New Review of Hypermedia and Multimedia
Vol. 00, No. 00, Month 2009, 1–19

A Recommendation Module to help Teachers Build Courses through the Moodle Learning Management System

Carla Limongelli,[a*] Matteo Lombardi,[b] Alessandro Marani,[b] Filippo Sciarrone[a] and Marco Temperini[c]

[a]*Engineering Department, Roma Tre University, Rome, Italy* ; [b] *School of ICT, Nathan Campus, Griffith University, Australia*; [c]*Department of Computer, Control and Management Engineering, Sapienza University, Rome, Italy*

(*v1.0 released July 2014*)

In traditional e-learning, teachers design sets of Learning Objects and organize their sequencing; the material implementing the Learning Objects could be either built anew or adopted from elsewhere (e.g. from Standard-compliant Repositories) and reused. This task is applicable also when the teacher works in a system for personalized e-learning. In this case the burden actually increases: for instance, the Learning Objects may need adaptation to the system, through additional metadata. This paper presents a module that gives some support to the operations of retrieving, analyzing and importing Learning Objects from a set of standard Learning Objects Repositories, acting as a recommending system. In particular, it is designed to support the teacher in the phases of: i) retrieval of Learning Objects, through a keyword-based search mechanism applied to the selected repositories; ii) analysis of the returned Learning Objects, whose information is enriched by a concept of relevance metric, based on both the results of the searching operation and the data related to the previous use of the Learning Objects in the courses managed by the Learning Management System; iii) Learning Object importation into the course under construction.

Keywords: e-Learning Systems, recommending Systems

[*]Corresponding author. Email: limongel@dia.uniroma3.it

ISSN: 1361-4568 print/ISSN 1740-7842 online
© 2009 Taylor & Francis
DOI: 10.1080/136146YYxxxxxxxx
http://www.informaworld.com

1. Introduction

Social networks and Semantic Web have empowered significantly, in recent years, the exploitation of information administered via the web: while the former allowed for a widespread increase in information availability (a few examples are Facebook, MySpace and Twitter), the latter has greatly influenced the possibility to make machine-processable the web contents.

Presently, several software frameworks are available for the implementation of collaborative platforms, aimed at supporting social interaction and exchange in various fields, from work to leisure. A notable example is the platform Elgg[1], a social networking engine enabling to create full-featured social networks and related applications, via an increasing variety of building blocks. In the field of distance education, many approaches to e-learning encompass the use of social networking systems and knowledge sharing (Limongelli *et al.* 2010, Micarelli *et al.* 2009, Deed and Edwards 2010, Benson *et al.* 2012, Nanni and Temperini 2012), while Learning Management Systems (LMS), such as Moodle[2], dotLRN[3], Claroline[4], or ATutor[5], support quite extensively collaborative learning through Web2.0 tools. For instance, ATutor provides, through Google, the Open Social engine, for student collaboration on learning activities.

Summing up, while the demand for distance learning is surging ahead, and e-learning platforms empower learners and teachers with convenient and augmented instructional opportunities (Limongelli *et al.* 2008, 2011a, Gasparetti *et al.* 2009), a social-collaborative factor in e-learning is also to be taken into account.

In this paper we mainly consider teacher-related aspects, in particular the support to Learning Objects (LOs) retrieval and reuse from standard repositories.

The long-standing rise of standards for e-learning has originated a wide availability of Learning Objects Repositories (LORs) on the Internet (e.g.: MERLOT, CNX and WISC-ONLINE), with a growing portfolio of structured learning materials in the form of LOs.

Building a new e-learning course is a critical and hard task for teachers, implying tasks such as setting the learning goals, organizing a concept map, acquiring and possibly adapting existing LOs, and ensuring that the course be delivered through the LMS. Consequently, a standing problem is how to support the teacher's ability to retrieve and select learning materials appropriate for their educational purposes, while making more efficient and speeding up the overall process of course construction (see Di Martino 2009, Neri and Colombetti 2009, for an overview on this topic).

This paper presents a system that can help teachers in the endeavor of finding learning material suitable for their course, by improving the process of retrieval and reuse of LOs, performed on different LORs at the same time. Moreover, during the process, the teacher is allowed to access valuable information about the previous usage of the retrieved LOs. The module is basically a centralized search engine, capable of applying meta-searching techniques over the proprietary databases provided by a variety of LORs. The system is currently a module working in the Moodle_LS LMS, which is an enhanced version of the standard Moodle LMS, aimed at supporting personalized e-learning (Limongelli *et al.* 2011b). The system acts as a search interface towards several standard compliant LORs.

[1] http://elgg.org (Accessed July 24 2015).
[2] http://www.moodle.org (Accessed July 24 2015).
[3] http://dotlrn.org (Accessed July 24 2015).
[4] http://www.claroline.net (Accessed July 24 2015).
[5] http://www.atutor.ca (Accessed July 24 2015).

Such integrated interface also allows for sharing data about the actual usage of a selected LO in the whole set of courses managed through the LMS. In particular, some statistics show how many times the LO was included in courses, and how it was placed in each one of those courses (i.e. in what position with respect to other LOs in the course, thus providing information about the context of its usage).

In this way, the module acts as a recommending system, giving the teachers a ranking list of LOs.

Our module is placed in an operational setting enabling support to both the construction of traditional "one fits all" courses, and the management of personalized/adaptive courses.

With respect to the LO retrieval problem, we think that the main added value offered by our approach lies in the combination of two main aspects.

Firstly, the teacher would operate the retrieval, analysis and selection of LOs without having to leave the system. In particular,

- the module implementing our retrieval and analysis tool is integrated in Moodle_LS;
- the teacher can perform search-by-keyword(s) queries in a common interface, without having to adapt the search patterns to each distinct LOR;
- working on the result list, the teacher can perform the analysis and the selection of LOs to be included in the course under development.

The second aspect is the possibility for the teacher to exploit social data about the previous usage of LOs in the LMS. In particular,

- a shared repository is available, collecting the LOs used in all the courses managed by the LMS;
- when a list of LOs is returned from a search query, it is ordered according to a relevance ranking, expressing the amount of previous usage (if any) of the retrieved LOs in the LMS. This makes it possible, for instance, to select LOs that have been adopted (hence endorsed) by someone else in the LMS.

This paper presents an empirical evaluation of the module extending the previous one presented in Limongelli *et al.* (2012). The evaluation aims to test two research questions concerning the usefulness of the system:

(1) on the one hand we verify how the use of the system can speed up the teacher's work of searching LOs;
(2) secondly we see whether the recommendations given by the system are deemed valuable by the teacher (i.e. the suggested LOs are selected).

After reporting the experimentation we also describe the idea of extending the searching service to the framework of a social network of teachers, so as to make the exchanging and endorsing of learning material more focused on the characteristics and demands of a teaching community.

Before moving ahead, it is worthwhile better positioning this work in relation to the general area of support to human activities. One of the relevant aspects of course construction in an LMS, lies in the fact that it can be fruitfully conducted collaboratively by a team of teachers operating on the same course, sharing out the work. In such a team, a cooperation plan would bind the members, in that each member is aware of the pedagogical inclinations and content preferences of the others and would therefore operate accordingly. Such a plan provides a valuable support to each teacher's work, helping to decide what should be looked for in the LOR(s) and what should be selected.

On the other hand, the merits of this kind of collaborative work are, by definition, confined in the team setting, hence finding a way to extend them toward a social setting seems to be a valuable goal.

In this regard, we dare to think that the approach to social-based support to LO retrieval, presented in this paper, makes a significant step in extending to individual teachers a support similar to the team collaboration-based one.

While the teacher here would have no directions from an explicit plan of collaboration (nor such a plan could be expected in a social setting), s/he still could profit from implicit knowledge, shared in the LMS, about the LOs used therein. This seems to be extremely helpful, as opposed to leaving the teacher conduct retrieval and selection operations on his/her own.

This paper is structured as follows. In the next section we provide the reader with motivation and a rationale of the work. The successive section shows some present-related work. We then discuss the module we implemented for LOs search and retrieval from available repositories, placing it in the framework of the general activity related to the Moodle_LS LMS. In Section 5 we report on a first evaluation of the system. We then discuss the social aspects of our work , considering the use of the system in a social network of teachers, and draw some conclusions.

2. Motivations and Rationale

The work described in this paper is part of a wider research initiative aiming to enrich the system Moodle_LS, which we already developed in Limongelli *et al.* (2011b), so as to provide the teacher with further didactic tools, supporting course authoring, namely the activities to feed the background repository with appropriate LOs. In particular this allows: i) the definition of search patterns for the LOs, ii) the actual search/analysis operations in LORs, iii) the selection of suitable LOs to be added to the repository, iv) and the actual integration of new LOs in the LMS. The main points in the overall protocol are the following:

- the use of concept mapping can help the teacher coherently plan the course, that is during a phase when the direct work with LOs is still to come. Moreover, through a portion of the concept map of the course, the system can suggest suitable search patterns for the next phase of the process, namely LOs retrieval;
- the use of retrieval and analysis tools, made available in the system, allows the teacher to search for LOs suitable to cover the learning needs revealed by the course design. In particular the possibility of searching in different standard compliant LORs, at the same time, and through a unified interface (thus abstracting from the individual search protocols established in each distinct repository) seems beneficial for the teacher, as it allows for quicker and less cumbersome operations;
- the introduction of social elements in the system allows for inspection of usage information about the LOs retrieved in the search phase: the teacher can evaluate the "local success" of a given LO by looking at the number of times it has already been used (i.e. inserted as learning activity) in other courses. We are aware that several other elements ought to be considered for a comprehensive evaluation of LOs, such as, the cognitive level of the course, however, surmising that the courses in the system are homogeneous under the above-mentioned cognitive aspect, we think that numeric information can still be quite useful: the higher the number, the more a LO is endorsed by peers (colleagues).

Other information offered by the module regards the pedagogical neighbours of a LO's usage in the LMS. Namely, for each course in which the LO has been used, a summary of what LOs have been used before or after that LO is shown.

Such a combination of information, about "how much" and "where" a teacher's colleagues used a LO, can help the teacher speed up the course construction, while possibly drawing inspiration from qualified peers.

This paper concentrates on the description of the work done with respect to the second and third points above, i.e., the "use of retrieval and analysis tools", and the "introduction of social elements in the system".

3. Searching LOs from LORs: Systems and Methods

In the last years, due to the pace of growth of distance education, a lot of LORs have been posted in the Internet by public and private institutions in order to share didactic material among private and public instruction communities (Kärger *et al.* 2006). Each LOR presents a particular way of storing LOs, i.e., what and which metadata are associated with a given LO. This makes it difficult to support interoperability and exchange of teaching materials, and in turn slows down the production of a new course based on LOs searcher through LORs.

3.1. *Learning Object Repositories*

Among the various LORs we took the following under consideration:

- MERLOT [1] is probably the most well known LOR worldwide. It is a centralized repository containing metadata only, pointing to objects located at remote locations. It is a stand-alone repository, acting as a portal for LOs. In addition to providing search and categorization, MERLOT offers a peer review service provided by communities of experts in different subject areas, containing more than 50,000 LOs, continuously updated. This stand-alone repository provides teachers with the Content Builder tool that explains why and how to use online materials. This repository also contains SCORM compliant LOs, each one tagged with the IEEE-LOM tags.
- CONNEXIONS [2] is a dynamic digital educational system consisting of an educational content repository and a content management system optimized for the delivery of educational content. Connexions is one of the most popular open education sites in the world. It features more than 25,000 learning objects or modules in its repository and over 1,000 collections (textbooks, journal articles, etc.), which are used by over 2 million people per month. LOs are tagged in XML.
- WISC-ONLINE [3] is a digital library containing over 25,000 Web-based LOs. The digital library of objects has been developed primarily by the faculty of the Wisconsin Technical College System (WTCS) and produced by multimedia technicians who create the LOs.
- ARIADNE [4] is the widest LOR with about 500,000 LOs. It was organized as a European network some time ago; nowadays, the Ariadne Foundation aims to promote the

[1] Merlot is a Learning Object Repository, available at http://www.merlot.org (Accessed July 24 2015).
[2] Connexions is a Learning Object Repository, available at http://www.cnx.org (Accessed July 24 2015).
[3] Wisc On Line is a Learning Object Repository, available at http://www.wisc-online.com (Accessed July 24 2015).
[4] Ariadne Foundation, available at http://ariadne-eu.org (Accessed July 24 2015).

sharing and reuse of learning material through a network of LORs spreading worldwide. The LOs search engine supported in Ariadne is based on proprietary technology and allows the use of the query language SQI (Simon *et al.* 2005).

3.2. *Learning Object Retrieval and Filtering*

Most of the existing recommending systems for Technology Enhanced Learning have students as final users (as reported in Drachsler *et al.* 2015, and references therein). Here, we focus our attention on teachers, the intermediate and indispensable link in the chain of the learning process. Our module wants to help teachers to concentrate themselves on the content of learning material, rather than on the search activity, without having to deal with associated activities such as the selection of an appropriate repository, and the use of its search protocols. In our approach, through the Moodle system, teachers will be able to perform all the steps necessary to produce learning courses. The first step is the search of suitable learning material by means of a recommending system.

Our system performs a first retrieval phase by selecting LOs that match the keywords written by the user. The search is carried out into the local database that collects all the LOs coming from the most popular LORs previously described in 3.1.

LORs structure the material on the basis of a LOM representation. All the LOs have a variety of metadata that describe the content. In our system, the recommending process is twofold: firstly LOs are retrieved and sorted using the standard TF-IDF metric, implemented by the MyIsam search engine embedded into the MySQL DBMS[1]; subsequently, the returned LOs array is processed using a filtering technique to produce the final ranking of items, representing the recommendations to the teachers.

Two major approaches exist in information filtering: content-based filtering and collaborative filtering (widely described in Lops *et al.* 2011, and references therein). A content-based filtering system selects items based on the correlation between the content of the items and the user's preferences, while a collaborative filtering system chooses items based on the correlation between people with similar preferences. When the delivered information comes in the form of a suggestion, an information-filtering system is called a recommender system. We follow the criteria based on a collaborative approach that gives priority to LOs already chosen by users in the community, that are supposed to be more relevant than other retrieved LOs. In fact, in our case, the community of teachers in the same instance of Moodle (e.g. a school or university) shares common background and teaching methodologies.

3.3. *Sharing and Reusing Learning Resources in Learning Management Systems*

The need for sharing and collaboration among teachers, on learning materials, has been surging ahead for several years now. There is a great variety of contributions in literature. In this paper we point out some of such contributions and then report on two projects dealing with the management of LORs by using the Moodle LMS.

In Hamid *et al.* (2013), the discussion on the needs for simplifying access to information, knowledge, and learning material content is extended to the consideration of open educational resources; the proposed solution is related to the support to work group in

[1] http://dev.mysql.com/doc/refman/5.6/en/fulltext-boolean.html (Accessed July 24 2015).

the production of open material, through a web service based on grid technology for collaborative authoring.

In Kotsovoulou (2013) collaboration is met in a perspective that is more related to the interaction among students and teachers, yet of interest for the methods of development and sharing of learning resources. The paper evaluates the educational import of social bookmarking and tagging in the activities of a Java Programming Language bounded community of students and teachers. The system can be "used as a course tool to create a pool of negotiated and accepted keywords supporting appropriate, effective and efficient retrieval of Java resources".

Bottino *et al.* (2011) present a system thought up to allow teachers to develop learning activities through pedagogical planning. In the discussion it is shown how educational activities can be specified and structured, both theoretically and practically, in order to be implemented in a "learning situation". In particular this approach allows teachers to share their "products", fostering collaboration and exchange throughout their community.

Two interesting initiatives developed to be used (if not exclusively at least organically) in Moodle are the Sharing Cart Moodle plug-in (Hinkelman 2009) and the DOOR open source "maker" for learning objects repositories[1]. Moodle is open-source and uses international standards for sharing e-learning content, so course materials can be shared between teachers of different schools. However, packaging the materials and moving them is notoriously difficult for teachers with little technical experience. For that reason, a dedicated repository module for easy searching and sharing in the LMS has been created (Hinkelman 2009). This module acts as a Moodle plug-in. We notice that in the present proposal, the phase of sharing among teachers is entailed as a second step, following the phase of supported retrieval of LOs from LORs.

The DOOR project (Digital Open Object Repository) makes available an Open Source software supporting the creation of learning objects repositories. It is integrated in Moodle as a plug-in and allows to build a place where to store digital contents in the form of LOs. In the created repository, LOs can be searched for by a teacher, and they can be included in one's own course or embedded in one's instructional unit. The system is compliant with international metadata standards, and implements the specifications IMS Metadata 1.2.1 and Content Package 1.1.3. The DOOR-Moodle plug-in allows Moodle teachers to browse more repositories seamlessly from a single Moodle course, and then select and import LOs with their metadata.

4. System Overview and Description of the Module

The main contribution of this paper is in the design and implementation of a module for the operations of search/retrieval/analysis/selection of LOs. The module is directly applicable to Moodle, in the form of a *plug-in*, yet it is also part of our long standing activity of enrichment of the Moodle_LS LMS (being Moodle_LS in turn a module extension of Moodle, providing the support to course personalization and adaptivity).

Moodle_LS is *our* natural first choice as an application field for the module, although the module itself has not been devised specifically for uses limited to personalized e-learning systems: it could be integrated in any instance of Moodle as well, being an extension of the *URL* module that enables a teacher to provide a web link as a course resource.

[1] DOOR project, available at http://door.sourceforge.net/index.html (Accessed July 24 2015).

In this section we start describing the general architecture of the enhancements developed for Moodle_LS, and then focus on the module, to show it at work and to report on its implementation issues.

4.1. *The Functional Architecture of the Overall System*

Here we show design issues of the whole system supporting the definition of courses in Moodle_LS. We assume that:

- the preliminary operations have been dealt with by the teacher, i.e. giving a name to the course or stating its articulation (in the Moodle style, such as defining the list of days for publishing learning activities, or the list of topics under which the activities are to be placed);
- for simplicity, the course is going to be constructed by selecting several LOs, already existing in standard supported repositories (i.e. manageable by the system);
- the final operations, such as the course delivery to the enrolled students, are to be dealt with by the LMS after the course construction.

The following protocol is applied during the phase of course construction:

(1) basically the process starts from the definition of a concept map of the course designed by the teacher;

(2) the teacher is then supported in delimiting a part of such map. The system elaborates on the part of map selected and defines a query (set of keywords computed based on the map);

(3) the search engine then is fed with the query (while some parameters in the interface allow the teacher to delimit the area of search (i.e. the repositories to be searched). A *full-text* search is performed through the selected LORs and a list of LOs is returned;

(4) the list of retrieved LOs is shown to the teacher, ordered according to a twofold principle of relevance:
   - a *basic relevance* is computed directly by the full-text search algorithm, based on *tf-idf* (Baeza-Yates and Ribeiro-Neto 1999); this reflects how much and how well the query keywords are matched by the available specification of the LO;
   - a *local-didactic relevance* (local to the system, its courses and teachers) is computed as the number of times the LO has been already used in courses of the system.

   The local relevance is considered prevalent over the basic one, as the fact that colleagues have used a LO in other courses can bear pedagogical significance for the inquiring teacher (at least for those LOs that the system already "knows"). So, the list of retrieved LOs is shown to the teacher in such a way that the known LOs are listed first (according to local relevance), followed by the others (those still never used in the system) in basic relevance order;

(5) the analysis of the list of retrieved LOs can now start. The analysis of a LO can proceed based on the data provided by the system, and the possible direct inspection of the LO. The data about the LO include:
   - the local-didactic relevance mentioned above;
   - the indication of what other LOs (not necessarily in the list produced by the query) have been in fact used in the course, in the *neighbour* of the LO. By neighborhood we mean the area, of the learning object sequence implementing the course, preceding and succeeding the position of the selected LO. The

teacher can state a *distance* from the LO, receiving an indication about what LOs are placed in the course within that distance, before or after the given LO. For instance, if lo_2 is the immediate predecessor of the LO lo_1 then the two LOs have distance 1, and the immediate predecessor of lo_2 will have distance 2 from lo_1;

(6) when the analysis of a given LO is terminated, if the teacher requests to add it to the course, the addition is performed, in the position decided by the teacher.

Fig. 1 shows the system's functional architecture.

The main logical steps of the aforesaid process are also shown in the following sequence:

Concept Map Building The teacher arranges the concept map of the course by defining the prerequisite relations only (Novak 1990)

Search The prerequisite concepts defined in the concept map are used as keywords for searching into LORs and into the local data base.

Temporary Storage All the retrieved LOs are gathered into a local temporary database. The teacher looks and selects the most interesting ones.

Local Storage The selected LOs are suitably metadated through the "Teacher Assistant" environment provided in Moodle_LS and stored in the local database where a deeper analysis is possible (Limongelli *et al.* 2010) for future search.

In particular, when a query is launched, its terms are first searched in the Local Storage and then into LORs. Bear in mind that the Local Storage is shared by all the system users (teachers). Over time, it expands through the information about the LOs imported in the courses managed by the teachers, progressively raising its accuracy and significance as a reference source for the teachers.

4.2. *The Module at Work*

The module for LO retrieval we discuss in this paper consists in a partial implementation of the protocol described in the previous sub-section: in particular it supports the execution of points 3 through 6 of the protocol list.

So the module is supposed to be called from the course construction interface of Moodle (and of Moodle_LS), where the teacher has already determined in what section of the course a new LO, related to certain keywords, has to be added. Notice that, consistently with Moodle, the course can be partitioned in several ways: in our example we show a course divided into a sequence of *Topics* and we are going to search a LO to be added in one of such Topic sections.

Fig.2 shows the course construction interface, where the link *Add a new activity or resource* associated to *Topic 3* is to be selected.

The module then sparks into action, in Fig.3, prompting the teacher to select the LORs where to search for, and to insert the keywords to search with.

The search produces a list of LOs (Fig.4): for each LO it is possible to inspect the contents, see usage information, and have the interface ready for the LO's import, from the *Learning Object preview*, *Recommendations*, and *Import in Moodle* buttons, respectively.

Fig.5 shows an example of usage information. In this case the LO *Arrays in Java* was used 6 times in courses managed by the system. In particular, the distance of the analysis is 1, and two more learning objects have often been used before it (twice each) while in the list of the LOs used after it the most frequent is *Introduction to OO Programming in Java - Arrays and For Loops.*

Finally, in Fig.6, the interface is ready to proceed with the LO's import in the course (activated by the *"Save ..."* buttons).

We might now discuss the possible usefulness of the module, by means of a simulated example.

4.2.1. A Use Case

Let's assume that the teacher T0 is in the system, along with a number of other colleagues. Let's also assume that the teachers in the system are coming from both T0's school and other schools with more or less comparable study programmes (e.g. they have students of similar age and with similar pedagogical aims).

T0 is building up her/his course, and intends to make it as a collection of already existing LOs, to be retrieved from standard repositories supported by our module. So, T0 starts looking for one of the needed LOs, by filling in the module with appropriate keywords.

Here is the list of retrieved LOs: $\{lo_4, lo_3, lo_5, lo_1, lo_6, lo_2\}$. We have omitted the indication of numbers of usages of the LO, however the order follows the relevance criteria described in the previous sub-section.

T0 spends some time on lo_1 and lo_6, concluding that s/he doesn't like the former, and that the latter is even worse. T0 examines lo_2 and lo_3 (liking them) and only glances at lo_4 and lo_5 *"Maybe they are good, or may be not ... we'll see."*.

The analysis is hardly conclusive: T0 might select one or more items among lo_2 through lo_5. So, T0 starts compulsing the usage information, and sees that

- lo_4 is the most used, by far (T0 thinks) *"Well, maybe it is good after all.. I might be going to like it!"*;
- only lo_2 was never used in the system so far;
- in some courses, where lo_4 was used, even lo_3 was used (mostly *before* of lo_4);
- also lo_5 appears sometimes where lo_4 is, although more rarely than lo_3, and mostly *after* lo_4;

T0 deepens the analysis and it turns out that the uses of lo_5, mentioned above, were performed by teachers that T0 happens to know and, on the whole, thinks fairly high of. So, T0 is presently quite inclined to adopt lo_4 and lo_5 for her/his course, with the former appearing before the latter rather than the opposite.

T0 is now ready to *import* those LOs in the course and to proceed with the rest of the course construction. However, in the data T0 still perused at this time, a distressful factor comes to surface: considering the LOs used around lo_4 in other courses, it seems that, when lo_4 and lo_5 are both appearing, sometimes lo_1 and/or lo_6 are used (not always, and with lo_6 appearing more rarely). This makes T0 reconsider lo_1 and lo_6. It turns out that s/he might admit that lo_1, still being far from excellent, could be useful after lo_5. However T0 still abhors lo_6 and will never select it for her/his course.

At the end, T0 imports lo_4 in the course, and also selects lo_5 and lo_1 to appear, respectively, after lo_4 and after lo_5. In the process, T0 had the possibility of quickly retrieving and examining a set of interesting LOs; s/he was also helped, or "influenced", by her/his colleagues' use of LOs.

4.3. *Module Implementation Issues*

One issue in the implementation of the module was in the automated interfacing with several standard LORs. LORs are usually interfaced with the outer world through a proprietary technology: the queries each one responds to must be shaped according to the individual LOR protocol, as is underlined by the LOR interface. Summing up, they are hardly available to the kind of massive (automatized, programmable) interrogations we are interested in.

In order to access each LOR in a programmable fashion, we adopted crawler technology. A (web) crawler is a software device, aimed at systematically scanning a network of information sources, comparing the information found against some built-in criteria, and elaborating the data of interest.

So as to produce a combined list of all the LOs coming from the different LORs, we had to devise a web-crawler for each LOR, each one customized to the respective LOR. The characteristics we had to process for such customization are basically subdivided into two categories:

(1) the way the repository search engine presents the results of a search (in short, the shape of the web page sent as an answer to the query),
(2) the set of information shown for each LO in the answer list.

With respect to such criteria we analyzed the four LORs previously introduced.

As described in previous sections, our module shows a general search engine to the user; it then collects the user query definition and, for each one of the LORs selected for the query, activates the corresponding crawler. That crawler is responsible for activating the individual LOR search engine, collecting the results and returning them to the module common user interface. The results coming from all the LORs are then shown (through a web page) to the user for further elaboration.

Basically, the crawler dedicated to one of the supported LORs works in three steps:

(1) it receives the query from the module interface and performs it on the associated LOR;
(2) it analyzes the answer that is usually a web page (an html document), produced by the LOR's search engine. In such document special patterns of information can be identified, which are characteristics of the LOR at issue; such patterns are identified and a list of LOs is acquired by the crawler for further elaboration.

    Regarding the different patterns that have to be treated in different LORs' search engine: the html page returned by Merlot search engine lists the selected LOs as a set of `<tr>` elements (rows of a table in html) rather than by other formats. That is an example of the patterns that the crawler has to recognize and manage, in order to extract the relevant data from the search results;
(3) after having distilled a list of LOs, however, the crawler has plenty of other information to manage before being able to send out its results.

    In particular, for each LO, it is once again a matter of i) connecting to the related web-page, ii) analyze the html document, and iii) extract information: the information is now a set of metadata, as they are managed and served by the LOR. Finally, the crawler can compose an xml document, conveying all the extracted information about each one of the LOs listed in the LOR search answer.

Eventually, the lists of LOs sent back by each crawler are integrated: in principle, each LOR describes its LOs using a different set of metadata; many of such metadata are different just in the name and not in the semantics. However, there might be cases in which something that is managed here is missing there: it is, for instance, the case of the "rating" metadata, expressing the assessment of the LOR users on the LOs, which is not managed in some of the LORs we analyzed.

After the above mentioned phase of data integration, the module can perform additional elaboration (mainly the ordering, according to the relevance criteria) and finally display the search results for the teacher.

5. Experimental Evaluation

In this section we propose the empirical evaluation of the system. It enables the analysis of all the LOs retrieved by means of a keyword-based search engine that basically computes a ranking of the retrieved LOs, taking into account the actual usage of them in all the courses currently managed by the LMS, together with the tf-idf metric, in order to overcome the cold start problem. As explained in Section 4, the metric used for the LOs ranking is the number of occurrences of each LO in all the courses running in the LMS at query time. The rationale behind the use of this kind of metric is that the more a LO is used in distinct courses, the more it could be used by another teacher.

In addition, the module interface displays the didactic positioning of the selected LO through the courses in which it appears. Namely, being the course divided in sections, as usual in Moodle, and being the LO located in one of such sections, the distribution of other LOs occurring in preceding and following sections is shown, up to a didactic distance (configurable by the teacher). This addition to the base metrics relies on the assumption that LOs appearing close enough to the LO at hand, are relevant for a further examination by the teacher, or at any rate they are more relevant than those occurring at a greater distance.

In the following subsection we discuss the *cold start* problem, arising in the experimentation of a context, such as the one of our module, where feeding the system is a pre-requisite for its usefulness. Then we describe the experimental settings, the applied statistical model, and the analysis of the results.

This part of the work expands a preliminary evaluation (Limongelli *et al.* 2012) where the users agreement was tested by a sample of 15 teachers. Here we address the following research questions:

(1) *Does the system speed up the work of the teacher in the searching activity?*
(2) *Does the system really help the teacher in her course building activity?*

With the first research question we want to test whether the system is useful or not for saving time in the searching activity, while through the second research question we want to test the system's ability to recommend the didactic materials to be included in a course.

5.1. *The Cold Start Problem*

The *cold start problem* concerns the prediction accuracy of a recommendation system: in most cases the accuracy grows with the amount of gathered data (Shani and Gunawardana 2011). In our module, this problem appears at the time t_0, i.e., when there are no courses in the platform yet or when there are only empty courses. In fact, at this time all the retrieved LOs would have zero occurrences, all being *cold items*. Our proposal is to overcome this problem by ranking ties only, basing on their tf-idf value. In this way, when the system displays a list of retrieved Los, all having zero occurrences, the list is ordered anyway and the teacher can select a LO at the top rank. As the system gets filled with courses and then with LOs, at the time t_i, the ranked list will be displayed with respect to the didactic distribution of the LOs in the LMS at time t_i, as shown in

Fig. 7 where a ranked list is highlighted by a red box. Subsequently, this double ranking rule is used for all ties in the retrieved list: when two retrieved LOs present the same number of occurrences in the list, they are ranked on the basis of their tf-idf value. For a deeper insight in this research area the reader can refer to (Leung *et al.* 2008, Bobadilla *et al.* 2012, Kim *et al.* 2010).

5.2. *The Experimental Plan*

To perform the empirical evaluation of the system, we run the following experimental plan:

- *System availability*: the system was made available on the Web[1], as a 3-tier web application. In Fig. 8 is shown the Moodle 2.4 instance in which our module is embedded;
- *The sample*: a sample of 25 teachers was formed, randomly selected from 5 subject areas of interest from high schools and universities: Computer Science, Math, Literature, Art and History and having 5 teachers each;
- *Experimental Protocol*: first each group of teachers chose a topic to build a new course on. The following five topics has been proposed: *Introduction to arrays in programming*(Computer Science), *Pythagorean triple*(Math), *Oscar Wilde*(Literature), *The Futurism Phenomenon*(Art) and *The French Revolution*(History). Five empty courses for each topic in the Moodle platform were built, assigning each teacher to a single course to increase the probability of exchanging materials among teachers belonging to the same topic group, in order to test the system's ability to recommend LOs already in use in the platform. Secondly each teacher was requested to perform the following actions:
  - *first research question:* to report on time consumption during retrieval, in both cases of hand-made retrieval (performed going on each stated LOR to submit stated queries) and module-supported retrieval. To this aim, six queries were randomly selected from a set of 50 queries proposed by the sample;
  - *second research question:* to build a course by using the module (i.e. selecting LOs from the answer lists produced by the module out of the queries).

  Constraints were on the minimum number of queries to be launched, 5, together with the maximum time to be spent on the system, 90 minutes. Teachers were provided with a user manual and notes about the tasks to be performed;
- *Statistical model*: two kinds of statistical models were used. Firstly, a t-test (see for example (Witte and Witte 2012)) on the time needed to retrieve LOs by hand Vs. the time needed by the system, has been carried out (first research question). Secondly, the analysis of the system's ability to recommend LOs;
- *Research Conclusions*: the research questions are discussed;
- *Happy Sheet Questionnaire*: a simple happy sheet questionnaire was submitted to users, to record useful feedback in terms of usefulness of the system. It was composed of several questions regarding the user's professional background together with some Likert 5-points questions regarding the use of the system.

We measured the following stochastic variables:

- The time spent by the user in retrieving LOs both at hand Vs. through the system, in order to show the usefulness of the system from a practical point of

[1] Contact authors to access the system.

view: its ability to save time. A t-test was performed to this aim on a sample of six queries;
- The number of LOs retrieved by the system;
- The number of queries submitted by each teacher of the sample;
- The number of LOs actually inserted into courses among those retrieved by the full-text based search engine.

5.3. *Data Analysis*

As mentioned above, two experiments were performed by the sample, concerning the first and the second research question.

- *First research question.* Fig. 10 shows the time spent on average by the sample in the retrieving process for the six sample queries. The sample was divided into two groups: the control group, composed by 12 teachers and having the task of launching the six queries manually, i.e., directly through the LORs Web interfaces, and the experimental group, composed by 13 teachers, launching the same queries using our module. All the queries launched through our module were less time consuming. We performed a t-test in order to verify whether the difference between the two data distributions were due to chance (the null hypothesis H_0) or not (the alternative hypothesis H_1). The t-test was performed both on the single queries and on the entire set of queries. Tab. 2 shows the results for the first case while Tab. 3 shows the t-test results for the global experiment. In all cases, the alternative hypothesis H_1 was verified with a $p - value \ll \alpha = 0.01$.;
- *Second research question:*. In Tab. 1 we report some statistical parameters on the use of the system, concerning its utility in recommending LOs. The users launched 300 queries, retrieving 595 LOs, using 73 of them in their courses. So, about 12% were deemed worthy to be used in real didactic plans. Another characteristic is that 63% of the selected LOs were selected in the first 5 positions of the displayed LOs ranked list. From Fig. 9 (a) we see the distribution of the queries launched by users: 72% were launched between 11 and 20 queries. Fig. 9 (b) shows that more than two thirds of the users considered the system useful (or very useful), while 15% of the users found no (or little) usefulness in the system. To evaluate the performance of the recommending feature, we first used the *Average Precision* (AP) (Baeza-Yates and Ribeiro-Neto 1999), as the ranked precision metric that, for a given query q, places emphasis on highly ranked correct predictions (hits). This metric is used mainly to evaluate the quality of search engines: given a query q it evaluates the output ranked list of documents d, averaging the precision values from the rank positions where a relevant document was selected by the user. Subsequently, the mean across all the queries of all the AP values was computed, obtaining the Mean AP value (MAP). In our experiment a document is considered a relevant document only if it is selected by the user from the first five positions (five being also the number of LOs shown to the user, who uses a *next* button to see the rest of the list, in groups of five). In order to have an evaluation of the recommending capability we computed the AP values only for the queries where the user selected a document from the didactic list and not from the tf-idf one, to give more emphasis to the didactic aspect of the recommendation feature. Moreover, since many queries returned very few documents, often less than five, the AP metric in these queries was computed on the returned number, i.e., 2, 3 or 4. Following these guidelines, the MAP value was computed, obtaining $MAP = 0.71$.

5.4. *Discussion*

The first part of the experiment, i.e., the t-test on the retrieval time, confirmed the first research question: the system helps reduce the time required by the retrieval process of LOs from LORs (with respect to the case of direct use of LORs by the teacher).

The second research question, regarding the recommending capability of the system, was tested by means of the AP variable, obtaining an encouraging value $AP = 0.71$. In particular, we saw that the majority of the LOs selected by the sample mainly come from the top (first 5 positions) of the related ranked lists retrieved by the system. We believe that these results support our choices: both the basic metrics (tf-idf), used for the cold start, together with the didactic one based on LOs usage, show to be useful to retrieve and select LOs for building a course. It seems reasonable to expect that the most used LOs at time t_i will also be the most selected at time t_{i+1}.

Finally, from the happy sheet questionnaire submitted to the sample, we have that the majority of the teachers about 80%, found the system useful.

6. Conclusions: Towards a Social Network for LO Retrieval

Social networks are widely used to support sharing and collaboration in educational environments (De Marsico *et al.* 2013, 2011, Sterbini and Temperini 2008, 2012). Various systems have that aim, varying from dedicated platforms, such as Content Management Systems (CMS) and LMSs, to the pervasive Facebook or Twitter. The benefits of sharing and collaboration are not limited to the support of students learning activities; on the contrary, they extend their application to the teachers' work. Many network-based systems have been proposed to support teachers, showing the value of social features in helping teachers with their ordinary work. We may point out three main categories where such systems can be placed:

(1) First, there are several *social networks dedicated to teachers*. They allow to exchange learning materials as well as ideas about their use and development. Some of these social networks are based on proprietary software such as Promethean[1], that provides the community with learning material produced by the users themselves. Other systems are dedicated mainly to the exchange of general information. Examples of those systems are Classroom 2.0[2], or Teacher Recess[3], all allowing teachers, and in general educators, to share various typologies of materials related to the teaching activity. In the category of social networks for teachers we could also include branches, dedicated to education, of very popular software systems, such as *Google for Educators*[4], *TeacherTube*[5], and *Twitter for Teachers*[6].

(2) Then, there are some *evolving LMSs*, and the *Community of Practice* (CoP) phenomenon. On the one hand, classic LMSs, such as Moodle and Blackboard, in their recent versions show integrated social collaborative features, extendable -if

[1] Promethean Planet, teaching, sharing and support community, available at http://www.prometheanplanet.com/en/ (Accessed July 24 2015).

[2] Classroom2.0, social network for Web 2.0, Social Media, and Participative Technologies in the classroom, available at http://www.classroom20.com/ (Accessed July 24 2015).

[3] Teachers Recess, social network for teachers, available at http://www.teachersrecess.com/ (Accessed July 24 2015).

[4] https://www.google.com/edu/training/ (Accessed July 24 2015).

[5] http://www.teachertube.com/ (Accessed July 24 2015).

[6] http://rossieronline.usc.edu/twitter-for-teachers/ (Accessed July 24 2015).

not dedicated - to teachers. Moreover, the recent years have witnessed a general bend of network-based e-learning towards social and collaborative features, also with the development of CoPs presented in Wenger (1998), De Marsico *et al.* (2014). *Social Media Classroom*[7] is an a example of such a trend. Teaching contents and teaching experiences can be made by observing the activity of other teachers, and by tagging and rating learning material, i.e., by contributing to promote content that can be more interesting for the whole community.

(3) Finally, LORs should be considered again: several of them are available, each one with its individual interaction and rating method. In her/his lonely dialogue with the LOR, the teacher cannot be supported by any social information or consideration, related to her/his being part of a community of teachers. Using such a social characterization might instead be beneficial during the phase of LOs retrieval: it could make the teacher's search parameters more representative of the querying person and of the context of the searching query.

Coming back to the module for LOs retrieval we have described in this paper, we think that its features are of interest with respect to both the second and third categories mentioned above, and can effectively support the teacher in the burdensome work of LOs retrieval and selection. On the one hand, the module can be integrated in a state-of-the-art LMS such as Moodle. Moreover, the module allows to exploit a set of information of social nature, about LOs returned by a search performed in several distinct LORs. That set of information is presently limited: a teacher can see how other teachers used LOs of her/his interest (only in terms of numbers and localization of usage), yet it can contribute to help teachers speed up and focus on a part of their job in course construction. As mentioned in the introduction, the partially social-based approach to LO retrieval, which we presented in this paper, is a step toward providing individual teachers with a support similar to the team-collaboration-based one, in which the knowledge about others' preferences and inclinations can guide one's decisions about LOs selection. This approach is limited by the boundaries of a given LMS, and its extension through groups of interoperable, standard compliant, LMSs seems an affordable item for future work.

Another viable development we are planning relates to the possibility of applying further analysis over the LOs gathered in the local datamart; this can be done through classical OLAP functionalities, able to refine and deepen the information about the usage of LOs through the courses managed by the LMS.

A further avenue of development we are planning to follow regards the application of artificial intelligence techniques to the retrieval and validation of LOs, and implies the design of specific *Teacher Models*: recommendation, analysis and sharing of LOs can be significantly refined once the pedagogical preferences of the teacher are represented and managed in the system.

References

Baeza-Yates, R. and Ribeiro-Neto, B., 1999. *Modern Information Retrieval.* Addison Wesley.

Benson, V., Morgan, S., and Tennakoon, H., 2012. A Framework for Knowledge Man-

[7] Social Media Classroom and Collaboratory, learning material design and social media for teaching, available at http://socialmediaclassroom.com (Accessed July 24 2015).

agement in Higher Education Using Social Networking. *International Journal of Knowledge Society Research*, 3 (2), 44–54.

Bobadilla, J., Ortega, F., Hernando, A., and Bernal, J., 2012. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems*, 26 (0), 225 – 238.

Bottino, R.M., Ott, M., and Tavella, M., 2011. Scaffolding Pedagogical Planning and the Design of Learning Activities: An On-Line System. *International Journal of Knowledge Society Research*, 2 (1), 84–97.

De Marsico, M., Limongelli, C., Sciarrone, F., Sterbini, A., and Temperini, M., 2014. UnderstandIT: A Community of Practice of Teachers for VET Education. *In*: *WEBIST 2014 - Proceedings of the 10th International Conference on Web Information Systems and Technologies*, Vol. 1 SciTePress, 338–345.

De Marsico, M., Sterbini, A., and Temperini, M., 2011. The Definition of a Tunneling Strategy between Adaptive Learning and Reputation-based Group Activities. *In*: *11th IEEE International Conference on Advanced Learning Technologies (ICALT 2011)* IEEE Computer Society.

De Marsico, M., Sterbini, A., and Temperini, M., 2013. A Framework to Support Social-Collaborative Personalized e-Learning. *In*: *Human-Computer Interaction. Applications and Services*, Vol. 8005 of *Lecture Notes in Computer Science*, Las Vegas, NV, USA, July 21-26. Springer Berlin Heidelberg, 351–360.

Deed, C. and Edwards, A., 2010. Using Social Networks in Learning and Teaching in Higher Education: An Australian Case Study. *International Journal of Knowledge Society Research*, 1 (12), 1–12.

Di Martino, B., 2009. Semantic Retrieval of Learning Objects with Schema Matching. *Journal of e-Learning and Knowledge Society*, 5 (3), 49–58.

Drachsler, H., Verbert, K., Santos, O., and Manouselis, N., 2015. Panorama of recommender systems to support learning. *In*: F. Ricci, L. Rokach and B. Shapira, eds. *Recommender Systems Handbook 2nd edition*. Springer.

Gasparetti, F., Micarelli, A., and Sciarrone, F., 2009. A web-based training system for business letter writing. *Knowledge-Based Systems*, 22 (4), 287–291.

Hamid, D.H., Alias, N., Omar, A.H., Islam, M.R., Saipol, H.F., Palil, S.Q., Ghani, A.C., and Ramli, N., 2013. Web-Based Service for Collaborative Authoring Learning Using Grid Portal. *International Journal of Knowledge Society Research*, 4 (2), 27–39.

Hinkelman, D., 2009. Sharing Learning Objects within a Teaching Team: How to Use Moodle for an In-House Repository. *The Japan Association for Language Teaching*, 33 (4).

Kärger, P., Ullrich, C., and Melis, E., 2006. Integrating Learning Object Repositories Using a Mediator Architecture. *In*: W. Nejdl and K. Tochtermann, eds. *Innovative Approaches for Learning and Knowledge Sharing.*, Vol. 4227 of *Lecture Notes in Computer Science* Springer Berlin Heidelberg, 185–197.

Kim, H., Ji, A., Ha, I., and Jo, G., 2010. Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation. *Electronic Commerce Research and Applications*, 9 (1), 73 – 83.

Kotsovoulou, M., 2013. Collaborative Tagging of Java Learning Resources: Bridging the Gap between Teachers and Students. *International Journal of Knowledge Society Research*, 4 (1), 12–29.

Leung, C.W., Chan, S.C., and Chung, F., 2008. An empirical study of a cross-level association rule mining approach to cold-start recommendations. *Knowledge-Based Systems*, 21 (7), 515 – 529.

Limongelli, C., Miola, A., Sciarrone, F., and Temperini, M., 2012. Supporting Teachers to Retrieve and Select Learning Objects for Personalized Courses in the Moodle_LS Environment. *IEEE International Conference on Advanced Learning Technologies*, 518–520.

Limongelli, C., Sciarrone, F., Starace, P., and Temperini, M., 2010. An Ontology-Driven OLAP System to Help Teachers in the Analysis of Web Learning Object Repositories. *Information Systems Management*, 27 (3), 198–206.

Limongelli, C., Sciarrone, F., Temperini, M., and Vaste, G., 2011a. The Lecomps5 framework for personalized web-based learning: A teacher's satisfaction perspective. *Computers in Human Behavior*, 27 (4), 1310–1320.

Limongelli, C., Sciarrone, F., and Vaste, G., 2011b. Personalized e-Learning in Moodle: The Moodle_LS System. *Journal of E-Learning and Knowledge Society*, 7 (1), 49–58.

Limongelli, C., Sciarrone, F., Temperini, M., and Vaste, G., 2008. LECOMPS5: A web-based learning system for course personalization and adaptation. *In*: *MCCSIS'08 - IADIS Multi Conference on Computer Science and Information Systems; Proceedings of e-Learning 2008*, Vol. 1 IADIS, 325–332.

Lops, P., De Gemmis, M., and Semeraro, G., 2011. Content-based Recommender Systems: State of the Art and Trends. *In*: F. Ricci, L. Rokach, B. Shapira and P.B. Kantor, eds. *Recommender Systems Handbook*. Springer US, 73–105.

Micarelli, A., Sciarrone, F., and Gasparetti, F., 2009. A Case-Based Approach to Adaptive Hypermedia Navigation. *Journal of Web-based Learning and Teaching Technologies. Special Issue on Adaptive and Intelligent Web-based Educational Systems*, 4 (1), 35–53.

Nanni, U. and Temperini, M., 2012. eLearning for Knowledge Management in Collaborative Architectural Design. *International Journal of Design Sciences and Technology*, 19 (2), 105–121.

Neri, M.A. and Colombetti, M., 2009. Ontology-Based Learning Objects Search and Courses Generation. *Applied Artificial Intelligence*, 23 (3), 233–260.

Novak, J.D., 1990. Concept Mapping: A Useful Tool for Science Education. *Journal of Research in Science Teaching*, 27 (10), 937–949.

Shani, G. and Gunawardana, A., 2011. Evaluating recommendation systems. *In*: F. Ricci, L. Rokach, B. Shapira and P.B. Kantor, eds. *Recommender Systems Handbook*. Springer US, 257–297.

Simon, B., Massart, D., van Assche, F., Ternier, S., Duval, E., Brantner, S., Olmedilla, D., and Miklós, Z., 2005. A Simple Query Interface for Interoperable Learning Repositories. *In*: *1st Workshop on Interoperability of Web-based Educational Systems, for the Fourteenth International World Wide Web Conference*, May 10–14. Chiba, Japan.

Sterbini, A. and Temperini, M., 2008. Learning from Peers: Motivating Students through Reputation Systems. *In*: *IEEE/IPSJ International Symposium on Applications and the Internet* IEEE Computer Society, Los Alamitos, 305–308.

Sterbini, A. and Temperini, M., 2012. Supporting Assessment of Open Answers in a Didactic Setting. *In*: *12th IEEE International Conference on Advanced Learning Technologies (ICALT 2012)*, Roma, Italy, 4–6 July 2012. IEEE Computer Society, Los Alamitos.

Wenger, E., 1998. *Communities of Practice: Learning, Meaning, and Identity*. Cambridge Un. Press.

Witte, R.S. and Witte, J.S., 2012. *Statistics*. John Wiley & Sons.

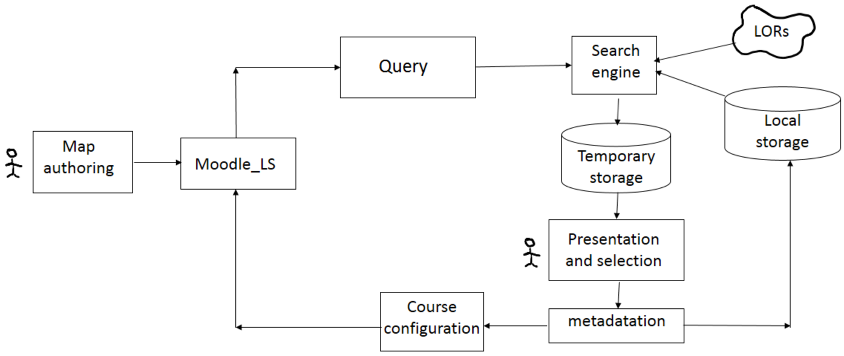


Figure 1. The functional architecture of the system. Local storage is populating as teachers search and select items from LORs.

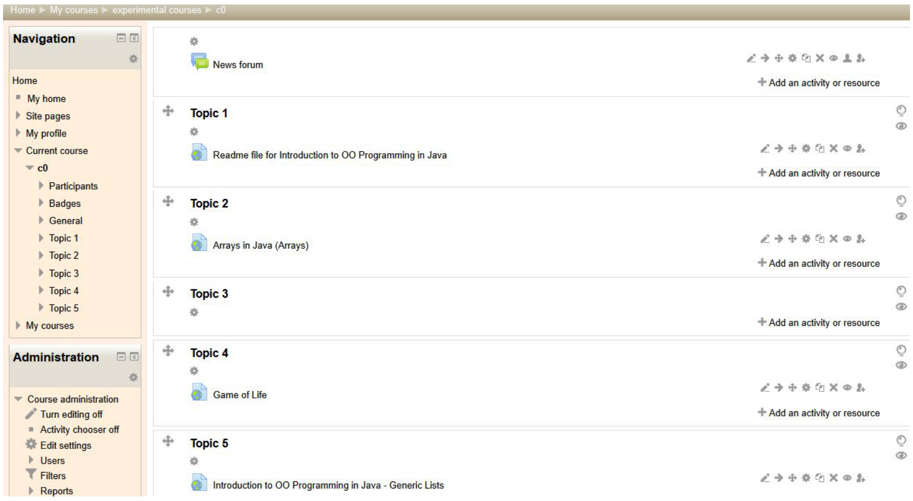
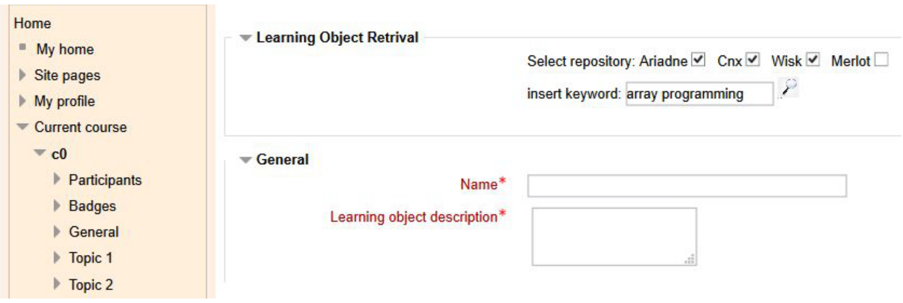


Figure 2. A new activity or resource has to be associated to *Topic 3*.



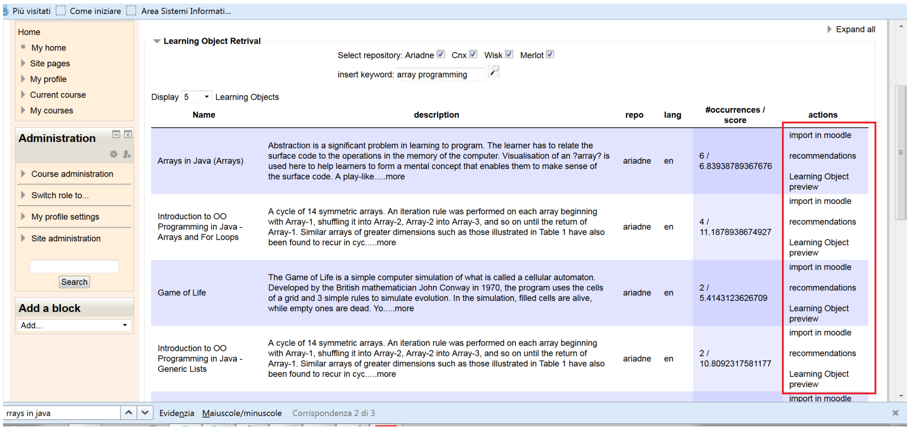
Select repository: Ariadne ☑ Cnx ☑ Wisk ☑ Merlot □

insert keyword: array programming

Name*

Learning object description*

Figure 3. The search interface.



Select repository: Ariadne ☑ Cnx ☑ Wisk ☑ Merlot ☑

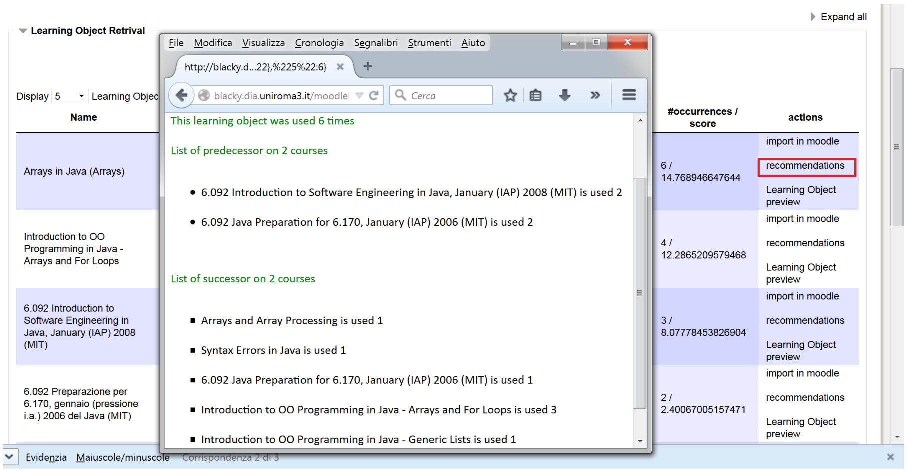
insert keyword: array programming

Display 5 Learning Objects

rrays in java

Name	description	repo	lang	#occurrences / score	actions
Arrays in Java (Arrays)	Abstraction is a significant problem in learning to program. The learner has to relate the surface code to the operations in the memory of the computer. Visualisation of an ?array? is used here to help learners to form a mental concept that enables them to make sense of the surface code. A play-like.....more	ariadne	en	6 / 6.83938789367676	import in moodle recommendations Learning Object preview
Introduction to OO Programming in Java - Arrays and For Loops	A cycle of 14 symmetric arrays. An iteration rule was performed on each array beginning with Array-1, shuffling it into Array-2, Array-2 into Array-3, and so on until the return of Array-1. Similar arrays of greater dimensions such as those illustrated in Table 1 have also been found to recur in cyc.....more	ariadne	en	4 / 11.1878938674927	import in moodle recommendations Learning Object preview
Game of Life	The Game of Life is a simple computer simulation of what is called a cellular automaton. Developed by the British mathematician John Conway in 1970, the program uses the cells of a grid and 3 simple rules to simulate evolution. In the simulation, filled cells are alive, while empty ones are dead. Yo.....more	ariadne	en	2 / 5.4143123626709	import in moodle recommendations Learning Object preview
Introduction to OO Programming in Java - Generic Lists	A cycle of 14 symmetric arrays. An iteration rule was performed on each array beginning with Array-1, shuffling it into Array-2, Array-2 into Array-3, and so on until the return of Array-1. Similar arrays of greater dimensions such as those illustrated in Table 1 have also been found to recur in cyc.....more	ariadne	en	2 / 10.8092317581177	import in moodle recommendations Learning Object preview
					import in moodle

Figure 4. List of suggested LOs and possible *inspection* actions on them.



This learning object was used 6 times

List of predecessor on 2 courses

- 6.092 Introduction to Software Engineering in Java, January (IAP) 2008 (MIT) is used 2
- 6.092 Java Preparation for 6.170, January (IAP) 2006 (MIT) is used 2

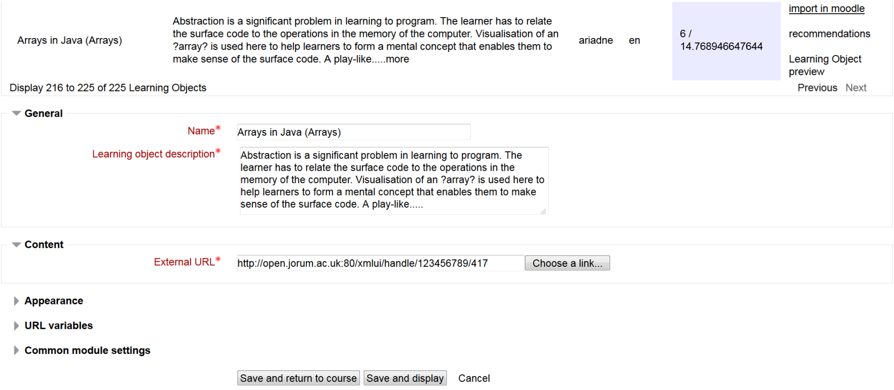
List of successor on 2 courses

- Arrays and Array Processing is used 1
- Syntax Errors in Java is used 1
- 6.092 Java Preparation for 6.170, January (IAP) 2006 (MIT) is used 1
- Introduction to OO Programming in Java - Arrays and For Loops is used 3
- Introduction to OO Programming in Java - Generic Lists is used 1

Display 5 Learning Objec

Name	#occurrences / score	actions
Arrays in Java (Arrays)	6 / 14.768946647644	import in moodle recommendations Learning Object preview
Introduction to OO Programming in Java - Arrays and For Loops	4 / 12.2865209579468	import in moodle recommendations Learning Object preview
6.092 Introduction to Software Engineering in Java, January (IAP) 2008 (MIT)	3 / 8.07778453826904	import in moodle recommendations Learning Object preview
6.092 Preparazione per 6.170, gennaio (pressione i.a.) 2006 del Java (MIT)	2 / 2.40067005157471	import in moodle recommendations Learning Object preview

Figure 5. Information about the usage of the selected LO in the overall LMS.



Arrays in Java (Arrays)	Abstraction is a significant problem in learning to program. The learner has to relate the surface code to the operations in the memory of the computer. Visualisation of an ?array? is used here to help learners to form a mental concept that enables them to make sense of the surface code. A play-like.....more	ariadne	en	6 / 14.768946647644	import in moodle recommendations Learning Object preview

Display 216 to 225 of 225 Learning Objects

General

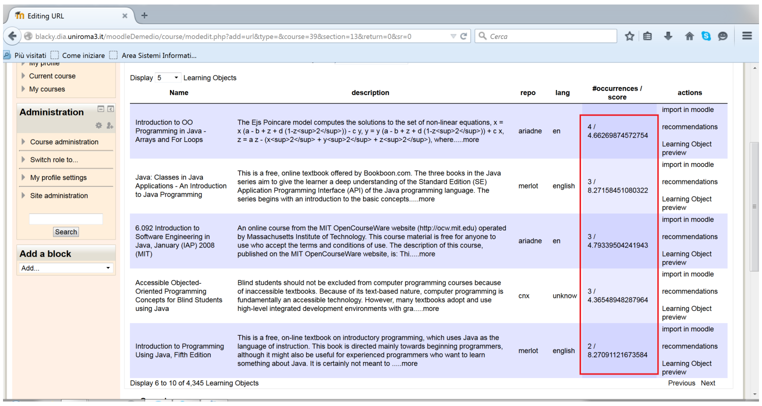
Name* Arrays in Java (Arrays)

Learning object description* Abstraction is a significant problem in learning to program. The learner has to relate the surface code to the operations in the memory of the computer. Visualisation of an ?array? is used here to help learners to form a mental concept that enables them to make sense of the surface code. A play-like.....

Content

External URL* http://open.jorum.ac.uk:80/xmlui/handle/123456789/417

Figure 6. The selected LO is automatically saved in the *General* Moodle field.

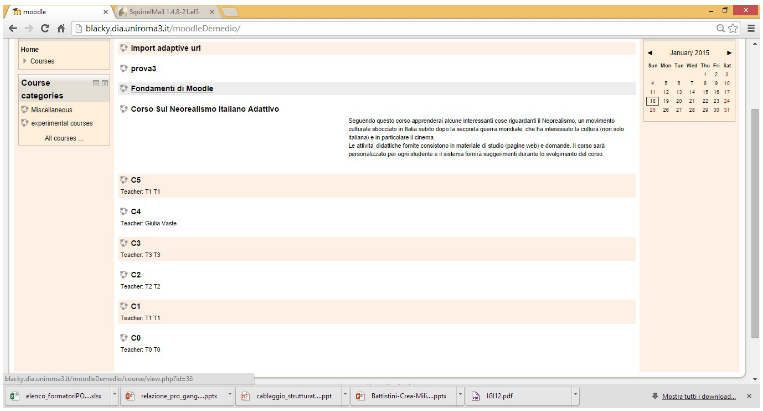


Display 5 Learning Objects

Name	description	repo	lang	#occurrences / score	actions
Introduction to OO Programming in Java - Arrays and For Loops	The Ejs Poincare model computes the solutions to the set of non-linear equations, x = x (a - b + z + d (1-z²)) - c y, y = y (a - b + z + d (1-z²)) + c x, z = a z - (x² + y² + z²), where.....more	ariadne	en	4 / 4.66269874572754	import in moodle recommendations Learning Object preview
Java: Classes in Java Applications - An Introduction to Java Programming	This is a free, online textbook offered by Bookboon.com. The three books in the Java series aim to give the learner a deep understanding of the Standard Edition (SE) Application Programming Interface (API) of the Java programming language. The series begins with an introduction to the basic concepts.....more	merlot	english	3 / 8.27158451080322	import in moodle recommendations Learning Object preview
6.092 Introduction to Software Engineering in Java, January (IAP) 2008 (MIT)	An online course from the MIT OpenCourseWare website (http://ocw.mit.edu) operated by Massachusetts Institute of Technology. This course material is free for anyone to use who accept the terms and conditions of use. The description of this course, published on the MIT OpenCourseWare website, is: Thi.....more	ariadne	en	3 / 4.79339504241943	import in moodle recommendations Learning Object preview
Accessible Objected-Oriented Programming Concepts for Blind Students using Java	Blind students should not be excluded from computer programming courses because of inaccessible textbooks. Because of its text-based nature, computer programming is fundamentally an accessible technology. However, many textbooks adopt and use high-level integrated development environments with gra.....more	cnx	unknow	3 / 4.36548948287964	import in moodle recommendations Learning Object preview
Introduction to Programming Using Java, Fifth Edition	This is a free, on-line textbook on introductory programming, which uses Java as the language of instruction. This book is directed mainly towards beginning programmers, although it might also be useful for experienced programmers who want to learn something about Java. It is certainly not meant to .....more	merlot	english	2 / 8.27091121673584	import in moodle recommendations Learning Object preview

Display 6 to 10 of 4,345 Learning Objects

Figure 7. The box highlights the column of rankings; the rankings are pointed out as the "number of occurrences in the system"/"tf-ldf value".



import adaptive url

prova3

Fondamenti di Moodle

Corso Sul Neorealismo Italiano Adattivo

Seguendo questo corso apprenderai alcune interessanti cose riguardanti il Neorealismo, un movimento culturale sbocciato in Italia subito dopo la seconda guerra mondiale, che ha interessato la cultura (non solo italiana) e in particolare il cinema.
Le attivita' didattiche fornite consistono in materiale di studio (pagine web) e domande. Il corso sarà personalizzato per ogni studente e il sistema fornirà suggerimenti durante lo svolgimento del corso.

C5
Teacher: T1 T1

C4
Teacher: Giulia Vaste

C3
Teacher: T3 T3

C2
Teacher: T2 T2

C1
Teacher: T1 T1

C0
Teacher: T0 T0

moodle
SquirrelMail 1.4.8-21.el5
blacky.dia.uniroma3.it/moodleDemedio/
Home
Courses
Course categories
Miscellaneous
experimental courses
All courses ...
January 2015
Sun Mon Tue Wed Thu Fri Sat
blacky.dia.uniroma3.it/moodleDemedio/course/view.php?id=36
elenco_formatoriPO....xlsx
relazione_pro_gang....pptx
cablaggio_strutturat....ppt
Battistini-Crea-Mili....pptx
IGI12.pdf
Mostra tutti i download...

Figure 8. The Moodle instance.

Table 1. Some data gathered from the sample.

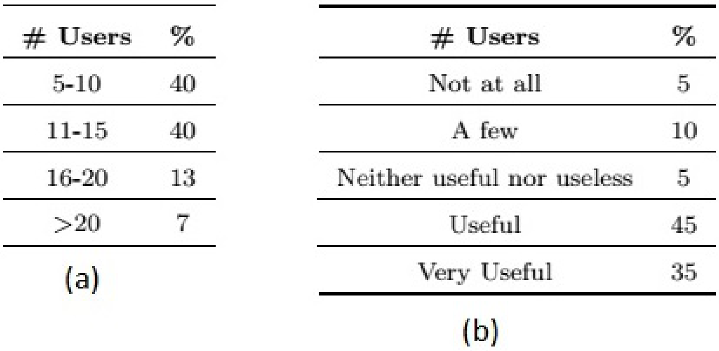
User	**# queries**	**# Retrieved LOs**	**# Used LOs**
1	12	25	3
2	15	45	4
3	10	20	2
4	9	20	3
5	13	35	0
6	11	25	2
7	7	15	3
8	11	10	2
9	15	25	2
10	15	45	4
11	15	25	7
12	15	10	2
13	12	25	1
14	6	30	8
15	13	20	1
16	8	15	2
17	16	15	1
18	17	25	3
19	11	35	3
20	12	20	3
21	9	15	2
22	13	25	7
23	12	30	4
24	15	25	2
25	8	15	2
Tot	**300**	**595**	**73**

Table 2. The t-test performed on each query launched by the sample. X_C and SD_C represent, respectively, the mean and the standard deviation of the control group, X_E and SD_E the mean and the standard deviation of the experimental group and the t column is the t Student variable. The last column p represents the p-values.

Query	$\overline{X_E}$	SD_E	$\overline{X_C}$	SD_C	t	p
Bernoulli Equation	10.78	2.56	13.21	2.1	3.6694	0.0003
Turing machine	18.9	3.31	95.16	24.5	15.4231	10^{-6}
Euler method	31.55	13.2	219.33	81.5	11.3721	10^{-7}
Java programming	8.94	2	12.23	5.5	2.8108	0.003
Michelangelo Buonarroti	13.45	5	24.37	12	4.2	0.00005
Emmanuel Kant	13.26	6	21.34	9	3.73	0.0002

Table 3. The t-test performed on all queries launched by the sample. X_C and SD_C represent, respectively, the mean and the standard deviation of the control group, X_E and SD_E the mean and the standard deviation of the experimental group and the t column is the t Student variable. The last column p represents the p-values.

Query	$\overline{X_E}$	SD_E	$\overline{X_C}$	SD_C	t	p
All queries	16.14	7.25	64.27	40.1	2.99	0.0079



# Users	**%**
5-10	40
11-15	40
16-20	13
>20	7

(a)

# Users	**%**
Not at all	5
A few	10
Neither useful nor useless	5
Useful	45
Very Useful	35

(b)

Figure 9. The use of the system by the sample (a) and the usefulness of the system as expressed by the sample through a 5-point likert scale(b).

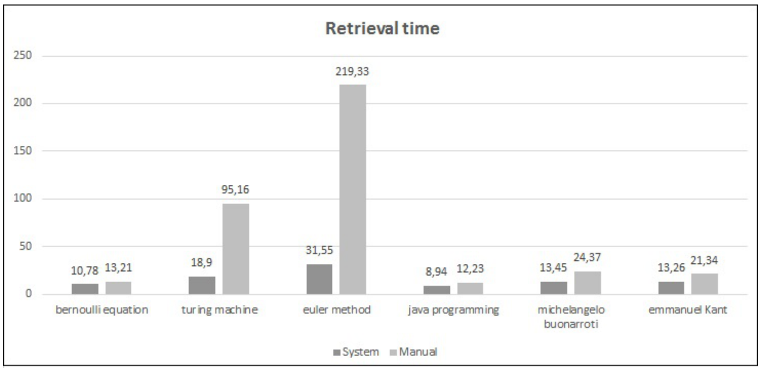


Figure 10. An example of the time consumed in the retrieving process: system (dark color) Vs. manual (light color). In the x-axis the query launched while in the y-axis the time measured in seconds.