Zenaminer: driving the SCORM standard towards the Web of Data

E. Mudu, L. Schiatti, G. Rizzo, A. Servetti

Heraklion, May 29
Linked Learning 2011
Agenda

• Idea
• SCORM specifications and structure
• LMS: state of the art
• Proposal
• Conclusions
• Perspectives
Idea

SCORM player able to publish SCOs into the Web of Data

Allow the creation of customized Web interface for SCORMs

Allow the migration of SCOs into the Web of Data where a “consumer” can infer, connect and aggregate data from different repositories
Why SCORM?

It's one of the most used standard packaging of e-learning contents

Potentially many SCORM objects are spread within the Web

Possibility to introduce and link them in the Linked Data cloud
SCORM structure

SCORM object defines how a course is organized.

It provides a tree structure.

SCO is a minimal reusable and shareable part.
SCORM example
Learning Management System: state of the art
LMS: limitations

* SCO designer cannot control all the graphical interface used but only the space dedicated for the SCO

* Difficult to integrate more information (text, video, ... )

* Contents and presentation are mixed, hard to adapt the interface for different devices (smart phone, tablet, notebook)
Goal

Publish educational data into the Web of Data

Link automatically comments to Web resources

RESTful application to implement these features
RESTful architecture
Publish educational data

- SCORM objects are sent to the REST controller
- REST controller elaborates each object and exports all raw data (text information)
- Then it publishes raw data, allowing users to build their preferred layout
1. **POST /scorms**
2. parse SCORM object and select html files
3. for each html, check if it is valid or not
4. for each html, extract slides and store the tree
Demos of customized views
Link comments to Web resources

To allow collaborative learning, Zenaminer considers all text comments edited by a user as a learning enrichment and shares them with all the users in a wiki style
Looking towards Linked Data

Each comment is annotated by means of a DBpedia Spotlight

DBpedia Spotlight
✓ extracts one named entity for each comment
✓ maps each comment to a DBpedia resource
POST /comment

1. POST /comment, enrichment='…'
2. invoke spotlight client
3. extract from enrichment list of entities
4. store list of entities
Conclusions

Publication of SCORM contents in the Web of Data as raw data separating the content from the layout. Any SCO designer is free to choose or design his preferred view

Automatic annotation for the integration of contents with hypertext link navigation in order to satisfy more effectively the information needs of the learner
Perspectives

Vocabulary for SCORM and URI schema
Fragments

Evaluation of those systems

Attach provenance/licence to SCOs
http://zenaminer.sourceforge.net/
http://sourceforge.net/projects/zenaminer/

http://www.slideshare.net/giusepperizzo/