



# Semantic Retrieval of Learning Objects with Schema Matching

Beniamino Di Martino

Seconda Università di Napoli  
beniamino.dimartino@unina.it

Keywords: Learning Objects, Ontology Web Language, Schema Matching, SCORM

## Abstract

This article covers semantic methods and techniques based on semantic web technology (Berners-Lee *et al.*, 2001) for content management in e-Learning platforms. The technique, and its prototypal implementation, involves the semantic search of learning objects (defined by David Wiley as “any digital resource that can be reused to support learning” (Wiley, 2002)) contained in SCORM (Sharable Content Objects Reference Model) “packages” (ADL) – a reference standard that ensures interoperability between different learning management systems (LMS). This technique integrates semantic techniques, combined with pattern matching techniques, to search for relevant learning objects for a given domain ontology used for queries, such as an ontology that describes the reference domain of a course of study. This correspondence is obtained through a matching operation, which returns a value of similarity between the ontology query (in OWL (Smith *et al.*, 2004)) and learning objects such as those contained within repositories of online courses.

## 1 Introduction

This article describes the construction and functionality of a prototype tool, preliminarily called ELTrieve, developed by the Computer Research Group of the Second University of Naples. The tool is used for the semantic search of learning objects, using a schema matching operation (for an overview of schema matching techniques see (Rahm & Bernstein, 2001; Shvaiko & Euzenat, 2005; Giunchiglia & Shvaiko, 2003)) for schema deriving, on the one hand, from representations of the ontology of reference (e.g. in OWL format), and on the other, from SCORM representations of learning objects (manifesto) in the XML format and contained in the `imsmanifest.xml` file. These patterns are parsed (through DOM Parsers) and represented by rooted directed graphs.

Natural language processing algorithms have also been used and integrated for the analysis of the text of individual documents in the e-learning course and described in the manifesto. Search capabilities, based on Google, have also been added to the tool for the retrieval of online educational materials.

The matching process involves mapping to indicate which elements of the input schema correspond (semantically and structurally). The result of the match is a set of mapping elements that specify the elements of the two schema that “match” within a similarity value of between 0 (no similarity) and 1 (high similarity), indicating a plausible correspondence. The schema matching technique developed combines two approaches: a structural approach, based on algorithms: Near Matcher, VF Isomorphism and Subgraph Matcher (the latter two based on the VF algorithm (Cordella *et al.*, 1999)); and a linguistic approach based on algorithms: String Distance (based on the Levenshtein Distance algorithm) and Synonym Matcher (based on the WordNet thesaurus). A detailed description of the selected algorithms and the technique is available in (Di Martino, 2009) where this technique is applied to the semantic search of Web Services.

## 2 The ELTrieve tool

In this section we shall describe the creation and functionality of the prototypical tool, preliminarily called ELTrieve. Below is a flow chart that illustrates the tool’s information processing phases.

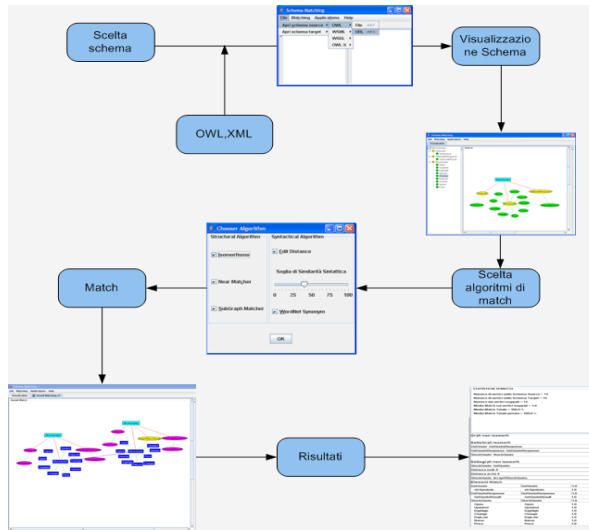


Fig. 1: Flowchart of processing phases

Below is a description of the tool’s interface panels for all processing phases. The preliminary stage involves loading the schema to be analyzed.

Figure 2 shows the graphs of the loaded schema and their navigation trees.

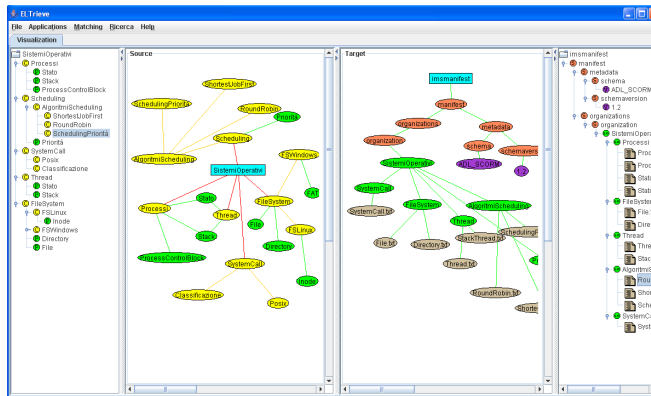


Fig. 2: Graphical User Interface - Schema View

Before carrying out the matching process of the two schemes, a selection is made of the appropriate structural and syntactical algorithms to be applied. We can also set, using a slider, the minimum degree of syntactic similarity lag

between the information contained within nodes of the two schemes under consideration.

Moreover, for a more detailed analysis of the contents of the didactic material in a given course, the user can select individual document normalization, removing noise words (punctuation, numeric characters, empty words) by stemming individual words, and finally indexing them and extracting the words needed most often.

Then the most frequently used words in each document are compared individually with all the ontology concepts to obtain the degree of similarity of the contents of individual documents with the ontology of reference. The comparison is made only if the “weight” of the word is greater than a threshold set by the user, using an appropriate slider.

Having selected the algorithms, which by default are all still active, the actual matching process can then be started. Figure 3 shows the graphical interface with the nodes of the two matching schema displayed in purple. A clearer description of each individual node match can be obtained by a right click on the node; the matching nodes appear in blue, connected by a link also in blue, as shown in Figure 6. Adjacent nodes also appear in blue if they match the adjacent nodes of the other node, which indicates that a structural match exists between the subgraphs of the two schemas. On the other hand, a left click on the document nodes will bring up the results of document analysis. All statistics and results of matches obtained can still be seen on the right side of the panel. In particular, the match statistics, Figure 3, include the following information: number of vertices of the source schema; number of vertices of target schema; number of vertices of the source schema mapped on the vertices of the target schema; syntactic match average of the matching vertices; total match average is the percentage of vertices of the source schema mapped on the vertices of the target schema; weighted match average is a percentage obtained by multiplying the total match average by the syntax match average. The structural match gives the statistics for the structural match between the two graphs: isomorphic or non isomorphic graphs; isomorphic subgraphs; non isomorphic subgraphs and their distance in terms of links and nodes; near match result. Finally the match element gives the statistics concerning linguistic matches in two nodes. For example, a value equal to 1.0 means that there is a perfect match between the two nodes.

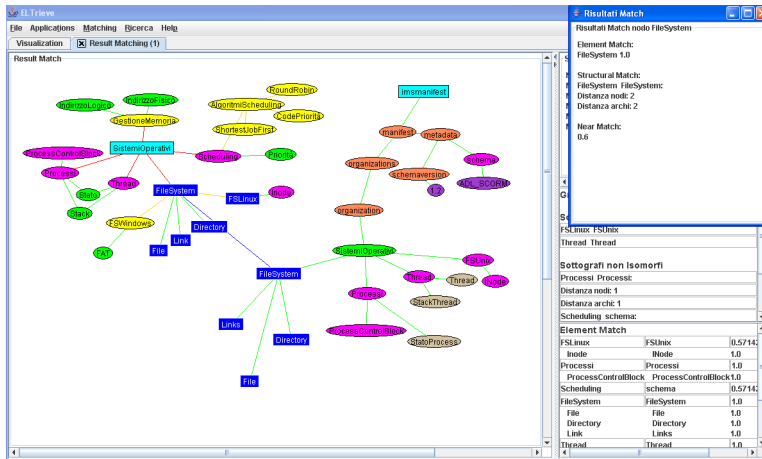


Fig. 3: Graphical User Interface - Match Results

Finally a web search feature was included in the tool by interfacing with the Google search engine. This search feature is useful for finding SCORM packages on the web that meet users' requests. These packages are described in the XML manifesto (imsmanifest.xml), which can then be matched with the OWL ontology that describes a given course, in order to be able to extrapolate lessons or learning objects that match the ontology concepts.

### 3 A Case Study

We now describe an application of the tool's features in a case study - an OWL ontology describing the basic concepts of a course on operating systems, and an XML document representing a SCORM file of an online course on Operating Systems. The ontology examined is presented below (using the tool for a building Protégé ontology), showing the hierarchy of classes and properties associated with them.

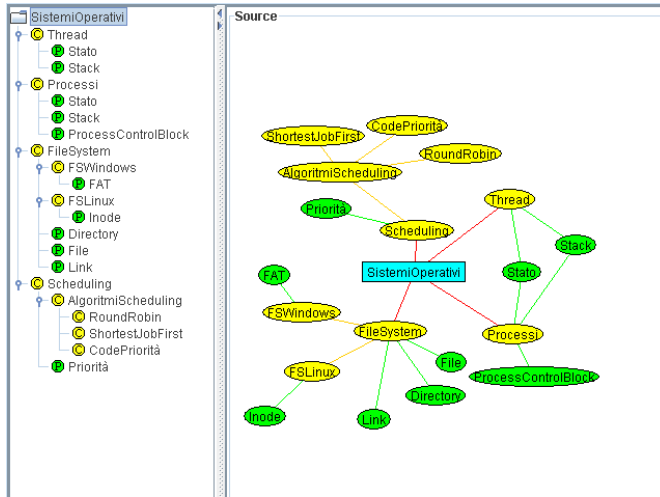


Fig. 4: Classes, properties and hierarchies

This ontology can effectively represent the description and structuring of a course of study, in this case an Operating Systems course with the following possible lessons: Processes, Threads, File Systems and Scheduling. Each lesson is divided into subtopics.

To define the XML documents in SCORM format (called imsmanifest) Learning Management System Dokeos 1.6.2 was used. The XML document, as already mentioned, is the result of SCORM 1.2 packaging together with the actual contents of the course (individual files) and is divided into four parts, of which the metadata is the descriptive part of the course that defines elements relating to the standard and elements that can be used for cataloguing. Organization defines the sequence of learning objects (SCO) within the course; there may be more than one organization for each package.

In this example, online Operating Systems course 1 consists of five main lessons: Processes, Scheduler, Algorithms, Scheduling, Windows FS and Unix FS. Each lesson is divided into a series of less complex topics, for example Windows FS deals with FAT, Unix FS deals with inodes. There is also a lesson that deals with scheduler policies, a lesson that deals in detail with scheduling algorithms and finally a lesson that deals with processes, process control blocks and process states.

Operating Systems course 2 includes four main lessons: Processes, Threads, File Systems and Unix File Systems. The lesson on processes is structured like Operating Systems 1, but the lesson on file systems is on files, directories and links in general while inodes are dealt with in the Unix file system lesson.

The main lessons in both courses represent the single learning objects while the topics of each learning object represent, in this case, text documents that deal with the same topics.

The results produced by the tool for the two examples will now be analyzed.

TABLE 1  
Match statistics

Match Statistics - Operating Systems Course 2	Match Statistics - Operating Systems Course 1
Nodes number on Source Schema = 19	Nodes number on Source Schema = 19
Nodes number on Target Schema = 21	Nodes number on Target Schema = 22
Matched nodes number = 10	Matched nodes number = 11
Syntax match average on matched nodes = 0.9142857142857143	Syntax match average on matched nodes = 0.9206349206349206
Total syntax match average = 52.63157894736842 %	Total syntax match average = 57.89473684210527 %
Total weighted match average = 47.1203007518797 %	Total weighted match average = 53.29991645781119 %

The first two statistics refer to the number of nodes of the two schema, while the third, certainly more interesting, shows that only 10 nodes in the ontology matched to some degree the XML document nodes that describe Operating Systems course 2, while in the second case they matched 11 nodes.

The fourth statistic gives the syntax match average, which tells us how far the elements of the two schemes are similar semantically. In the comparisons, the values are equal to: 0.91 for Operating Systems 2 and 0.92 for Operating System 1. This means that the concepts in the two schemas are very similar.

The most interesting and useful match statistics in this case is the total weighted match average, which in one result gives information on the number of OWL ontology concepts that have actually matched the XML document elements and the syntax average. In this case we have quite low values in both cases: 53% for Operating Systems 1 and 48% for Operating Systems 2.

Structural matching produced the following results: in the first case we have three isomorphic subgraphs (Windows FS, Linux FS and scheduling algorithms), while in the second there is only one (thread). The distance between the two subgraph processes is also shown, which is equal to one for both nodes and links, as the stack element is missing in one of the two. In Operating Systems course 1 there is also a scheduler at a distance of 1 from the scheduling ontology in OWL. Instead, in the second example the file system subgraphs are at a distance of 2. Figure 6 shows matches for the scheduling algorithms of the OWL schema with the same element of the Operating Systems 1 course.

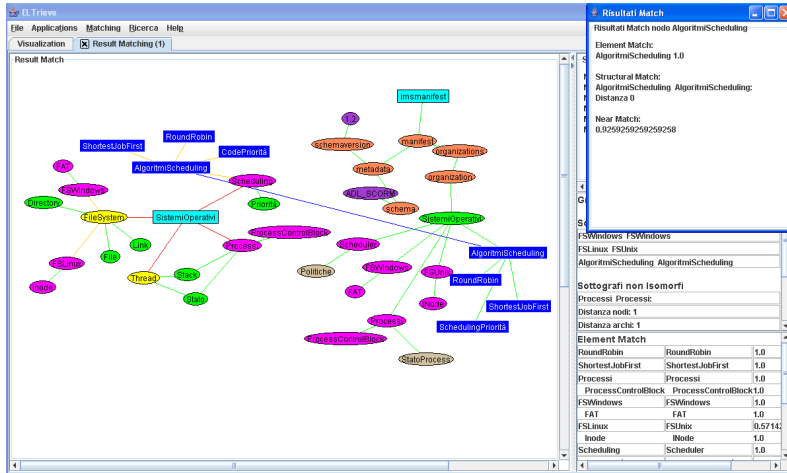


Fig. 5: Node match details – Scheduling Algorithms

The Match Element indicates that the element has been mapped with the scheduling algorithms of the XML schema; a semantic similarity was found between the two nodes equal to 1.0 (the maximum allowable).

The Subgraph Match indicates that their two subgraphs are identical in that the distance is 0 for both nodes and links. To show this, the adjacent nodes are also in blue since they matched the adjacent nodes of the other node. Finally the NearMatcher result is shown, which in this case is equal to 0.922, due to the fact that the priority nodes of the ontology and priority scheduling of Operating Systems 1 course have a similarity match of 0.55.

These results confirm that the lesson on scheduling algorithms in Operating Systems 1 course responds to the request of the same lesson in the ontology of reference, both structurally and semantically. To confirm this, furthermore, we can also analyze other individual documents of the lesson as shown below.



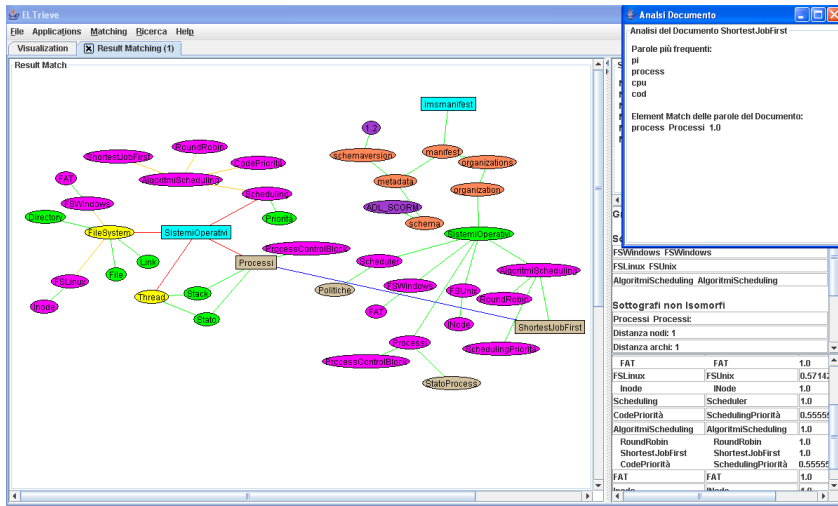


Figura 6: Single document analysis – ShortestJobFirst

A left click on the document ShortestJobFirst opens a pop-up containing information on the single document analysis. In the example it is clear that the most frequent words in the document ShortestJobFirst are: CPU, process, cod. Of these, only process has a similarity match equal to 1.0 with the concept of process in the ontology. Grey-coloured nodes indicate that they correspond to the ontology concepts that match the most frequent words of the document. In the document RoundRobin, too, the most frequent word is process, which has a similarity match of 1.0 with the concept of process in the ontology. The document SchedulingPriorità, however, unlike the previous has no salient words, in which case the document content may be of little relevance (these results, however, are not shown for reasons of space).

From the results of these simple examples we may conclude that we have identified semantically three learning objects from Operating Systems course 1: Scheduling Algorithms, Unix FS and Windows FS. While from the Operating Systems 2 course we can identify the learning objects FileSystem and, once again, Unix FS.

## Acknowledgments

The work described was supported by the MUR-FAR “LC3 – Laboratorio Pubblico-Privato sulla Comunicazione delle CONoscenza Culturali” project and the Campania Region - Law 5 “MAGDA.” I would like to thank Carlo Baia

and Angelo Martone for prototype implementation.

## BIBLIOGRAPHY

---

- Advanced Distributed Learning Initiative: <http://www.adlnet.org>.
- Berners-Lee T., Hendler J, Lassila O. (2001), *The Semantic Web*. Scientific American, May 2001.
- Cordella L.P., Foggia P., Sansone C., Vento M. (1999), *Performance evaluation of the VF Graph Matching Algorithm*, Proc. of the 10th ICIAP, IEEE Computer Society Press, pp. 1172-1177, 1999.
- Di Martino B. (2009), *Semantic Web Services Discovery based on Structural Ontology Matching*, International Journal of Web and Grid Services (IJWGS), Inderscience Press, Vol. 5, n. 1, pp. 46-65, 2009.
- Gilleland M., *Merriam Park Software*, Levenshtain Distance Algorithm, <http://www.mriampark.com/ld.htm>
- Giunchiglia F., Shvaiko P. (2003), *Semantic matching*, The Knowledge Engineering Review Journal, 18(3):265-280, 2003.
- Rahm E., Bernstein P. (2001), *A survey of approaches to automatic schema matching*, The International Journal on Very Large Data Bases (VLDB), (10(4)):334–350, 2001. Smith M. K., Welty C., McGuinness D. L. (2004), *OWL Web Ontology Guide*, <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>.
- Shvaiko P., Euzenat J. (2005), *A survey of schema-based matching approaches*, Journal on Data Semantics (JoDS), IV, 2005.
- Wiley D. (2002), *Connecting learning objects to instructional design theory: a definition, a metaphor, and a taxonomy.doc*, in: Wiley D.(Ed.), The Instructional Use of Learning Objects, 2002.