling and design phases of a web service to expose the evaluation functions of Mooshak - a contest management web system that is being used also as an eLearning tool for computer science courses. The purpose of exposing the evaluation functions of Mooshak as services is the integration with other systems that do not require its web interface. A Learning Management System (LMS) is an example of such systems. An LMS plugin can embed a program editor on its web interface and use such a service to evaluate students' programs. A similar example is an Integrated Development Environment (IDE) with a plugin supporting direct submission to Mooshak.

To model a programming exercise evaluation service in the context of a network of eLearning systems we used an eLearning framework. An eLearning framework supports the creation of flexible eLearning processes using service oriented approaches. Based on a survey we identified the e-Framework as one of the most prominent eLearning framework initiatives. The e-Framework provides a number of key concepts such as a service genre, service expression and service usage model (SUM). It provides also a knowledge base with definitions of many genres, expressions and SUMs that are recurrently used in eLearning. An outcome of this work was a contribution to the e-Framework of the definition of an evaluation service genre, identifying its purpose and functions. This service genre was instrumental to propose a service usage model involving other service genres, namely the Learning Objects Repository (LOR) and the LMS. We contributed also with a definition of a service expression for the evaluation of programming exercises. An implementation of the proposed service expression evaluates an attempt to solve a programming exercise and produces a detailed report. This evaluation report includes information to support exercise assessment, grading and/or ranking by client systems. Nevertheless, the report itself is not an assessment, since it does not include a grade and it does not compare students, contestants or teams. As part of the service expression proposed to the e-Framework we provided a definition of the API in the SOAP and REST flavors. We defined also the Evaluation Response Language (ERL) used in the responses of the evaluation service.

After modelling the evaluation service we redesigned Mooshak to support it. Our main design options were to support the evaluation service concurrently with the existing web features and to minimize the changes to the current code base. Mooshak has a single front controller to process all HTTP requests related to its web interface. The controller is responsible for tasks such as user authentication/authorization, session management, logging and auditing. The evaluation service is completely stateless and the authentication and authorization requirements are different. Thus, we designed a second front controller specifically to process the three types of service requests defined in the modelling stage. The new commands introduced by the model - `listCapabilities`, `evaluateSubmission` and `getReport` - are implemented reusing the business and data layers already implemented in Mooshak.

Any contest managed by a Mooshak installation can be set to support the evaluation service. To process a `listCapabilities` request Mooshak lists all programming languages configured for contests accessible as services. An `evaluateSubmission` request receives as parameters a capability, a programming problem and a program. The programming problem can either be a reference to a programming problem already loaded into Mooshak, or an URL of a learning object containing a programming problem in a remote LOR. In this last case the programming problem is downloaded and converted to the internal representation of Mooshak. The submitted program code is evaluated in the usual way with a small twist. Program evaluation in Mooshak produces a report using an HTML template. To evaluate a program code received by the `evaluateSubmission` request, this template is replaced by a XML template compatible with the ERL language defined for service responses.

We are currently implementing the described evaluation service in Mooshak. Certain parts, such as the communication with external repositories and conversion of programming problems to the internal format of Mooshak, are already being tested. When completed this service is expected to integrate an heterogeneous network of eLearning types of systems, including the LMS, the IDE and the LOR.