Modelling a network of heterogeneous eLearning systems

Ricardo Queirós and José Paulo Leal
CRACS/INESC-Porto & DCC/FCUP
University of Porto
Portugal

June 30, 2011
Table of contents

1. Introduction
   1.1 Context
   1.2 Motivation

2. eLearning Frameworks

3. The E- Framework

4. Evaluation service description
   4.1 Service Genre
   4.2 Service Expression
   4.3 Service Usage Model

5. Validation

6. Conclusion
1.1 Introduction - context

- networks of eLearning systems providing services related to **automatic evaluation of programming exercises**
- involves the work in unison of several system types

![Diagram showing components of an eLearning system]

- problems:
  - disparate and heterogeneous systems types
  - standard fragmentation in eLearning realm
1.2 Introduction - motivation

- these types of systems have a completely different nature
  - Some expose their functions as web services (e.g. LOR or EE)
  - Some have their own web interfaces (e.g. LMS)
  - Some were not even designed to interact in the eLearning realm and must be extended for that purpose (e.g. ERE)

- modelling a network with such heterogeneity is challenging

- ultimate goal:
  - model a network of heterogeneous eLearning systems
  - automatic evaluation of programming exercises as a case study

- twofold contribution
  - survey of the efforts made to adapt SOA in eLearning
  - design of a evaluation service
2. eLearning Frameworks (1)

- a specialized software framework
- several initiatives to adapt SOA to eLearning
- organized in two groups:

  - **abstract frameworks:**
    - specifications, recommendations and best practices for the development of eLearning systems
    - e.g. IMS AF, OKI and IEEE LTSA

  - **concrete frameworks:**
    - service designs and/or components that can be integrated in actual implementations of artifacts
    - e.g. E-Framework, SIF and OUSS
2. eLearning Frameworks (2)

- trend is the appearance of the concrete frameworks

- none of the concrete frameworks actually implement artifacts
2. eLearning Frameworks (3)

- compare 5 of these frameworks: impact and maturity, architectural models, adopted standards and user groups

<table>
<thead>
<tr>
<th>Facets</th>
<th>Features</th>
<th>LTSA</th>
<th>OKI</th>
<th>IAF</th>
<th>SIF</th>
<th>E-F</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1st version date</td>
<td>1996</td>
<td>2003</td>
<td>2003</td>
<td>2003</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Last vers. Date</td>
<td>2001</td>
<td>2006</td>
<td>2003</td>
<td>2009</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Cited projects</td>
<td>-</td>
<td>3</td>
<td>-</td>
<td>37</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Contributions</td>
<td>inactive</td>
<td>yes</td>
<td>yes</td>
<td>Yes</td>
<td>yes</td>
</tr>
<tr>
<td>Architectural Models</td>
<td>Main model</td>
<td>layered</td>
<td>layered</td>
<td>layered</td>
<td>flat</td>
<td>layered</td>
</tr>
<tr>
<td></td>
<td>SOA</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>Yes</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>Content format</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>SCORM</td>
<td>SCORM</td>
</tr>
<tr>
<td></td>
<td>Metadata</td>
<td>LOM</td>
<td>LOM</td>
<td>LOM</td>
<td>LOM</td>
<td>IMS CP</td>
</tr>
<tr>
<td></td>
<td>Web Service</td>
<td>SOAP</td>
<td>SOAP</td>
<td>SOAP,</td>
<td>SOAP,</td>
<td>SOAP,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>REST</td>
<td>REST</td>
<td>REST</td>
</tr>
<tr>
<td></td>
<td>Lang. bindings</td>
<td>-</td>
<td>JAVA, PHP, C#</td>
<td>JAVA</td>
<td>JAVA</td>
<td>JAVA</td>
</tr>
<tr>
<td>User Groups</td>
<td>Framework users</td>
<td>ESV</td>
<td>ESV</td>
<td>IMS</td>
<td>ESV</td>
<td>ESV</td>
</tr>
<tr>
<td></td>
<td>End users</td>
<td>HE</td>
<td>HE</td>
<td>HE</td>
<td>K-12</td>
<td>HE</td>
</tr>
</tbody>
</table>

- E-Framework and SIF are the most promising frameworks
  - they are the most active projects
  - both with a large number of implementations worldwide
3. E-Framework

- the most prominent e-learning framework currently in use
- aims to facilitate interoperability in HEI and research
- has a knowledge base to support its technical model
  - via a service-oriented approach
  - composed by 3 technical components

<table>
<thead>
<tr>
<th>Components</th>
<th>Description</th>
<th>User role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service</td>
<td>A collection of related behaviours that describe an abstract capability.</td>
<td>No technical expert (e.g. IT Manager)</td>
</tr>
<tr>
<td>Service Expression</td>
<td>A specific way to realise a service genre with particular interfaces and standards.</td>
<td>Technical expert (e.g. Developer)</td>
</tr>
<tr>
<td>Service Usage Model</td>
<td>The relationships among technical components (services) used for software applications.</td>
<td>Domain expert (e.g. Business Analyst)</td>
</tr>
</tbody>
</table>

- other components such as specifications and standards (e.g. IMS Metadata, LOM) are used by service expressions but are not defined by the E-F
4. Evaluation service description

- definition of an evaluation service
- to be used by an Evaluator
  - to mark and grade exercises in
    - computer programming courses
    - programming contests
  - to participate in processes integrating different system types
    - Programming Contest Management Systems
    - Learning Management Systems
    - Exercises Resolution Environments
    - Learning Object Repositories
- contribution of this service to the E-Framework
  - Service Genre
  - Service Expression
  - Service Usage Model
4.1 Text File Evaluation Service Genre (1)

- responsible for the assessment of a text file with an attempt to solve an exercise described by a LO

- 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
4.1 Text File Evaluation Service Genre (2)

- **ListCapabilities function**
  - informs the client systems of the capabilities of a particular evaluator
  - capabilities depend strongly on the evaluation domain (e.g. programming language compiler or interpreter)
  - each capability has a number of features to describe it (e.g. language name and version - Java 1.5)
4.1 Text File Evaluation Service Genre (3)

- **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
    - a single attempt to solve a particular exercise
    - (optionally) a specific evaluator capability necessary for a proper evaluation of the attempt.
  - the response includes
    - a circumstantial report about the respective evaluation of the requester attempt
    - a ticket for a later report request
4.1 Text File Evaluation Service Genre (4)

- **GetReport function**
  - allows a requester to get a report for a specific evaluation
  - 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
  - the report does not compute a grade, points or classification (XSLT post-transformation)
4.2 The Evaluate - Programming Exercise SE (1)

- Evaluate - Programming Exercise
  - specializes the Evaluate service genre
  - evaluation of an attempt to solve an exercise defined as a learning object
  - focus on the **automatic evaluation of programming exercises**

- Service expression characterized by:
  - Use & Interactions
  - Applicable Standards
  - Interface Definition
  - Usage Scenarios
4.2 The Evaluate - Programming Exercise SE (2)

- Use & Interactions
  - how the service functions are combined to produce a workflow

```text
Learning:LMS

Repository:LOR

Evaluator:EE

search(XQuery query) : XML

List of LO's URLs and meta-data

retrieve(URL loid) : LO

LO

ListCapabilities() : ERL

capabilities

getCapabilities(): ERL

evaluateSubmission(URL loid, String attempt, ID capability) : ERL

GetAsset(URL loid, String asset) : Asset

Asset file

ticket [and report]

GetReport(String ticket) : ERL

ticket and report
```
4.2 The Evaluate - Programming Exercise SE (3)

- Applicable Standards
  - enumerates all the domain and technical standards
  - e-learning content standards
    - IMS Content Packaging (IMS CP)
    - IEEE Learning Object Metadata (IEEE LOM)
    - EduJudge Metadata (extension IMS CP)
  - e-learning communication standards
    - IMS Digital Repository Interoperability (IMS DRI)
4.2 The Evaluate - Programming Exercise SE (4)

- Interface Definition (1)
  - formalizes the interfaces of the service expression, namely the syntax of requests and responses of its functions
  - exposes its functions as SOAP and REST web services

<table>
<thead>
<tr>
<th>Function</th>
<th>Web Service</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>ListCapabilities</td>
<td>SOAP</td>
<td>ERL ListCapabilities()</td>
</tr>
<tr>
<td></td>
<td>REST</td>
<td>GET /evaluate/ &gt; ERL</td>
</tr>
<tr>
<td>EvaluateSubmission</td>
<td>SOAP</td>
<td>ERL Evaluate (Problem, Attempt ,Capability)</td>
</tr>
<tr>
<td></td>
<td>REST</td>
<td>POST /evaluate/$CID?id=LOID &lt; PROGRAM &gt; ERL</td>
</tr>
<tr>
<td>GetReport</td>
<td>SOAP</td>
<td>ERL GetReport(Ticket)</td>
</tr>
<tr>
<td></td>
<td>REST</td>
<td>GET $Ticket &gt; ERL</td>
</tr>
</tbody>
</table>

- All these functions respond with an XML document complying with the Evaluation Response Language (ERL)
4.2 The Evaluate - Programming Exercise SE (4)

- Interface Definition (2)
  - ERL is formalised in XML Schema
  - two main elements: *request* and *reply*

- the former echoes the request function and its parameters as received by the evaluation service
- the later contains the output to that request
4.2 The Evaluate - Programming Exercise SE (5)

- Usage Scenarios
  - characterizes the types of workflows in which the SE is used
  - these workflow types can be classified as curricular and competitive learning

- **curricular learning**
  - *practical classes*: instant feedback provided to students with the failed test cases and providing hints to solve them
  - *programming assignments* combining automatic and human evaluation both feedback and grading are relevant
  - *examinations* different grading policies can be implemented by the client based on the tests cases successfully completed

- **competitive learning**
  - relies on the competitiveness of students to increase their programming skills
  - common goal of several programming contests (e.g. IOI, ICPC, IEEE Extreme)
  - Each programming contest type has its own set of rules (own policy for grading and ranking submissions)
4.3 Text File Evaluation Service Usage Model

- describes the needs, requirements, workflows, management policies and processes within a domain
5. Validation

- to evaluate the usefulness of these service definitions we made a concrete definition of a SE
6. Conclusion

- main contributions
  - survey on eLearning frameworks
  - contribution of an evaluation service for programming exercises to the E-Framework
    - Service Genre
    - Service Expression
    - Service Usage Model
- to evaluate the usefulness of these service definitions we made a concrete definition of an Evaluation SE
- currently...
  - ...finishing the implementation of the SE in Mooshak
  - ...including ERE in the network (IMS bLTI/CC)