Programming Exercises Interoperability Language

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

The concept of Learning Object (LO) is fundamental for producing, sharing and reusing content in eLearning. In essence a LO is a container with educational material and metadata describing it. Since most LOs just present content to students they contain documents in presentation formats such as HTML and PDF, along with metadata on these documents and the LO itself using generic standards such as LOM and SCORM. If the LO includes an exercise for automatic evaluation then a resource must formally described it. The Question and Tests Interoperability (QTI) is an example of standard for this kind of definitions that is supported by several eLearning systems. However, QTI was designed for questions with predefined answers and cannot be used for complex evaluation domains, such programming exercise evaluation.

A programming exercise requires a collection of files (test cases, solution programs, etc) and special data (compilation and execution lines, feedback messages, etc). These resources are interdependent and processed in different moments of the programming problem life-cycle. In a SOA based eLearning system these tasks may even be processed in different services, which presents a risk of lack of consistency in the data describing a programming problem. For instance, imagine that the restrictions presented on the problem description (say, the size of the input data) do not match those actually used on test cases, as it unfortunately happens sometimes. In this case a student or team that ignores an efficient algorithm to solve the problem may not attempt at all, while a less knowledgeable student or team may manage to solve it with a naive or brute-force approach.

The tasks of the programming exercise life-cycle, when resources from the LO must be processed, are the following:

1. **Selection** - of a programming exercise based on its metadata from a repository of learning objects;
2. **Presentation** - of the problem statement in different formats (e.g. HTML, PDF);
3. **Generation** - of input and output files from a input description and a solution program, at problem creation time;
4. **Validation** - of the programming exercise to verify that tests cases (generated of created manually) meet the specification presented on the exercise description;
5. **Evaluation** - of a submission against test cases, performed on an evaluation engine;
6. **Feedback** - sent to the learner when errors are detected;
7. **Classification** - of a submission based on the result of each test.

The best approach to ensure the consistency and completeness of data involved in these tasks is to avoid fragmenting it. For this purpose we developed and XML dialect called PExIL, standing for Programming Exercises Interoperability Language. The aim of PExIL is to consolidate in a single document all the data required in the programming exercise life-cycle. The formal definition of PExIL uses XML Schema.

A PExIL document is an autonomous part of a LO and should not be confused with its metadata. The purpose of a PExIL document is to provide data to different agents processing a programming problem while ensuring its consistency and completeness. Nevertheless, PExIL was designed to simplify the extraction of metadata on the programming problem and to be easily integrated into an LO. We are taking as reference the IMS Common Cartridge (IMS CC) specification. The IMS CC specification aims to formalize the organization and distribution of digital learning worldwide and fortunately was designed to be straightforward to extend, meeting the needs of a target user community through the creation of application profiles.

The authors are currently developing a Java package with a collection of tools to process PExIL documents. For instance, this package includes tools to generate HTML and PDF descriptions from PExIL, as well as a tool to generate test files from input descriptions in PExIL and a solution program referred in this document. Another tool validates the resources that compose the LO against the specification in the PExIL document. These tools are being used to improve and validate the language itself but we expect them to be useful also in the development of several components that will integrate an eLearning system for programming, including an exercise authoring environment and an exercise resolution environment, among others.