OPEN U

BLOOM
European hub for online & blended learning, mobility & networking

Co-funded by the Erasmus+ Programme of the European Union

BLOOM hub Course catalogue and tagging system engine proof of concept

October 2022

BLOOM hub Course catalogue and tagging system engine proof of concept

This report is the deliverable of D2.10 of the OpenU project
By J. Posel, H. Steller (Freie Universität Berlin)

Co-funded by the Erasmus+ Programme of the European Union

Disclaimer: Funded by the European Commission in the framework of the Erasmus+ co-funded project OpenU (606692-EPP-1-2018-2-FR-EPPKAC-PI-POLICY).
The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Table of contents

1. Course catalogue, supported with semantics engine and tagging system and proof of concept

The BLOOM Hub is designed to be a portal supporting the aggregation of both shared course content, already existing at institutions on site, as well as a repository for newly designed and jointly developed materials. The collected information and its metadata need to be managed in order to remain accessible for the target audience. For obvious reasons, upscaling of the available content bears the risk of creating data silos, perhaps even data haystacks, which are prone to hide and loose the broad diversity available to users of the Hub or its instances in other contexts. To remedy this, it was imperative to only let courses be collected with accompanying structured information and metadata. This enhanced data pool enables a “rich” experience for the users, meeting their expectations in orientation within the course catalogue offers by giving not only a broad overview by title and hosting institution, but allowing to narrow down by topics, prerequisites, and additional attributes.

The required infrastructure and back-end mechanisms have been introduced in Deliverables D2.15, D2.7 and D2.11. To complement the feature, a tagging systematic has been weighted and a tag tool has been implemented.

While the course catalogue is ready and able to be fed with course information both by means of importing and parsing XML files, as described in Deliverable D2.15, or by manual addition of course information, it currently has been fed with informational data from Freie Universität Berlins Department of Mathematics and Computer Sciences courses from the current winter semester 2022/2023. This is mainly due to organizational circumstances rather than technical reasons. Indeed, in the current term there are no actual courses offered that are freely available and open to be joined by third party students. The course setting used within the Open U projects Work Package 3 experimentations were linked to the summer term of 2022, and these courses have by now ended.

We expect that, within the relevant universities and European university alliances, there will be agreements on which courses can be opened for students at partner universities, especially regarding the possibility of automation and the relevant local guidelines, like course access quota or the like.

2. Development of the tagging tool

In information systems, a tag is a keyword or term assigned to a piece of information (such as an Internet bookmark, multimedia, database record, or computer file). This kind of metadata helps describe an item and allows it to be found again by browsing or searching. Tags are generally chosen informally and personally by the item's creator or by its viewer, depending on the system, although they may also be chosen from a controlled vocabulary[1].

The main advantage is flexibility. Users can freely use tags to choose how to classify data. New categories can be created without additional requirements like place in hierarchy, as well as old ones can be replenished without significant changes in the structure.

Unlike traditional hierarchical categorization systems, where only one category can be assigned to an object, tagging allows assigning two or more different categories to a material at the same time. Another advantage of tags is that they are easy to learn. Moreover, the threshold for their usage is quite low .

[1] Some users, however, see tags not as metadata but as "just more content": Berendt, Bettina; Hanser, Christoph (2007), http://icwsm.org/papers/2--Berendt-Hanser.pdf. Retrieved Decvember 19th, 2022.

However, the usage of tags also has certain disadvantages. If the amount of information increases, it leads to an increase in the number of tags. Increase of tags leads to a less systematic approach to organization in the system, respectively. If there are few tags, then they are easy to navigate. But when there are hundreds of them, this becomes a much more challenging task. Moreover, the duplication of tags contributes to a decrease in order. Thus, the main disadvantage of tags is a decrease in the degree of organization of information.

An example of such misuse would be the following sentence:

#maybe #someone #could #get #used #to #use #tags #properly #because #it #is #very #annoying #to #have #too #many #and #unnecessary #tags (#grammar #rules #english)

2.1 Motivation behind developing Tag tool

One of the main goals of the Bloom Hub project is to create a centralized global overview for courses across all universities, where each university can post data about their courses. Students from each of these universities could get an opportunity to join any course at any university.

For example, a computer science student from a university of Spain needs to complete an algebra course, which is no longer available due to place constraints. However, there is a similar course in a university of France. The student gains access to a system, which provides a list of all similar courses across all universities, a registration tool to be able to participate in a course and also a testimony of completion of a course, which will be accepted in any of those universities.

Development of a project on such a scale is an unprecedented case. This and the fact that each university has its own unique system of study organization, makes the development a very difficult, sensitive, and delicate matter. Therefore, as an initial step to a big goal the team of FU Berlin got a task to introduce a system of tagging to SAKAI and create an overview for all sites with filtering by tags.

As a matter of a fact, LPS has its own system of tagging, which includes over 30 tags, and the number is growing. Each Course and Ancillary Course (potentially Whiteboard Site or Section) has an opportunity to get tagged in the system. However, Whiteboard Sites did not use a tagging system. Therefore, the goals were formulated:

- “Expand the tagging system of courses and ancillary courses onto Whiteboard Site system.
- Implement an export of tag data and tag-site/course-relation to Whiteboard.
- Create an overview for administration, where all Whiteboard Sites can be displayed and filtered by tags”

2.2 Tags in LPS

LPS possesses its own tagging system, which is used for systematization of courses and ancillary courses. Due to recent integration steps with the BLOOM Hub, the tag system was expanded with multi-language support.

Main information that contains Tag objects are Id, name (eng, ger, fr), description (eng, ger, fr), system info and, of course, relation data to courses or ancillaries in LPS, as shown in Fig. 2.

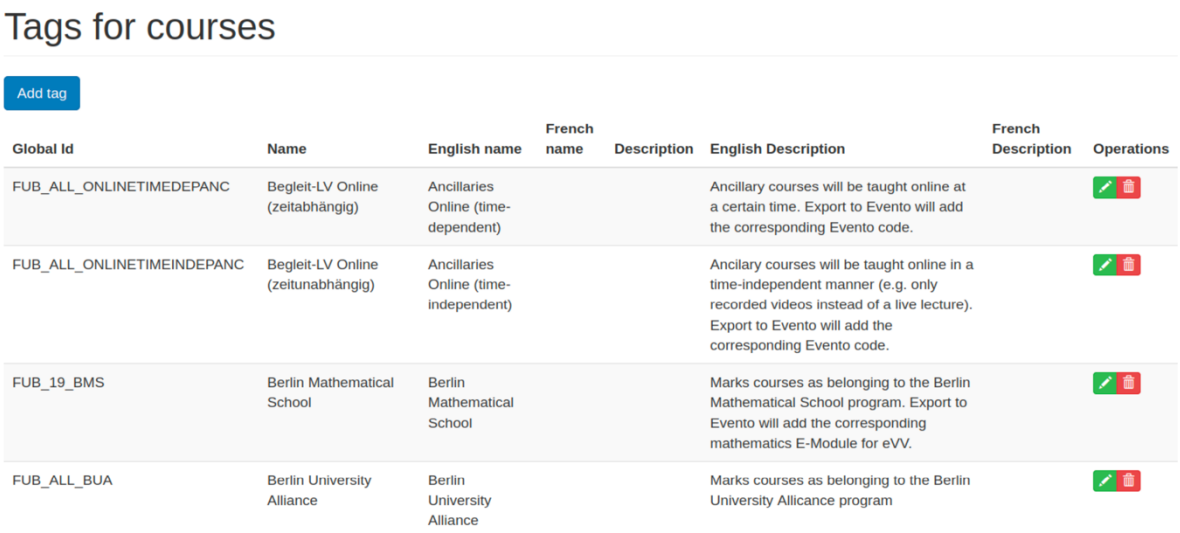

Fig. 2: Tags overview in LPS, Source: screenshot of LPS tag overview

One of the most demanding parts of the LPS system is synchronization with Evento, a local IT system in use at Freie Universität Berlin. Due to heavy restrictions of data usage, data flow must go through a very specific route: LPS must synchronize the majority of data regarding study organization (like Room Reservation, Course List, etc.) with Evento.

All course data from LPS must then be converted and exported to Evento. The converted version of course data from Evento must be further exported to Whiteboard. Due to the limitations of Evento, it is impossible to export a „lossless“ version of course data to Evento. Therefore, the Evento version of course data is not an accurate representation of LPS course data.

On the other hand, Whiteboard as a SAKAI-based system has its own data organization. LPS data must be exported and converted twice before it reaches the Whiteboard users. (LPS -> Evento, Evento -> Whiteboard) Considering the reasoning above, the course data must remain flexible in order to sustain the conversion and export processes. Due to flexible nature of tagging, tags are used in LPS to organize course data in multiple spheres:

- Evento export:
  Tags provide data organization in Evento, due to multiple limitations of Evento data structure. In fact, tags are amongst few instruments, which can provide reasonable data organization in such restricted conditions.
- Scheduler:
  As scheduler algorithms widely use a system of constraints, tags showed a complementary behaviour to constrain the system. Tags were integrated into scheduler without much effort and expanded its functionality
- Additional cases:
  Tags aid in specific cases, like adding an Online Tag functionality during COVID-19 restrictions, etc.

Unlike other systems, where the tagging system was used with an idea of expanding existing functionality, introduction of tagging system in LPS was caused by the necessity of providing support to crucial elements of the LPS system like Evento export and Scheduler. Therefore, tagging in LPS has a very important role in terms of functionality.

2.3 Tags in (native) Sakai

As Whiteboard is a SAKAI-based system, it is necessary to analyse if SAKAI already can provide the required functionality. As a matter of a fact, SAKAI already has an implemented system of tags.

As in LPS, main information that contain Tag objects are Id, name, description, Id of creator, timestamp of creation, timestamp of alteration and, of course, relation data to other objects in the system.

In SAKAI, the implemented version of the tagging system has a more general nature, than in LPS. The implementation consists of 2 parts: Description of a Tag as a DAO and an implementation of "Taggable" Interface, which, in theory, allows classes that inherit this "taggable" interface, to use the functionality of tags.

Elements that inherit "taggable" functionality:

- Assignments
- Forum posts
- Tests & Quizzes

However, as a matter of a fact, neither SAKAI Sites nor the bare Course Management System contain any integration with tags. Therefore, the required functionality concerning Bloom Hub Project does not exist in SAKAI by default and must be developed by external means, manually.

SAKAI introduced a unique solution for tagging system. Collections are sets of tags, where tags can be freely assigned to two or more different collections. In general, the collection system could be called "tagging-of-tags." One of the main drawbacks of the tagging system, namely the decrease in the degree of organization of information, when the number of tags is very high. Potentially this system is a partial solution to this problem through adding a rudimentary hierarchy of tags.

SAKAI tagging system provides tools for complete administration of tagging system. There are 3 ways to manipulate tags in SAKAI:

1. SAKAI Tool
   Creation, alteration, or deletion of tags in the special tool with a graphic shell for users (administrators). Also, this shell provides the functionality of managing collections: creation, alteration, or deletion of collections; adding or deleting tags to collections.
2. Quartz jobs
   Import and Synchronization of tags with external sources may be done through implemented Jobs, which allow fetching of tags from archives or external URLs.
3. DirectAPI
   Java-based interfaces for REST communication with SAKAI databases. Using specific requests according to documentation, DirectAPI provides access to server tag administration functionality without any graphical shell.

Concluding from the findings above: the tagging functionality was designed to be an extra functionality that can be added for any external tools for SAKAI by programmers with minimal efforts. However, in the SAKAI system itself it is barely used.

2.3 Weighting of options regarding the implementation of tags

The biggest question of adding tag functionality to Whiteboard was: "Should existing SAKAI tags be used or should a new data structure be implemented?" SAKAI tags have a tool that represents their

functionality. To answer this question, first an analysis of an existing tag tool and what benefits it can provide must be conducted.

- Administration of tags: Existing tool provides full access to basic editing tags: adding new ones, changing, or deleting existing ones. Although this functionality exists, it would be harmful to use it. LPS already has a tool that provides functionality of editing tags. As LPS is a core provider of data in all MyCampus project and has the top priority regarding data, it would be necessary to organize an export of tags from Whiteboard to LPS, and instead of one-way synchronization, two-way synchronization would be necessary. This would greatly increase the complexity of the problem (going from the consistency of tag data, to even new responsibilities for administration stuff) and would also violate the hierarchy of systems (Whiteboard is a "subordinate" system for LPS)
- Tag class fields: Even if both LPS and SAKAI systems contain java classes and DAOs, which are called "Tag", their content is not the same. They have similar fields, like "id," "name" or "description". However, during recent updates, LPS tags got a new language-based structure of names and description. Those changes were part of integration of SAKAI and LPS into the Bloom Hub Project, which emphasis is to provide an access to lectures of universities across all Europe. Therefore, due to language expansions, SAKAI DAOs themselves must have been altered in order to satisfy the demands of multi-language project. Which means alteration of SAKAI code on core level, which could lead to unpredictable consequences respectively. Other fields of SAKAI-Tag would contain information of by whom and when the tag was created. The problem is, this information is neither needed, nor welcomed in Whiteboard.
- Collections: SAKAI has a functionality of gathering tags into united sets of tags, namely collections. In the future this could be a very useful functionality, because one of the weaknesses of tagging is a loss of systematic approach, if a number of tags rises. However, as LPS data is in priority, such functionality must be implemented in LPS first, in order to export it to Whiteboard. LPS neither has such functionality implemented, nor is it on the waiting list at the moment.

Summarizing the findings above, the implemented functionality of the existing tool either is not required for purposes of the project or would cause additional complications.

Furthermore, a new tool would benefit of being more flexible in construction and implementation (one way sync), as well as from the possibility to use "the template" with technologies of LPS (acceleration of development, easier to maintain).

The next discrepancy came out during the planning on how the data about relation between tags and courses/sites should be organized.

Properties:

- SAKAI uses properties in order to sort, categorize, filter or search data on sites (existing functionality)
- LPS already populates export data with properties (semester, lecturers, departments, etc.). Just add another property.
- SAKAI already extracts properties into a special table in db. Each property has its own row (site_id, property_name, property_value) No need to change anything on SAKAI side
- Restricted to API of SAKAI Sites (fetching Sites does not include properties, after initial fetch can fetch all properties for each Site only)

Separate table:

- One-to-Many Relation (site_id, tag_global_id)
- Flexibility (benefits of created template: can create own repository and define specific SQL Queries, any combinations of sites and tags available to be defined)
- Performance (SQL optimizations to find only specific sites by tags)

Considering all the findings, no clear choice on what solution should be chosen could be made. Therefore, the decision was made to implement both solutions, starting with the one with properties as a simpler one.

2.4 implementation of tags

Implementing transfer of tag data required the following adjustements:

1. LPS: unlike Courses, Tags belonged to data with a relatively small amount of relations to other database objects, similar to such objects like Faculties or Departments. Therefore, the first step of implementation was to inject tag data into export of faculties, departments, and other similar data. Due to many similarities, the implementation of tag data injection was made similarly to other data in this section of export. Due to maturity of the code, the injection did not require extensive testing and went without significant issues.
2. Eval-Lehre: The import part went similar to LPS export: all data objects already had a separate table and an implemented extraction of data after import. The task was to implement the similar code to others: creation of TagDAO and injecting data into a fresh established table. The tricky part here was, that the import process was spread across different parts of Eval-Lehre, therefore required attention and accuracy to correctly name and place the implemented code. Again, the testing went without significant issues due to maturity of the import code.

Next the implementation of the transfer of relations between SAKAI Sites and tags data was undertaken:

- Properties:
  1. LPS: Extended Courses Query with Tags, Injected tag data into properties
  2. KVV-Manager correctly reads all properties including Tags and adds them to property table
- Separate table:
  1. Instead of packing tag data into properties, created another element, which contains tag ids on the same level with properties.
  2. Created a separate table for relation data between tags and sites
  3. Implemented population of this table with ids of tags and sites

Then a frontend tag tool for users (administrators) was implemented, as shown in Fig. 3:

1. Created a copy of the template and adjusted its configuration
2. Created connection to tag data, implementing Tag Entity
3. Added already implemented APIs (Kitchensink-API) to fetch sites from database
4. Implemented fetching and parsing of tag data
5. Implemented a View for users, which included a table for sites and a form to show sites by faculty/department and filter by tags
6. Added several frontend frameworks like select2 or dataTables framework to provide a responsive design and better user experience

7. Due to relative “youth” of methods used in the implementation, more thorough testing was required, however no significant errors or deviations in behaviour were found

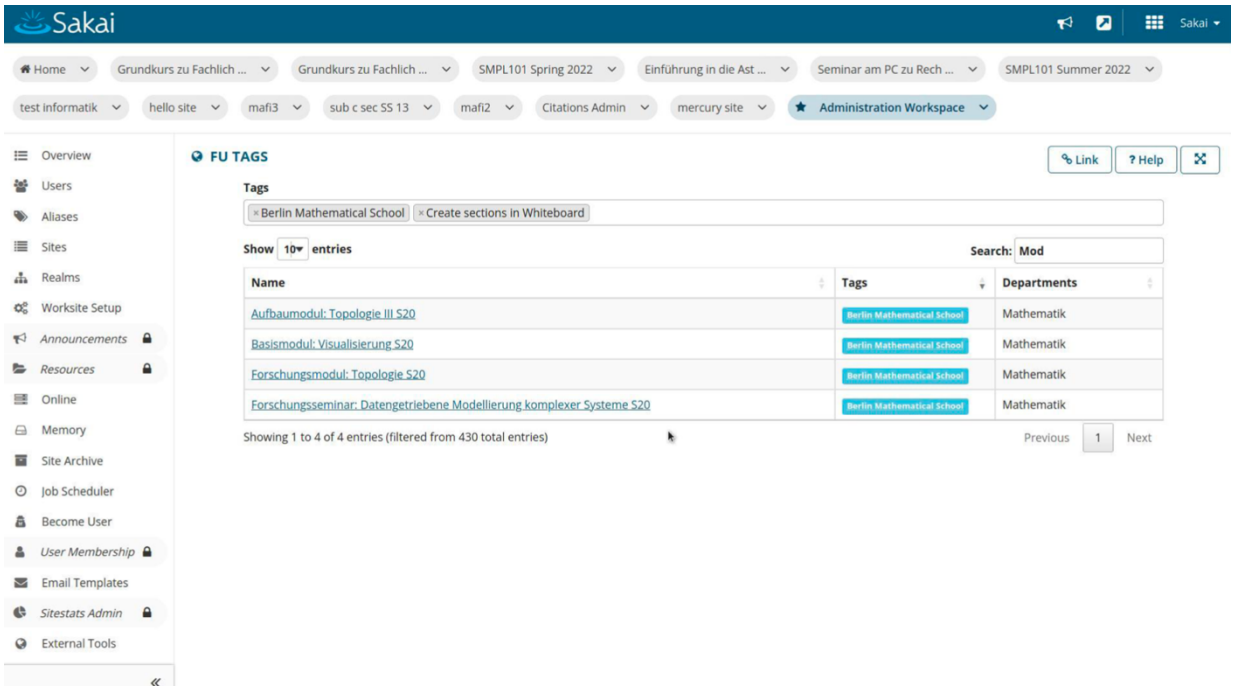

Fig. 3: Tag tool. Source: screenshot of locally installed system

The tag data transfer implementation went quite fast due to the existence of similar implementations. The implementation of the template allowed to significantly speed up the pace of development of the new tag tool and avoid many “pitfalls,” due to well-known technologies and solutions from LPS.

We believe the tag tool should provide a solid foundation for further development of fully functionable global course overview as part of the BLOOM Hub.

Project Number: 606692-EPP-1-2018-2-FR-EPPKA3-PI -POLICY

Project Duration: 47 months

Start date: 20-02-2019

End date: 19-12-2022

Coordinator: Université Paris 1 Panthéon-Sorbonne

Co-funded by the
Erasmus+ Programme
of the European Union