DELIVERABLE Project Acronym: DARIAH-RC Grant Agreement number: 2014-1-IE02-KA203-000125 Project Title: DARIAH Reference Curriculum D13 Technical Report on Project Portal Development Revision: 1.6

Matej Durco, Tanja Wissik

Funded by the Erasmus+ Programme of the European Union

DELIVERABLE

Project Acronym: **DARIAH-RC**

Grant Agreement number: **2014-1-IE02-KA203-000125**

Project Title: **DARIAH Reference Curriculum**

D13 Technical Report on Project Portal Development

Revision: 1.6

Authors:

Matej Ďurčo (Österreichische Akademie der Wissenschaften, Austria)

Tanja Wissik (Österreichische Akademie der Wissenschaften, Austria)

Contributors: Agiati Benardou (Athena research and innovation center in information communication & knowledge technologies, Greece), Claire Clivaz (University of Lausanne and Swiss Institute of Bioinformatics, Switzerland), Marianne Huang Ping (Aarhus University, Denmark), Stef Scagliola (Erasmus Universiteit Rotterdam, Netherlands), Susan Schreibmann (Maynooth University, Ireland), Toma Tasovac (Belgrade Center for Digital Humanities, Serbia), Dalibor Pančić (Österreichische Akademie der Wissenschaften, Austria).

Project funded by the Erasmus+ Programme	
Dissemination Level	
PU	Public

Revision	Date	Author	Organisation	Description
0.1	18.08.	M. Durco		
0.2	01.09.	T. Wissik		
0.3	12.10.	M. Durco		reorganized
0.4	19.10	T. Wissik		Content writing
0.5	20.10	T. Wissik		Content writing
0.6	23.10	D. Pancic		Data Model
0.7	27.10	T. Wissik M. Durco		Content writing
0.8	28.10.	T. Wissik		Content writing
0.9	29.10.	T. Wissik		Content writing, reorganized
1.0	30.10.	T.Wissik		Content writing, new data model, content consolidation
1.1.	31.10.	T. Wissik		Content consolidation, alignment with new preliminary version of WP2 user requirement report
1.2	02.11	T. Wissik		Final formatting
1.3	07.03	T. Wissik		Graphics, Content
1.4	17.03	T. Wissik, M. Durco		Graphics, content, incorporation of consortium feedback
1.5	18.3	M. Durco, T. Wissik, M. King		reorganization of requirements, and sections, revised text, English language revision
1.6	18.3.	M. Durco, T. Wissik		clean up

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Table of contents

1 Introduction

The project will deliver training materials for the digital arts and humanities in different languages and make them available via an online e-learning platform. This report elaborates on the implementation of such a platform. It describes the main user scenarios, it collects user and technical requirements, defines the data model and functional specification and explores technical solutions. The report has been created within WP 4 Infrastructure Development with input from all partners, especially WP 2 (user requirements).

As a first step, we performed desktop research on what kind of solutions and projects on portals for training materials exist and what kind of systems they are using.

The second step was an evaluation of different tools. There are different evaluation methods and criteria for e-learning systems (e.g. Kurilovas & Dagiene 2009). We chose to start from user requirements and a mapping of the user requirements from WP2 to functionalities available in existing systems. Finally, we determined which solution would suit the requirements and other circumstances within the project best.

2 User scenarios & requirements

We distinguish three types of users (or roles) for the platform:

- Teachers/trainers/providers of training material
- Students
- Administrator

2.1 User scenarios

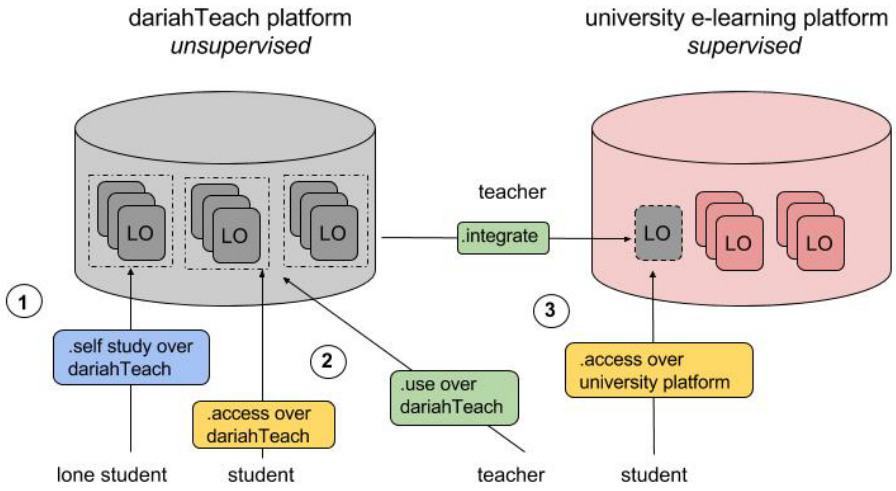

Fig. 1: Different user scenarios

For the project we have identified three main user scenarios:

- Use of learning material by lone students for self-study;
- Use of learning material by teachers in an already existing course using the dariahTeach
- platform;
- Use of learning material by teachers in an already existing course by using the material on the institutional e-learning platform.

2.2 User requirements

The development of the platform is guided by the user requirements deliverables as they were identified in a survey with potential users in WP2 (see deliverable D7). We considered user requirements not only from the section “I. Platform” of the deliverable, but also section “II. Modules” and “Others”. As the report concentrates mainly on user requirements from the instructor's point of view, WP4 carried out an additional small focus group discussion on possible user requirements from the student’s point of view. Some more user requirements came up during the discussion within the project consortium.

Additionally some of the user requirements collected in WP2, repeat similar aspects in other words. Some requirements contradict each other or contradict some of the requirements given by the consortium or already defined within the project. For example the requirement, of the structure of the platform and the content in modules, lessons, units and sections and the user requirements “The platform should be a free-structure environment (“Like a blog with semantic capabilities”). In the cases of contradicting requirements, the consortium has to decide which should be prioritized.

In an effort to consolidate the findings, we reformulated the requirements slightly and grouped them into following categories:

• Content model / Content handling / Metadata

• User Interface & Interaction

• User Management

• General / Framework / Development

Some requirements were not clear and would need further clarification in the consortium. They are listed under the section.

This set of requirements shall serve as basis for the development / adoption of the platform. However, even though we reduced the original number, by merging similar requirements, we still receive a set of 37 requirements which is a relatively high number, given the available implementation resources. Thus it might be necessary for the consortium to set priorities, and the evaluated solutions will be matched against.

In the next step, we have translated the 37 user requirements into general standard LMS system functions or CMS functions/plugin and have given a concrete LMS System or CMS System example or plugin that supports the function. Due to the whole design of the project, we have looked only into Open-Source LMS and as examples we have chosen Moodle and LearnPress the primary candidates for implementation after the review in section 4. Some of the user requirements could not be translated into specific CMS or LMS functionalities because they refer

rather to procedural/organisational aspects of the project like user requirement no. 31 “Ensure sustainability by cooperation with DARIAH” or no. 27 “The platform should be user-tested during its development”.

Content model / Content handling

1 Support multilingual content

2 Support interactive material (e.g. not just static information), multiple-choice exercises

3 Support hierarchical structuring of the content

4 Allow the use of parts of the content in other courses

(ad-hoc grouping of material)

5 Allow to dynamically embed material from other sources (YouTube, GitHub, document repositories)

6 Allow audio as material

Metadata

7 Support LOM for describing the material

8 Indicate copyright / licensing / availability information

User Interface & Interaction

10 Multilingual interface

11 Download learning material

12 Support serendipity principle - find something you had not thought of, searched recommendation (“you may like this”)

13 Use metadata for search and navigation

14 Allow to integrate LOs into other environments

15 Allow to share courses/ material via social media

16 Indication how long approximately a task would take and what would be the next task (like a task menu)*

17 Display attribution/ license on every item/ page

18 Support collaborative processes / interaction between users (students and teachers), synchronous and asynchronous

19 Group calendaring / scheduling

20 Feedback option for users (students), like “I liked the exercise”

21 Support tests / assessment features (for courses and lessons)

22 Integrated XML Editor, also to validate XML Structure

User management & Access

23 Allow both anonymous and authenticated access

24 Means for role-based user administration

25 Provide a personal workspace & customisation

General / Development

26 Web-based

27 Usability (platform should be user-tested during its development)

28 Integrate external tools

29 Allow programmatic access (API) to the platform

30 Be explicit about what the system can and cannot do

31 Ensure sustainability by cooperation with DARIAH

Questionable

32 The Platform should be editable having open document forum

33 Platform should be a free-structure environment “like a blog with semantic capabilities”

34 Store all material in one place

35 Platform should include alternative ways to FAQ

36 Nobody searches for learning outcome.

37 Should not be isolated (“Always bad to be a silo”)

3 Data Model

Based on the current/ preliminary descriptions of modules we formulated a data model, i.e. what kind of entities we need to deal with and what are their properties and the relationships between them.

3.1 Main Entities

The data model consists of four main types of entities:

- Courses - Sections - Lessons - Parts of a lesson

Courses are a collection of sections. Sections are a grouping of lessons. Lessons are core structural elements. Parts of a lesson cannot be shared independently.

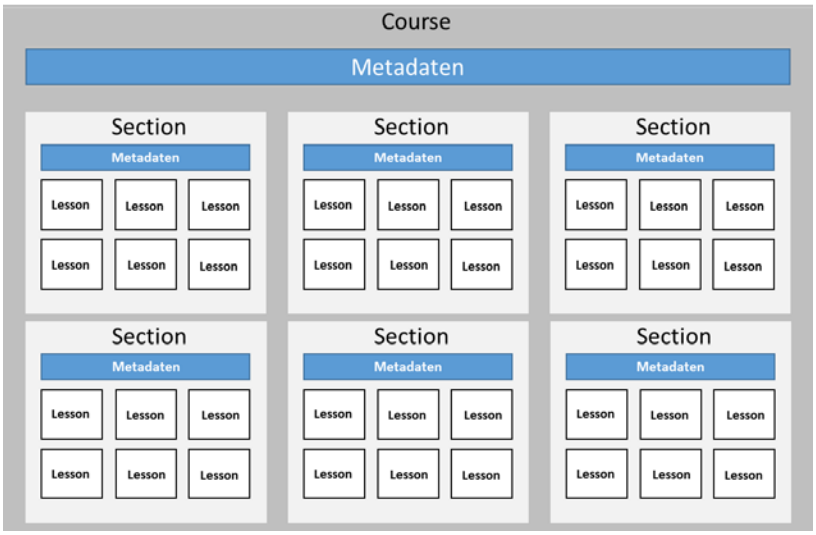

Fig. 2: Abstract model of the relation between course, section and lessons.

3.2 Learning Objects

While the above mentioned entities serve mainly for structuring of the material, the actual content is stored in the individual learning objects (LO). LOs can be of many different types, ranging from simple texts, images, audio, video, to interactive quizzes, code, applications, or even whole wiki-systems or databases.

Furthermore, especially for the multimedia material, the content can come from remote places, e.g. videos on YouTube, and needs to be embedded dynamically.

3.3 Metadata

In the last years, several open metadata standards have been developed. The IEEE Learning Object Metadata (LOM) defines the structure of metadata for learning objects. There is a separate section in LOM (9. Classification) caters for a flexible, generic way of classifying the material. Thus it is the decision of the content creator which classification system(s) will be used. In the context of DARIAH, the primary choices would be TaDiRAH1[1] or NeMo Ontology[2].

The integration of ECTS credits as metadata needs further investigation.

4 Overview of technical solutions for an e-learning platform

There are a wide range of projects and platforms that could be used for delivering training materials.

Generally, the solutions can be grouped into two different categories (however we have to be

[1] https://github.com/dhtaxonomy/TaDiRAH
[2] http://nemo.dcu.gr/

aware that these categories are not exclusive and most solutions feature a combination of the two aspects).

4.1 Static, repository-like solutions

In this kind of platform, training and learning material are only stored and can be retrieved and then used elsewhere. There are no collaborative or interactive functionalities foreseen like forums, chats, or feedback. These repositories are essentially the storage of learning materials especially created for e-learning (cf. Roy et al. 2010). Kurilovas (2010) calls this type of repository Learning Object Repository (LO Repository). Examples of such LO Repositories are: LearnAlberta[3] or OpenStax[4].

Another solution is the aggregator that does not store the learning objects themselves, but only the metadata which facilitates searching in the metadata categories like language, topic etc. linking to the original source.. Examples for such aggregators include: Ariadne[5] or Open Educational Resources (OER) Commons[6].

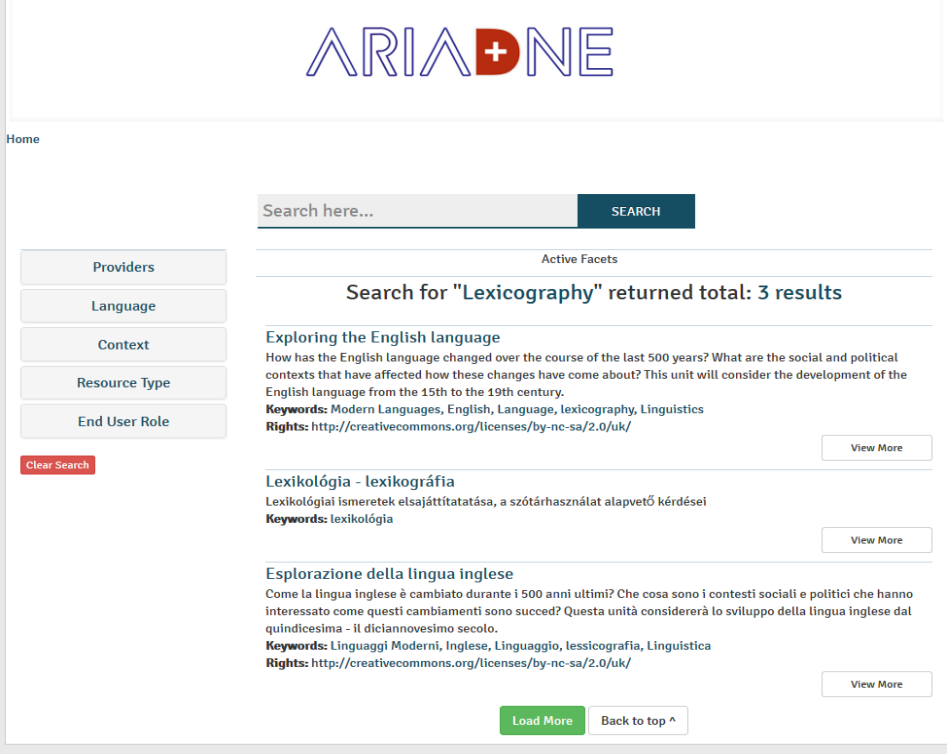

Fig. 3: Result of a search query for "Lexicography"

[3] http://www.learnalberta.ca/
[4] https://openstaxcollege.org/
[5] http://www.ariadne-eu.org/
[6] https://www.oercommons.org/

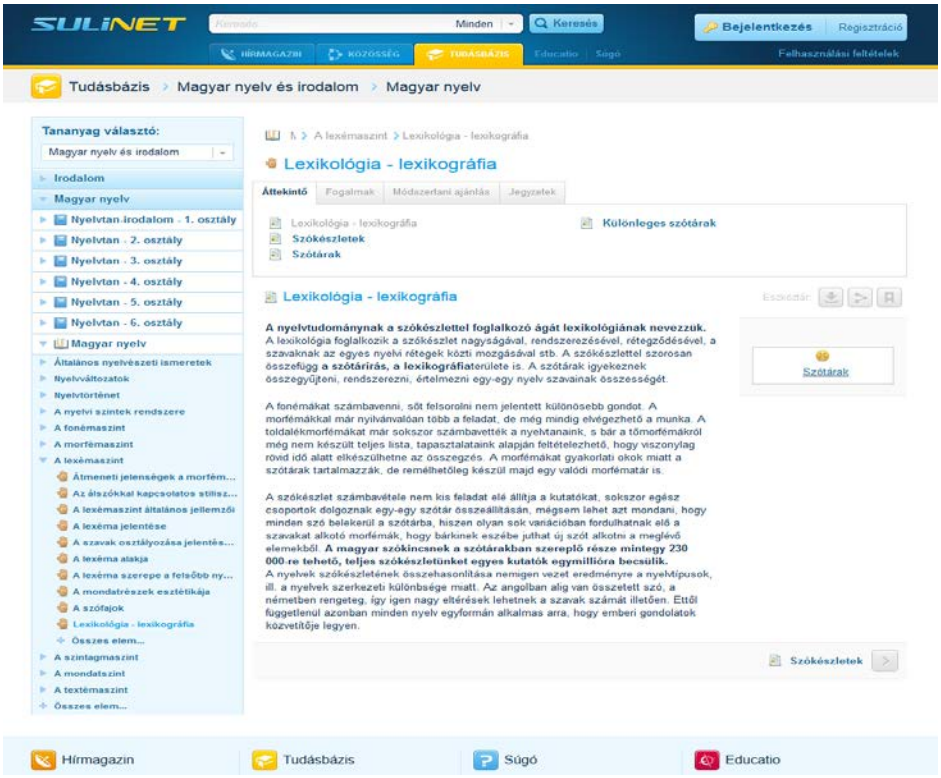

Fig. 4: Forwarding to the selected learning object

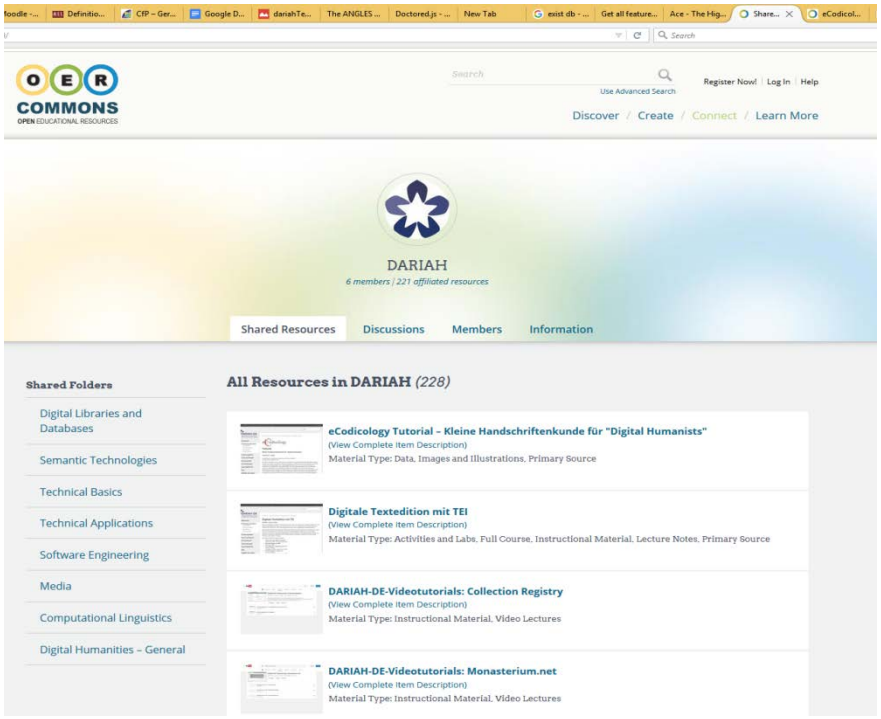

Fig. 5: Search results for "DARIAH" in OER commons, Open Educational Resources

The static, repository-like solutions are not really suitable for this project since a number of user requirements call for interaction an collaboration (e.g. user requirements 18, 20, 21).

4.2 Collaborative and interactive solutions

These solutions integrate material provision with collaborative and interactive parts. There are a wide range of tools with collaborative and interactive solutions. There are, for example, tools that support collaborative production processes for co-authoring texts like Google Docs or Wikis, or communication tools that provide for videoconferencing (e.g. skype or Google Hangouts), chats and forums.[7] Interactive assessment functionalities are also often integrated including multiple choice quizzes or feedback functionalities such as feedback questionnaires for students or like buttons.

Another option for delivering e-learning courses are learning management systems, such as Blackboard and Moodle. According to Paulsen (2002) a Learning Management System is a system that "organizes and provides access to online learning services for students, teachers, and administrators." These are software applications that provide access to learning content, facilitate the administration of the learning content in combination with collaborative and interactive functionalities like blogging, discussion forums, quizzes, integration of social media and facilitate the organization of user groups. Often also the term learning platform is used.

Furthermore, we can discern between dedicated LMS systems, designed specifically for e-learning and rather generic CMS systems that can be adapted for the specific needs of e-learning for example with special plugins. In this context, also MOOCs, Massive Open Online Courses, should not be mentioned. MOOCs are online courses which allow a large numbers of course participants. Examples for MOOCs are Coursera or *Stanford Online*.[8]

Originally, "LMS constitute the asynchronous part of e-learning technologies" (Jahn et al. 2012) but today they also might include functionalities for synchronous learning, like videoconferencing or chat functionalities or it is possible to integrate virtual meeting rooms into an LMS system.

In the survey of Paulsen (2003) with 113 institutions, and 32 institutions used self-developed LMS Systems. Self-developed LMS Systems can be for example a relational database as described by Deperlioglu et al. (2011). On the other hand 78 institutions out of 113 used commercial or already existing LMS systems (ibid). The survey showed that more institutions relayed on already existing solutions.

Most e-learning systems offer a similar basic set of features including:

- Assignment submission
- Discussion forum
- File upload/ download capacity
- Grading functionalities
- Instant messages

[7] Cf. List of Learning Tools
[8] For more examples see e.g. https://www.mooc-list.com

- Online calendar
- Online news and announcement (institution and course level)
- Assessment functionalities like quiz
- Wiki
- Plugins for social media[9]

In the next sections we briefly describe different solutions for e-learning platforms.

4.2.1 Dedicated learning management systems

Moodle[10] is a free and open-source software learning management system written in PHP. Moodle is designed to be responsive and accessible, and can be used on both desktop and mobile devices. The core functionalities of Moodle, apart from creating courses and managing students, include a personalized dashboard, an all-in-one calendar, file manager, text editor, notification functionalities for alerts on new assignments or deadlines, collaborative tools and activities such as assignments, chats, feedback surveys, forums, glossaries, quizzes, wikis. It is also possible to integrate external tools to allow participants to interact with LTI compliant learning resources and activities on other web sites.[11] There are also a wide range of plugins available[12] and it is also possible to program new plugins.

The edx platform[13], a free and open source course management system (CMS), is programmed in Python and consists mainly of two main components, the Open edX Studio for creating courses by the instructors and the Open edX Learning Management System for students accessing the course content.[14]

The ELMS Learning Network (ELMSLN)[15] is an open source modular educational technology platform for building and sustaining innovation in course technologies. ELMSLN is 100% open source and primary based on Drupal technology, a content management system.[16]

Blackboard Learn[17] (previously the Blackboard Learning Management System), is a virtual learning environment and course management system developed by Blackboard Inc offered as a commercial service. It is web-based server software which features course management, customizable open architecture, and scalable design that allows integration with student information systems and authentication protocols[18]. The main collaboration/ interaction and communication functionalities are: announcements, chat, discussions, and mail. The platform supports the creation of articles, assignments, posting of videos and other media types. There are

[9] http://elearningindustry.com/choosing-online-learning-platform-makes-sense
[10] www.moodle.org
[11] https://docs.moodle.org/29/en/Features
[12] https://moodle.org/plugins/
[13] https://github.com/edx/edx-platform
[14] Cf. https://open.edx.org/about-open-edx
[15] https://www.elmsln.org/
[16] Cf. https://www.elmsln.org/, cf. Collins, M. & Ollendyke, B. (2015). ELMS Learning Network: An EdTEch Platform for Sustaining Innovation. White Paper. Accessed 29.10.2015.
[17] http://www.blackboard.com/learning-management-system/blackboard-learn.aspx
[18] https://en.wikipedia.org/wiki/Blackboard_Learn

also grading functionalities available.

4.2.2 LMS plugins for CMS

There are not only complete LMSs on the market, there is also the possibility of using LMS plugins for content management systems in order to create e-learning platforms.

A content management system, often also called CMS, is a software that allows storing, editing and publishing content from a central interface. CMSs are often used to run websites, then they are often called web content management system. With a web content management system content can be created, managed and stored on websites. This content can be text and embedded graphics, photos, video, audio, and code that displays content or interacts with the user. Usually CMS systems have a front end and a back end. In order to create e-learning courses and deliver teaching materials, there are several learning management system plugins for content management systems.

In the following we will concentrate on

- Opigno for Drupal
- LearnPress for WordPress

Opigno[19] is an open source drupal based (Drupal 7) application to create e-learning platforms. **Drupal**[20] is a free open source software package based on PHP and MySQL that allows easy organizing, managing and publishing content on the web, with an endless variety of customization due to over 17 000 plugins. One of these plugins is Opigno. Opigno provides flexible building blocks and APIs to create a custom e-learning system. The created e-learning systems are SCORM and Tin Can compliant. Opigno is compatible with other drupal modules like collaborative and engagement tools. The core unit of Opigno is “the Course”, a flexible group entity that allows you to group users together. It is possible to add functionality to these courses like quizzes, certificates etc. [21]

LearnPress[22] is a free open source LMS based on WordPress to create online courses. **Wordpress**[23] is a free and open source content management system based on PHP and MySQL to create blogs and websites. There are a log of plugins to costumize the blog or website and one of the plugins is LearnPress. The core unit of LearnPress is the course and within the course it is possible to create lessions, or a quiz. The lesson content can be text, video or audio. The course can be created, administrated and also exported. It is also possible to integrate collaborative functions like Forum into LearnPress. LearnPress also provides some plugins, called add-ons and it is also possible to develop its own add-ons.

[19] https://github.com/devekko/opigno
[20] https://www.drupal.org/
[21] https://www.drupal.org/project/opigno
[22] https://github.com/LearnPress/LearnPress
[23] https://de.wordpress.com/

5 Technical implementation

Based on the overview in section 4, we can distinguish the following approaches for the implementation:

a. Programming everything from scratch b. Using existing Repository Systems c. Using existing Content Management Systems, ideally with specific plugins for e-learning d. Using existing Learning Management Systems e. Mixture of different approaches

Given the number of existing solutions and the limited resources within the project a. is a rather theoretical option. Given a number of requirements for collaborative and interactive features b. is also not considered further. Below we concentrate on c. (LearnPress) and d. (Moodle). Mixing different systems (e.) would again complicate the implementation and therefore a very valid reason is required to go to the trouble.

The implementation of the portal will be based on one of the existing technical solutions introduced in Section 4. The two principal options are the application of a general content management system (CMS) combined with specific plugins for e-learning or a more specialized learning management system (LMS).

Currently Moodle seems to cover most of the identified user requirement, so this is our primary target for the implementation. Moodle is a well-established LMS, created in 2002, has a well maintained documentation website and a plugin registry, with more than 1000 plugins to find the required plugins. Moodle is very flexible, and we can adapt the user interface, and customise most of the terminology, that is used in the menus and navigation bars.

LearnPress is a fairly new plugin for WordPress meant to support LMS functionality. However even though WordPress plugins can also be used, the support for extended functionality (the offer of specialized plugins) is limited. Further the available documentation is not satisfactory and it is not clear how well it is being adopted and how long it will be maintained (critical size of the community). On the other hand, having the static project page implemented in WordPress, it would simplify the integration to have the e-learning platform based on the same system. Considering all of these factors Moodle is the primary candidate for the implementation, with LearnPress as a fallback. To allow for early prototyping (trying out in practice if the solutions meets the needs), ACDH set up a test instance of Moodle. You can see a screenshot of the test instance in Fig. 6 overleaf.

5.1 Multilinguality

One important feature of the developed platform is multilinguality, both in the user interface and in support for multi-lingual content.

Regarding the user interface, both Moodle and LearnPress are available in multiple language:

• LearnPress is available in 8 languages, English, French, Indonesian, Italian, German, Polish, Russian, and Dutch. WordPress is released in 157 languages or varieties

• Moodle is available in more than 100 languages.

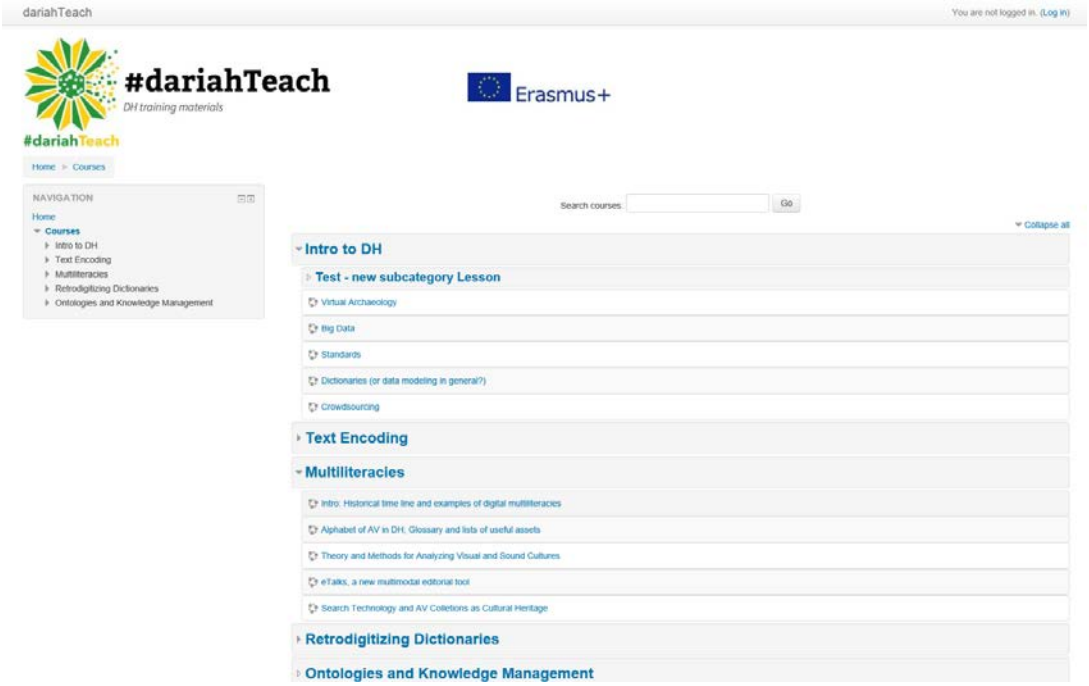

Fig. 6: Screenshot of test instance of Moodle with the 5 basic modules

The multilingual content needs to be delivered by the providers of the materials. The platform only needs to be "language-aware", i.e. it must allow the indication of the language of the material and ideally also offer a means to keep different language version of one learning material together.

It needs to be investigated further to which extent the platform can support the workflow for translation. However given the variety of materials and the many different approaches, it is not feasible, nor useful to try to integrate some machine translation functionalities into the platform.

In this respect, it will be nevertheless worthwhile to monitor the new TraMOOC project[24], and evaluate the possible synergies with the dariahTeach project.

5.2 Custom metadata

An important requirement is to be able to describe the learning material according to the LOM metadata format. It turns out that the existing LMS are rather rigid in this respect, offering only a fixed set of fields for describing the entities. We found moodle plugins for custom fields, however.

A promising approach seems to be the feature for publishing courses[25] to a community hub (e.g. http://community.moodle.de/, or mooch, http://hub.moodle.org/), where a structured description is

[24] http://tramooc.eu/, Translation for Massive Open Online Courses http://www.eamt2015.org/files/downloads/EAMT2015_Proceedings.pdf, p. 217

[25] https://docs.moodle.org/30/en/Publishing_a_course#Updating_a_course_backup

required for every course to be exported. The foreseen fields cover quite well a basic set of fields in LOM. The disadvantage is, that these are only stored on the side of the community hub and only for courses that are pushed to the community hub. On the other hand the functionality of the community hub to share and exchange courses and learning material is a requirement within the project, so this is a feature that needs to be adopted. It needs further investigation, in how far it can cover the requirement for custom metadata.

5.3 Web-based XML-Editor

There are a lot of open-source and commercial online XML-Editors, however for the scope of our project we looked at existing open source web-based solutions. The requirement being web-based, desktop solutions like Oxygen are not an option.

This is a list of a few existing web-based XML-Editors (Some of them are also installed as demos on ACDH test-servers):

- **ANGLES online XML editor**[26]

  Demo on ACDH server available[27].

  The editor provides syntax highlighting, schema-based tag-suggestion, validation with verbose error description.

- **Doctored.js Editor**[28]

  Demo on ACDH server available[29].
  It tries to be user-friendly by not showing the tags, but the nested boxes solutions do not necessarily aid the comprehension.

- **eXide**[30],

  Demo on ACDH server available[31].
  eXide is tightly bound to eXist database. We have not used it so far as a separate component.

- **CodePlex**[32]
- **Turtelbite XML Editor**[33]
  Turtelbite XML Editor is a web-service, where you have to create an account. It is user friendly because it uses no braces and it is tree view based.
- **ACE**[34]

[26] http://mith.us/angles/
[27] https://minerva.arz.oeaw.ac.at/angles/demo/
[28] http://holloway.github.io/doctored/
[29] https://minerva.arz.oeaw.ac.at/doctored/
[30] https://github.com/wolfgangmm/eXide
[31] https://minerva.arz.oeaw.ac.at/exist/apps/eXide/index.html
[32] http://xmlwebpad.codeplex.com/
[33] http://xml-editor.kaegi.net/get-it.php
[34] https://ace.c9.io/#nav=about

ACE is an embeddable code editor written in JavaScript.

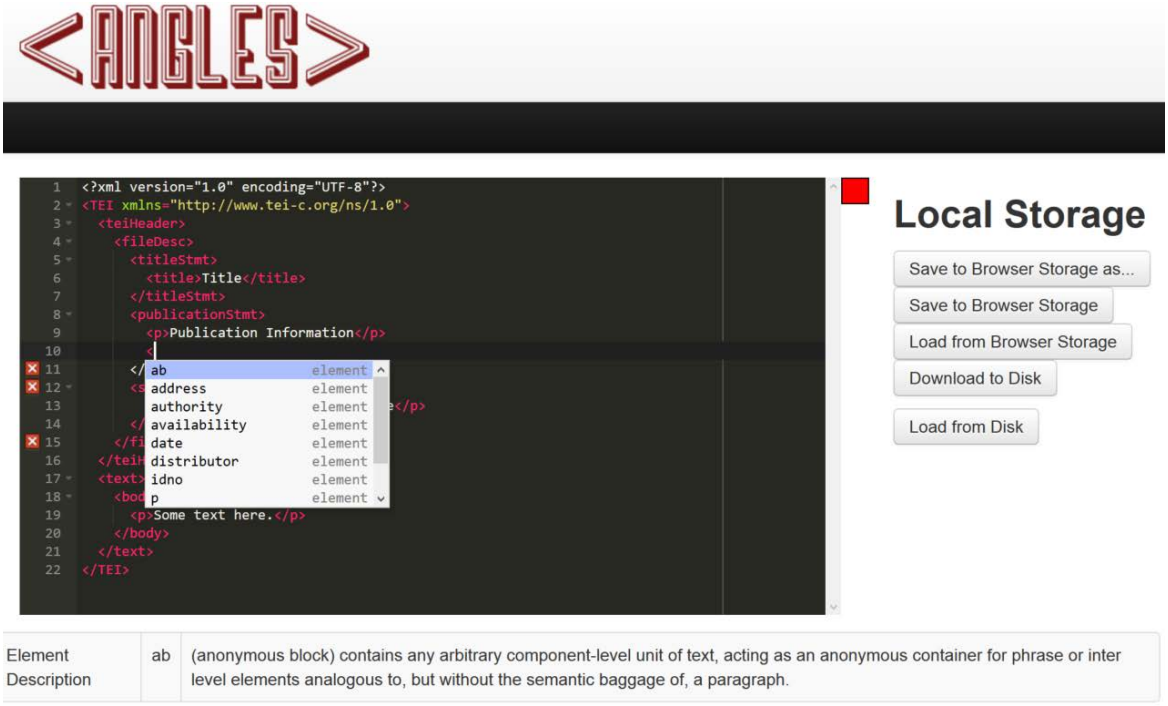

Fig. 8: Screenshot of the doctored.js editor

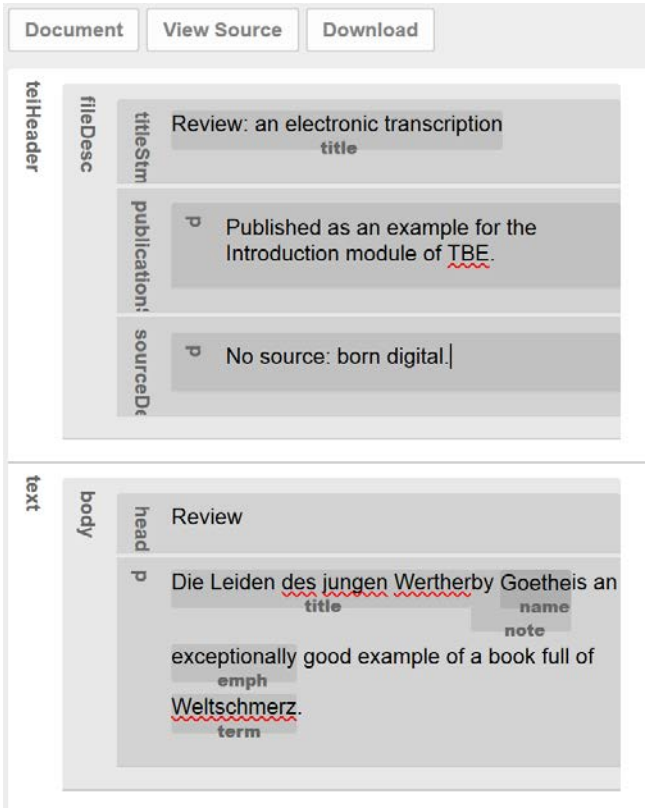

Fig. 1: Screenshot of the doctored.js editor

6 References

Deperlioglu, O. / Sarpkaya Y. / Ergun E. (2011). Development of a relational database for learning management system. TOJET - *The Turkish Online Journal for Educational Technology* October 2011, Volume 10 Issue 4, pp. 107-120. Available at: http://www.tojet.net/articles/v10i4/10411.pdf

Kurilovas, E. (2009). Methods of Multiple Criteria Evaluation of the Quality of Learning Management Systems for Personalised Learners Needs. *European Conference on Technology-Enhanced Learning* (EC-TEL'2009), 1-10, Retrieved from: http://wwwis.win.tue.nl/lms-ale-09/Kurilovas_paper.pdf

Kurilovas, E. / Dagiene V. (2010). Learning Objects and Virtual Learning Environments Technical Evaluation Criteria. *E-Journal of E-Learning* – EJEL, Volume 7, Issue 2.

Paulsen, M. F. (2003). Experiences with Learning Managements Systems in 113 European Institutions. Educational Technology & Society, 6 (4), 134-148. Available at: http://www.ifets.info/journals/6_4/13.pdf

Paulsen, M. F. (2002). Online Education: Discussion and Definition of Terms. In Paulsen, M. F.; Keegan, D.; Dias, A.; Dias, P.; Pimenta, P.; Fritsch, H.; Föllmer, H.; Micincova, M. & Olsen, G. Web- Education Systems in Europe, Hagen: Zentrales Institut für Fernstudienforschung, Fern Universität, 23-28. Preliminary version available here: https://www.porto.ucp.pt/open/curso/modulos/doc/Definition%20of%20Terms.pdf

Roy, D. / Sarkar, S. / Ghose S. (2010). A Comparative Study of Learning Object Metadata. Learning Material Repositories, Metadata Annotation & an Automatic Metadata Annotation Tool. In: Joshi / Boley / Akerkar (eds.), (2010). *Advances in Semantic Computing*. Vol 2. pp 103 - 126.

Jahn, M. / Piesche C. / Jablonski, S. (2012). Flexibility requirements concerning the design of synchronous e-learning systems. *Interactive Technology and Smart Education*. Vol. 9 No. 4, pp. 233- 345.

18