Modelling text file evaluation processes

José Paulo Leal[1] and Ricardo Queirós[2]

[1]CRACS & INESC-Porto LA, Faculty of Sciences, University of Porto, Rua do Campo Alegre, 1021 4169-007 Porto PORTUGAL
zp@dcc.fc.up.pt
[2]CRACS & INESC-Porto LA, Faculty of Sciences, University of Porto, Rua do Campo Alegre, 1021 4169-007 Porto PORTUGAL
ricardo.queiros@eu.ipp.pt

Abstract. Text file evaluation is an emergent topic in e-learning that responds to the shortcomings of the assessment based on questions with predefined answers. Questions with predefined answers are formalized in languages such as IMS Question & Test Interoperability Specification (QTI) and supported by many e-learning systems. Complex evaluation domains justify the development of specialized evaluators that participate in several business processes. The goal of this paper is to formalize the concept of a text file evaluation in the scope of the E-Framework – a service oriented framework for development of e-learning systems maintained by a community of practice. The contribution includes an abstract service type and a service usage model. The former describes the generic capabilities of a text file evaluation service. The later is a business process involving a set of services such as repositories of learning objects and learning management systems.

Keywords: e-learning, SOA, interoperability.

1 Introduction

The majority of e-learning systems include the automatic evaluation of quizzes as a feature. Quizzes have the advantage of being generic and usable in any learning domain. However, the most effective types of exercises in any learning domain, both for knowledge acquisition and for student grading, are seldom quizzes. For instance, it is hard to imagine learning computer programming without actually programming. An attempt to solve a programming exercise is written in a specific language (a programming language) that cannot be evaluated simply by comparing it with predefined answers, as in quiz evaluation.

Text file automatic evaluation differs significantly from quiz evaluation based on the IMS Question & Test Interoperability (QTI) specification. QTI describes a data model for questions and test data and, since version 2.0, extends the IEEE Learning Object Metadata (LOM) standard with its own meta-data vocabulary. QTI was designed for questions with a set of pre-defined answers, such as multiple choice, multiple response, fill-in-the-blanks and short text questions. It supports also long text answers but the specification of their evaluation is outside the scope of the QTI. In

fact, the evaluation of text files requires extra resources and specialized metadata. For this reason the authors consider that QTI is not adequate for text file automatic evaluation, as would be expected since it was not designed for this purpose. Extensions to learning object specification have to be developed to support text file evaluation [4]. Unlike text file evaluation, QTI quiz evaluation is integrated in many e-learning systems, especially in Learning Management Systems (LMS). On one hand text file evaluation is too specialized to justify its integration in a general LMS. On the other hand, provided as a service it can used by many kinds of systems. For instance, a programming evaluation service may have as clients programming assignment managers, self-evaluation tools and contest management systems. Its services can also be used by plug-ins of extensible systems, such as LMS or Integrated Development Environments (IDE).

The motivation for this research comes from the experience of the authors with systems such as Mooshak [5] and EduJudge [3]. The former is a contest management system for ICPC contests that is being used since 2002 also as an e-Learning tool in computer programming courses. The later is a system developed for enabling the access of LMS to the collection of programming exercises of the UVA on-line judge. Both systems include automatic evaluation components that if recast as services could provide their functions to different types of e-Learning systems.

The goal of this paper is to formalize services and processes involving text file evaluation in the scope of an e-learning framework. The purpose of an *e-learning framework* is to support the integration of systems within educational institutions using a Service Oriented Architecture (SOA) [1]. In this paper the authors report on the contribution to a particular e-learning framework – the e-Framework. This framework was selected based on a previous survey [2] since it has an active community of practice and accepts abstract definitions of services as contributions.

The contribution described in this paper includes an abstract definition of a type of service and a description of a business process model. The service modelled by the proposed definition receives a text file with an attempt to solve an exercise and produces an evaluation report. The exercise is referenced as a learning object (LO) available on an interoperable repository [3] supporting extended definitions of learning objects [4]. The business process model relates several abstract services definitions from the e-Framework, including the proposed service.

Examples of the applicability of this service usage model can be drawn from different areas, although the authors are particularly interested in the automatic evaluation of programming exercises. A program evaluation service compiles a program source code, executes it with test data and compares obtained and expected outputs contained in a learning object. Other examples of evaluator services process different types of text files: an electronic circuit evaluator receive a description of a circuit, injects input signals, simulates the circuit and compares output signals; a diagram evaluator receives a description of a diagram (e.g. UML) – a typed graph – and tries to create a graph homomorphism with a solution. In all cases the service receives both a text file attempting to solve an exercise and a reference to an exercise specified as a learning object, containing other files with special roles in the evaluation process, and produces a detailed evaluation report.

The remainder of this paper is organized as follows. Section 2 details the evolution towards the e-learning frameworks and introduces the e-Framework.

Section 3 proposes a new abstract service type for text file evaluation and section 4 builds a service usage model using this abstract type. As a validation of these proposals, section 5 presents the definition of a concrete service based on the text file evaluation genre and a concrete service usage model for evaluation of programming exercises. Finally, a summary of the major contributions of this paper and a prospect of future work are presented.

2 Evolution towards e-learning frameworks

The architectures of e-learning platforms had a considerable evolution in the last two decades. Starting with the early monolithic systems developed for specific learning domain to domain-independent systems featuring reusable tools that can be used virtually in any e-learning course [6]. These last systems follow a component oriented architecture in order to facilitate tool integration. Integrated environments have been successfully used to leverage the advantages of ICTs, but have also been target of criticism. These systems, based on pluggable and interchangeable components, led to oversized systems that are difficult to reconvert to changing roles and new demands such as the integration of heterogeneous services based on semantic information and the automatic adaptation of services to users (both learners and teachers).

These issues triggered a new generation of e-learning platforms based on services that can be integrated in different scenarios. This new approach provides the basis for SOA. In the last few years there have been initiatives [2] to adapt SOA to e-learning [7]. These initiatives, commonly named e-learning frameworks, had the same goal: to provide flexible learning environments for learners worldwide. Usually they are characterized by providing a set of open interfaces to numerous reusable services organized in genres or layers and combined in service usage models. These initiatives use intensively the standards [8, 9, 10] for e-learning content sharing and interoperability developed in the last years by several organizations (e.g. ADL, IMS GLC, IEEE). Based on a previous survey [2], the authors conclude that E-Framework (E-F) [11,12] and Schools Interoperability Framework (SIF) [13] to be the most promising e-learning frameworks since they are the most active projects, both with a large number of implementations worldwide.

In the E-F, the on-line community is the corner stone of the contribution process [14]. The technical model of the E-F structures the contributions as service genres, expressions and usage models. A *service genre* is an abstract definition of a type of service. A *service expression* is the formalization of the implementation approaches of a specialization of a service genre. A *service usage* model describes the requirements and processes within a particular domain relating them to a collection of service genres or expressions.

On SIF it's impossible to make this type of contribution to the abstract framework. However, developers are encouraged to contribute with new agents, such as learning objects repositories.

3 Text File Evaluation Service Genre

In the e-Framework a *service genre* describes generic capabilities of a specific service expressed in terms of their behaviours, without prescribing how to make them operational.

In this section a text file evaluation service genre is proposed to the E-Framework. A service of this genre is responsible for the assessment of a text file with an attempt to solve an exercise described by a LO. It supports three functions:

- `ListCapabilities:` provides the requester with a list of all the capabilities supported by a specific evaluator;
- `EvaluateSubmission:` performs the evaluation of a submission to a given exercise, using some of the available capabilities;
- `GetReport:` accesses a detailed report of a previous evaluation.

In the following sub-subsection the three service internal functions are detailed.

3.1 The ListCapabilities function

The **ListCapabilities function** informs the client systems of the capabilities of a particular evaluator. Capabilities depend strongly on the evaluation domain. For instance, in a computer programming evaluator the capabilities are related with the programming language compiler or interpreter. Each capability has a number of features to describe it and for a programming language they may be the language name (e.g. Java) its version (e.g. 1.5) and vendor (e.g. JDK). On an electronic circuit simulator a capability may be a collection of gates that are allowed on a circuit and features may be the names of individual gates.

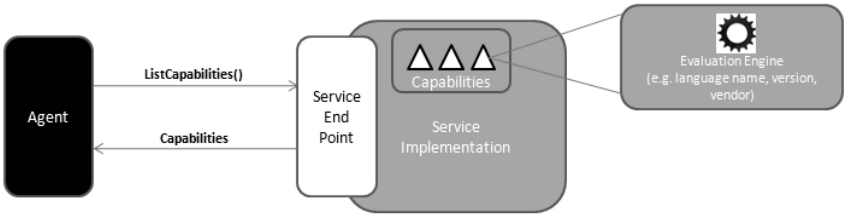

Figure 1 – The ListCapabilities function.

In this function, the request doesn't accept any parameter and the response returns a list of all capabilities of the evaluator. Each capability is described by a list of features, with a name and a value.

3. 2 The EvaluateSubmission function

The **EvaluateSubmission** function allows the request of an evaluation for a specific exercise. The request includes an exercise or a reference to an exercise

represented as a learning object held in a repository and a single attempt to solve a particular exercise. The request may include a specific evaluator capability necessary for a proper evaluation of the attempt. The response returns a ticket for a later report request and may return also a circumstantial report about the respective evaluation of the requester attempt.

A schematic of this function is shown in Figure 2. The service endpoint provides the interfaces for the requests and responses for the evaluation functionality. Internally the service implementation may include several features (indexing, queuing, transforming, flow control, etc.) needed to provide the defined functionality and a connection with a remote data source holding the objects such as a LOR.

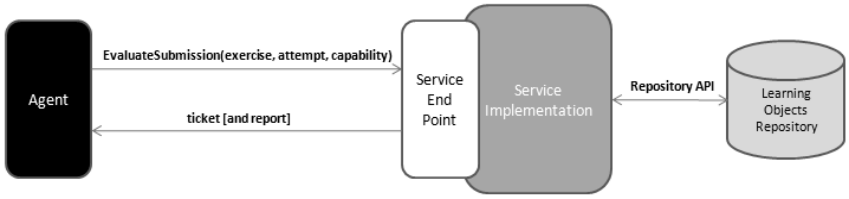

Figure 2 – The EvaluateSubmission function.

The evaluator returns a report on the evaluation, if it is completed within a predefined time frame. The report must contain information about the assessment of the attempt but should not reach to any conclusion. The raw data sent to the client can be used as input for other systems (e.g. classification systems, feedback systems).

In any case the response will include a ticket to recover the report on a later date. Requesting a report using a ticket is supported through another function called GetReport detailed in the next sub-subsection.

3. 3 The GetReport function

The **GetReport function** allows a requester to get a report for a specific evaluation. The report included in this response may be transformed in the client side based on a XML stylesheet. This way the client will be able to filter out parts of the report and to calculate a classification based on its data. The request of this function includes a ticket sent previously by the service in response to an evaluation. The response returns a report about an evaluation.

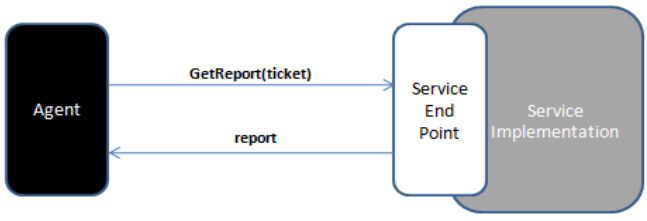

Figure 3 – The GetReport function.

4 Text File Evaluation Service Usage Model

In the E-Framework, a Service Usage Model (SUM) describes the needs, requirements, workflows, management policies and processes within a domain. A SUM is composed of either Service Genres or Service Expressions, but not a mixture. In this section the SUM for the text file evaluation of learning objects is detailed. The E-Framework has 22 distinct elements to describe a SUM, 12 are required elements and the rest is either recommended or optional. For the sake of terseness just a subset of the SUM content based on the templates provided by the E-Framework is detailed. In concrete is described the SUM diagram, the technical functionality, the structure and arrangement of the functions and the data sources and services used.

The **SUM Diagram element**, depicted in Figure 4, defines a visual representation of the SUM for presentation purposes. This type of diagram is suggested by the E-F templates [12]. It organizes business processes in columns. For each business process the summary and name are highlighted in square rectangles in the top and the services genres it includes as ovals. Data sources are represented in the footer of the diagram.

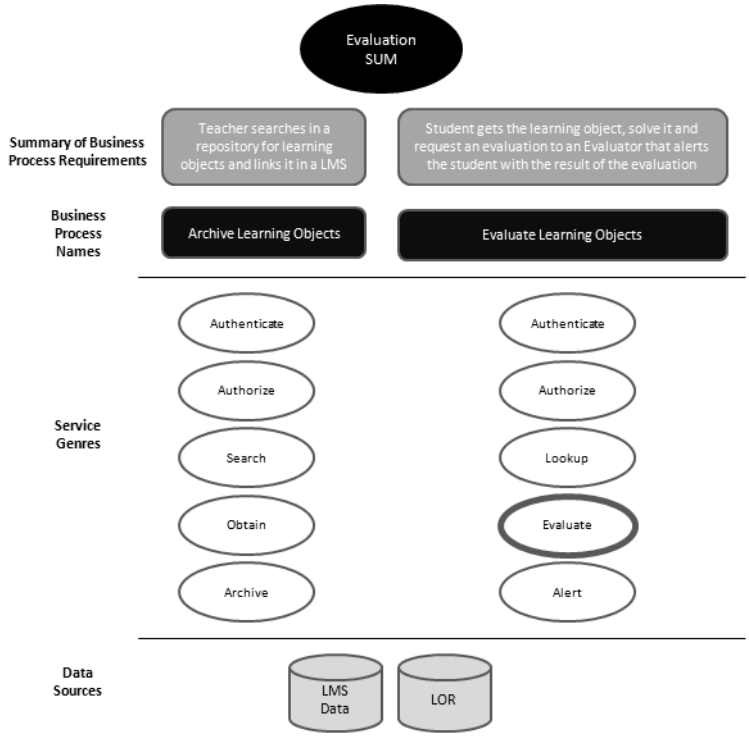

Figure 4 – The SUM diagram.

In the first business process called Archive Learning Objects, the teacher searches in a repository for learning objects. Then, it selects the most appropriate and archives it, for instance, in a LMS for future use.

The Evaluate Learning Objects business process details the attempt of the student to solve a particular learning object and the request for its evaluation. In this business process the Evaluation Service Genre, detailed in the previous subsection, was used. This service includes the EvaluateSubmission function that returns ticket for a later report request and may also return an evaluation report. The report could be sent to both student and teacher or be transformed for a personalized notification about the evaluation of the students' attempt.

The **Functionality element** categorizes the functions supported by the SUM from a system viewpoint. The functions used in this SUM are organized as follows: common functions (Authenticate and Authorize), repository functions (Search, Obtain, Archive, Lookup and Alert) and evaluation functions (Evaluate).

The **Structure & Arrangement element** illustrates how a SUM is used in a particular business process by identifying the services used, data sources and their interactions within the SUM. An apt illustration of the use of this SUM is the pedagogical learning process involving the evaluation of programming exercises, presented in the following section.

5 Validation

The contribution of this paper is twofold and includes the abstract definition of a text evaluation service genre and a SUM involving this genre. To evaluate the practicability of these abstracts definitions we made a concrete definition of a service expression based on the proposed service genre, a programming exercise evaluation service. This service expression was then used to define a concrete service usage model for solving programming exercises in the context of computer programming course.

The definition of the programming exercise evaluation service was also done in the context of the e-Framework. The new service expression specializes the proposed service genre by refining its behaviours and requests, and by specifying implementation approaches such as applicable standards and interface definitions. Details of this specialization process can be found elsewhere [15].

We are currently developing an evaluation engine based on this service expression. The implementation is based on Virtual Machines (VM) to execute the programs on a safe and controlled environment and is divided into five components, two controlling the evaluation service and other three supporting the execution of the programs on the VM. The five independent components give the evaluation engine a higher scalability. The use of VM allows us to manage a high number of capabilities such as languages and programming environments from different operating systems, including obsolete versions.

A text evaluation service with the features outline in the previous paragraphs was designed for a SUM involving the evaluation of programming exercises. At the heart of this SUM resides an evaluation engine – a service of the text file evaluation genre – supplying its services to several e-learning systems such as LMS, LOR or

experimentation environments. An example of an experimentation environment would be an IDE such as Eclipse, with plug-ins to interoperate with other services, where students would solve their programming exercises. Figure 5 shows a concrete business process model based on the proposed SUM.

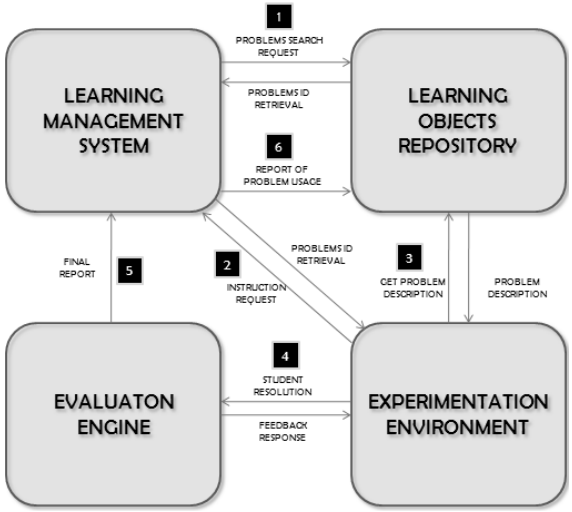

Figure 5 – Service integration in a pedagogical e-Learning process.

The teacher 1) searches for relevant exercises in the repository. Then the learner 2) gets the exercises from the LMS. The Experimentation Environment 3) recovers the exercises descriptions from the repository and shows it to the learner. After coding the program the learner send an attempt 4) to the Evaluation Engine. The learner may submit repeatedly, integrating the feedback received from the Evaluation Engine. In the end, the Evaluation Engine 5) sends a grade to the LMS that records it and reports 6) the LO usage data back to the repository. This last task will provide data for future adaptability services that will adjust the presentation order in accordance with the effective difficulty of programming exercises (not the difficulty stated on the LO) and the needs of a particular student.

6 Conclusion and future work

In the research presented in this paper the authors modelled text file evaluation services and detailed a contribution to the E-Framework, consisting of a new Service Genre and a new Service Usage Model. In the Service Genre the authors made an abstract description of the behaviours expected from a text file evaluation service. In the Service Usage Model the relationships between services through business processes and the usage scenario based on a particular domain - the automatic evaluation of programming exercises – were presented.

To validate the proposed model the authors presented a concrete service expression consisting of a programming exercise evaluation service, and a concrete usage model based on this service expression.

In the continuation of this line of work these contributions to the e-Framework will be used to design actual implementations of text file evaluation services, not only of programming exercises and also for other domains, such as UML diagrams.

The proposed service usage model will be used to implement an e-learning process centred in the automatic evaluation of programming exercises. This e-Learning process will integrate the services exposed by several systems, including an LMS, an IDE and a LOR. These systems are being extended to enable them to expose some of their functions as services and/or to consume services of other systems. With this approach we expect to achieve an integrated e-learning environment for computer programming with the best of bread tools in each category.

References

1. Girardi, R.: Framework para coordenação e mediação de Web Services modelados como Learning Objects para ambientes de aprendizado na Web (2004)
2. Leal, J.P. and Queirós, R.: eLearning Frameworks: a survey. Proceedings of International Technology, Education and Development Conference 2010, Valencia, Spain (2010)
3. Leal, J.P., Queirós, R.: CrimsonHex: a Service Oriented Repository of Specialized Learning Objects. In: ICEIS: 11th International Conference on Enterprise Information Systems, Milan (2009)
4. Leal, J.P., Queirós, R.: Defining Programming Problems as Learning Objects. In: World Academy of Science, Engineering and Technology, vol. 58, pp.1033-1040 (2009)
5. Leal, J.P and Silva, F.: Mooshak: a Web-based multi-site programming contest system Software, Practice & Experience, Volume 33 , Issue 6, Pages: 567 - 581, ISSN:0038-0644 (2003)
6. Dagger, D., O'Connor, A., Lawless, S., Walsh, E., Wade, V.: Service Oriented eLearning Platforms: From Monolithic Systems to Flexible Services (2007)
7. Wilson. S, Blinco, K. Rehak, D: Service-Oriented Frameworks: Modelling the infrastructure for the next generation of e-learning systems (2004)
8. IMS CC Specification, Version 1.0 Final Specification: http://www.imsglobal.org/cc/index.html
9. Bohl, O., Scheuhase, J., Sengler, R. and Winand, U.: The shareable content object reference model (SCORM)-a critical review, Proceedings of the International Conference on Computers in Education, pages 950-951(2002)
10. Hatala, M., Richards, G., Eap, T., Willms, J.: The interoperability of learning object repositories and services: standards, implementations and lessons learned. Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, ACM, New York, NY, USA (2004)
11. S. Wilson, K. Blinco, D. Rehak: An e-Learning Framework. Paper prepared on behalf of DEST (Australia), JISC-CETIS (UK), and Industry Canada (2004)
12. e-Framework Technical Walk-through: http://www.e-framework.org/Portals/9/docs/e-Framework%20technical%20walk-through%20v1.1.pdf
13. Collins, L.: Schools Interoperability Framework White Paper. Schools Interoperability Framework Association, Washington DC (2005)

14. Stock, K., Butchart, B., Higgins, C. and Chen, Y.: From Here to Eternity: An Experiment Applying the e-Framework Infrastructure for Education and Research and the SUMO Ontology to Standards-based Geospatial Web Services, International Journal of Spatial Data Infrastructures Research, Vol. 5, 1-57, United Kingdom (2010)
15. Leal, J.P.; Queirós, R.; Ferreira, D: Specifying a programming exercises evaluation service on the e-Framework, *accepted for publication in the proceeding of ICWL 2010, Shanghai*.