Chapter 3
Making Programming Exercises Interoperable with PExIL

Ricardo Queirós
CRACS and ESEIG/IPP, Porto, Portugal

José Paulo Leal
University of Porto, Portugal

ABSTRACT

Several standards have appeared in recent years to formalize the metadata of learning objects, but they are still insufficient to fully describe a specialized domain. In particular, the programming exercise domain requires interdependent resources (e.g. test cases, solution programs, exercise description) usually processed by different services in the programming exercise lifecycle. Moreover, the manual creation of these resources is time-consuming and error-prone, leading to an obstacle to the fast development of programming exercises of good quality. This chapter focuses on the definition of an XML dialect called PExIL (Programming Exercises Interoperability Language). The aim of PExIL is to consolidate all the data required in the programming exercise lifecycle from when it is created to when it is graded, covering also the resolution, the evaluation, and the feedback. The authors introduce the XML Schema used to formalize the relevant data of the programming exercise lifecycle. The validation of this approach is made through the evaluation of the usefulness and expressiveness of the PExIL definition. In the former, the authors present the tools that consume the PExIL definition to automatically generate the specialized resources. In the latter, they use the PExIL definition to capture all the constraints of a set of programming exercises stored in a learning objects repository.

DOI: 10.4018/978-1-4666-2669-0.ch003
INTRODUCTION

The concept of Learning Object (LO) is fundamental for producing, sharing, and reusing content in eLearning (Friesen, 2005). 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, and metadata describing these documents using mostly Learning Objects Metadata (LOM) or other generic metadata format. When a LO includes exercises to be automatically evaluated by an eLearning system, it must contain a document with a formal description for each exercise. The Question and Tests Interoperability (QTI) (IMS Global, 2012) is an example of a 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 as the programming exercise evaluation (Queiros & Leal, 2009). A programming exercise requires a collection of files (e.g. test cases, solution programs, exercise descriptions, feedback) and special data (e.g. compilation and execution lines). These resources are interdependent and processed in different moments in the lifecycle of a programming exercise.

The lifecycle comprises several phases: in the creation phase the content author should have the means to automatically create some of the resources (assets) related to the programming exercise such as the exercise description and test cases and the possibility to package and distribute them in a standard format across all the compatible systems such as Learning Management Systems (LMS) and Learning Object Repositories (LOR); in the selection phase the teacher must be able to search for a programming exercise based on its metadata from a repository of learning objects and store a reference to it in a learning management system; in the presentation phase the student must be able to choose the exercise description in its native language and a proper format (e.g. HTML, PDF); in the resolution phase the learner should have the possibility to use test cases to test an attempt to solve the exercise and the possibility of automatically generating new ones; in the evaluation phase the evaluation engine should receive specialized metadata to properly evaluate the learner’s attempt and return enlightening feedback. All these phases require a set of interdependent resources and specialized metadata whose manual creation would be time-consuming and error-prone.

This chapter focuses on the definition of an XML dialect called PExIL (Programming Exercises Interoperability Language). The aim of PExIL is to consolidate all the data required in the programming exercise lifecycle, from when it is created to when it is graded, covering also the resolution, the evaluation and the feedback. We introduce the XML Schema used to formalize the relevant data of the programming exercise lifecycle. The validation of this approach is made through the evaluation of the usefulness and expressiveness of the PExIL definition. In the former, we use a PExIL definition to generate several resources related to the programming exercise lifecycle (e.g. exercise descriptions, test cases, feedback files). In the latter, we check if the PExIL definition covers all the constraints of a set of programming exercises stored in a learning objects repository.

The remainder of this chapter is organized as follows. Section 2 traces the evolution of standards for LO metadata and packaging. In the following section, we present the PExIL schema with emphasis on the definitions for the description, test cases, and feedback of the programming exercise. Then, we evaluate the definition of PExIL and conclude with a summary of the main contributions of this work and a perspective on future research.
17 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage: www.igi-global.com/chapter/making-programming-exercises-interoperable-pexil/73172?camid=4v1


Related Content

**Design Recovery of Web Applications Transactions**
www.igi-global.com/chapter/design-recovery-web-applications-transactions/4928?camid=4v1a

**A Unified Software Reengineering Approach towards Model Driven Architecture Environment**
Bing Qiao, Hongji Yang and Alan O’Callaghan (2005). *Software Evolution with UML and XML* (pp. 55-100).
www.igi-global.com/chapter/unified-software-reengineering-approachtowards/29610?camid=4v1a

**Preservation of Data Warehouses: Extending the SIARD System with DWXML Language and Tools**
www.igi-global.com/chapter/preservation-data-warehouses/73177?camid=4v1a

**Towards a UML Profile for Building on Top of Running Software**
www.igi-global.com/chapter/towards-uml-profile-building-top/30551?camid=4v1a