CrowdLearn: Collaborative Engineering of (semi-)Structured Learning Objects

Darya Tarasowa, Ali Khalili, Sören Auer

Abstract

A special activity, that became possible nowadays, is the *collaborative e-learning and collaborative engineering of learning materials.* Such an opportunity is available in educational systems, based on the *wiki* paradigm. However, Ward Cunningham's wiki paradigm was mainly *only* applied to unstructured, textual content thus limiting the content structuring, repurposing and reuse. Some steps in this direction are made by *Semantic Web* technologies and their combination with the wiki paradigm. In many potential usage scenarios, however, the *(semi-)structured* educational content (e.g. presentations, questionnaires, diagrams, etc.) should be managed and the collaboration of large user communities around such content should be effectively facilitated. We present the CrowdLearn concept for collaborative engineering of the semantically structured learning objects as well as a model-driven approach to generate the wiki-like application operating in the spirit of CrowdLearn. We implement and evaluate the approach with Slidewiki - an educational platform mainly dealing with slide presentations. The article also comprises a usability evaluation with real students.

1 Introduction

Changing reality poses new conditions and requirements for e-learning platforms. As it is mentioned in [13], "*the amount of knowledge that we deal with is much bigger than before, the interrelations between different forms of information are much more complex*". Due to this, new approaches for learning process organisation have to be created. Thus, a model in which a teacher is the monopolising agent and the authorised representative of knowledge is no more adequate. Learners should be allowed to have their own knowledge and share it with teachers, in other words teachers and learners could switch their roles during the learning process.

A special activity, that became possible, is the *collaborative e-learning and collaborative engineering of learning materials.* Such an opportunity is available in educational systems, based on the *wiki* [9] paradigm. Since its inception in the early 2000s, wiki technology became an ubiquitous pillar for enabling large-scale collaboration. Wiki approach enabled the creation of the largest encyclopedia of human-mankind edited by tens of thousands of volunteer editors – Wikipedia. Another successful example of large-scale collaboration around textual educational content is the *Wikibooks.org* resource. However, Ward Cunningham's wiki paradigm was mainly *only* applied to unstructured, textual content thus limiting the content structuring, re-purposing and reuse. Some steps in this direction are made by *Semantic Web* technologies and their combination with the wiki paradigm. Two kinds of semantic wikis exist: semantic text wikis, such as Semantic MediaWiki [7] are based on semantic annotations of the textual content; semantic data wikis, such as OntoWiki [6], [2] are directly based on the RDF data model. In many potential usage scenarios, however, the content to be managed by a wiki is neither purely textual *nor* fully semantic. Often *(semi-)structured* content (e.g. presentations, questionnaires, diagrams, etc.) should be managed and the collaboration of large user communities around such content should be effectively facilitated.

However, a proper community collaboration, authoring, versioning, branching, reuse and re-purposing of (semi-)structured educational content similarly as we know it from the open-source software community is currently not supported. To address the issue we develop the CrowdLearn concept. We determine it as the concept that exploits the wisdom, creativity and productivity of the crowd for the creation of rich, deep-semantically structured E-Learning content. In the paper we present the concept, as well as a conceptual data model for its implementation. The paper also comprises an important implementation issues, that we faced when developed an example CrowdLearn application.

2 CrowdLearn Concept

To develop the concept, we have collected requirements from the research literature in the educational domain. Analysing them, we produced an own view of how to improve the learning quality. In the section we present the base concepts, that guided us within the development process.

Active learning It is a fact, that when you try to teach, your first student is yourself. In other words, by explaining the material to others you understand all the details more clear. This fact could be taken as a paradigm for our approach, as well as an active learning itself. Users can play a *teacher* role for one domain and a *student* role for another. Even being "students", users are able to make suggestions, fix errors, discuss issues equally to the content owners ("teachers"). This concept allows users to feel responsibility, the learning process becomes more encouraged. Even answering the questionnaire, user has a possibility to make an impact by analysing the quality of the questions and making suggestions of how to improve them. Thus, the knowledge is being created not only explicitly by contributors, but also implicitly through discussions, answering the questionnaires, in other words through native learning activities.

Collaboration and social networking We see the CrowdLearn concept as an application of crowdsourcing techniques to the e-learning content engineering. Thus, semantically structured learning objects can be edited in a truly collaborative way. For enabling the high-level collaboration, we propose to put the content under the version control, similar with the wiki paradigm. Another important condition is the support of social networking activities. Both content owners and students are able to participate in discussions about correctness of the material. Following the individual learning objects allows users to receive messages about all the changes; watching the individual authors facilitates organising user groups for different domains.

Education for all *"Imagine a world in which every single person on the planet is given free access to the sum of all human knowledge. That's what we're doing. And we need your help"*. This citation belongs to Jimmy Wales, one of the Wikipedea founders. We aim to apply this concept to (semi-)structured content, that is why we use wiki-paradigm of content creation and publishing. All the content in the system should be under "CC-BY" license, as we want all the contributors' names been visible on each small piece of the content. Thus, all the content can be used in any activities (edit, download and change locally, present for any other people outside the platform etc.) until it includes these names.

(Re-)structure,reuse and re-purpose Instead of dealing with large learning objects (often whole presentations or questionnaires), we decompose them into fine-grained *learning artifacts*. Thus, instead of a large presentation a user will be able to edit, discuss and reuse individual slides; instead of a whole questionnaire she will be able to work on the level of individual questions. This concept efficiently facilitates the possibility of reuse and re-purpose the learning objects. To implement the concept, we use the WikiApp data model, briefly described in Section 3.1. The detailed overview of the model is presented in.

3 CrowdLearn Implementation

We implement and evaluate the concept with Slidewiki - web-based crowdlearning platform. Slidewiki deals with two types of (semi-)structured learning objects: slide presentations and questionnaires. The (semi-)structured organisation of the material significantly increases the possibility of reuse and repurposing, facilitates collaborative activities and allows to apply version control. To implement these possibilities we used an instance of the WikiApp data model, which natively supports versioning and structuring of the different content objects. Also, the WikiApp data model allowed us to use model-driven generation approach.

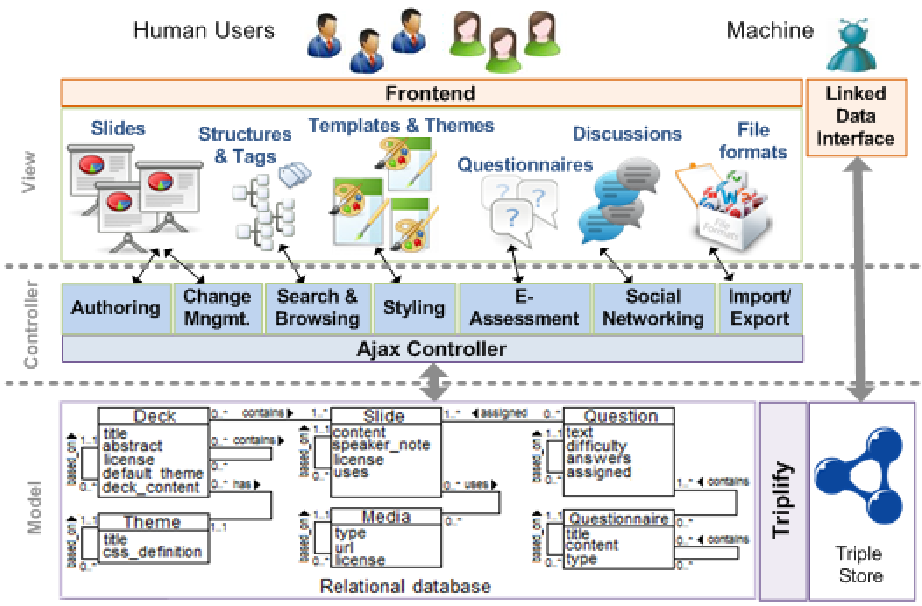

Figure 1: Bird's eye view on the SlideWiki MVC architecture.

3.1 Data Model

The WikiApp data model is a refinement of the traditional entity-relationship data model. It adds some additional formalisms in order to make users as well as ownership, part-of and based-on relationships first-class citizens of the data model. The conceptual view of the model is presented in Figure 2

The WikiApp data model assumes that all content objects are versioned using the *timestamp* and the *base-content-object* relation. In the spirit of the wiki paradigm, there is no deletion or updating of existing, versioned content objects. Instead new revisions of the content objects are created and linked to their base objects via the *base-content-object* relation. All operations have to be performed by a specific user and the newly created content objects will have this user being associated as their owner. *Watching* the users, as well as *following* the learning objects operations are natively supported by the model. This allows users to receive the information about changes of the followed content object or new objects created by the watched user. Also, these operations allow to easily find the followed object or user.

Our SlideWiki example application uses two implementations of WikiApp data model. The first implementation is used for managing slides and presentations. It includes individual slides (consisting mainly of HTML snippets, SVG images and meta-data), decks (being ordered sequences of slides and sub-decks), themes (which are associated as default styles with decks and users) and media assets (which are used within slides). The second implementation was developed for managing questions and questionnaires (tests). It includes questions

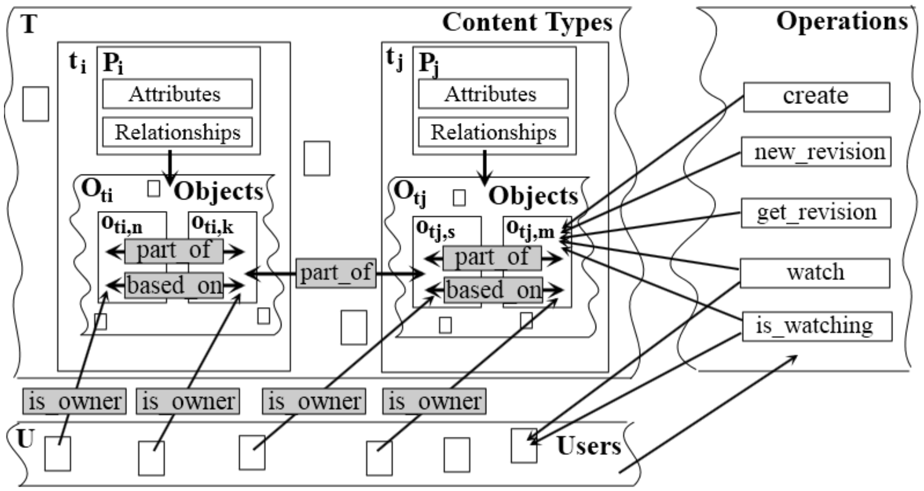

Figure 2: Conceptual view of the WikiApp data model.

for the slide material (the question is assigned to all slide revisions), questionnaires (which could be organized manually by user or created automatically in accordance with the deck content), and answers (which are the part of the questions).

We implicitly connected these two WikiApp instances by adding two relations. Firstly, we assigned questions to slides. Thus, during the learning process users are able to try answer the questionnaires and have a look the assigned slide if necessary. The important issue here is that we assign question not to individual slide revision, but for the slide itself. This decision gives an opportunity to create new slide revision, that already has a list of questions, collected from other revisions. Secondly, we assigned questionnaires to concrete deck revisions. Thus automatically created test saves the structure of deck revision, which it is assigned to. This allows us to use module-based assessment to score the test results. Figure 3 shows relations between objects in SlideWiki. These two special relations are indicated on the diagram by the dashed arrows.

3.2 Architecture

The SlideWiki application makes extensive use of the model-view-controller (MVC) architecture pattern. The MVC architecture enables the decoupling of the user interface, program logic and database controllers and thus allows developers to maintain each of these components separately. As shown in Figure 1, the implementation comprises the main components authoring, change management, import/export, frontend, social networking, linked data interface, search, e-learning and styling. We briefly walk-through these components in the sequel.

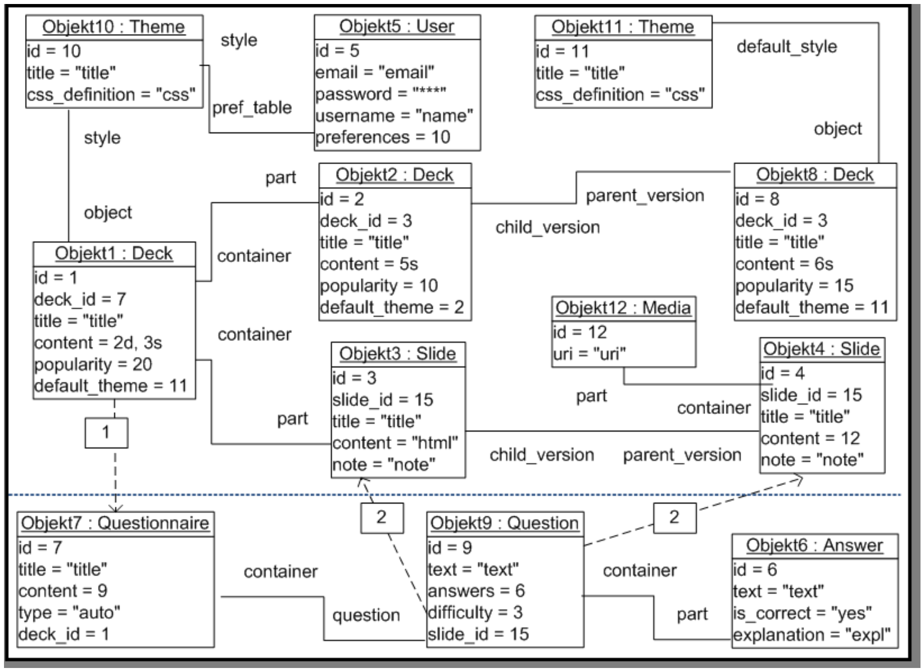

Figure 3: Example object diagram for Slidewiki. ***1*** *- connection between questionnaires and deck revisions to enable module-based scoring;* ***2*** *- connection between questions and slides to allow users see the related material when answering the questionnaires*

Authoring. SlideWiki employs an inline HTML5 based WYSIWYG (What-You-See-Is-What-You-Get) text editor for authoring the presentation slides. Using this approach, users will see the slideshow output at the same time as they are authoring their slides. The editor is implemented based on ALOHA editor[1] extended with some additional features such as image manager, source manager, equation editor. The inline editor uses SVG images for drawing shapes on slide canvas. Editing SVG images is supported by SVG-edit[2] with some predefined shapes which are commonly used in presentations. For logical structuring of presentations, SlideWiki utilizes a tree structures in which users can append new or existing slides/decks and drag & drop items for positioning. When creating presentation decks, users can assign appropriate tags as well as footer text, default theme/transition, abstract and additional meta-data to the deck.

Change management. There are different circumstances in SlideWiki for which new slide or deck revisions have to be created. For decks, however, the situation is slightly more complicated, since we wanted to avoid an uncontrolled

[1] `http://aloha-editor.org/`
[2] http://code.google.com/p/svg-edit/

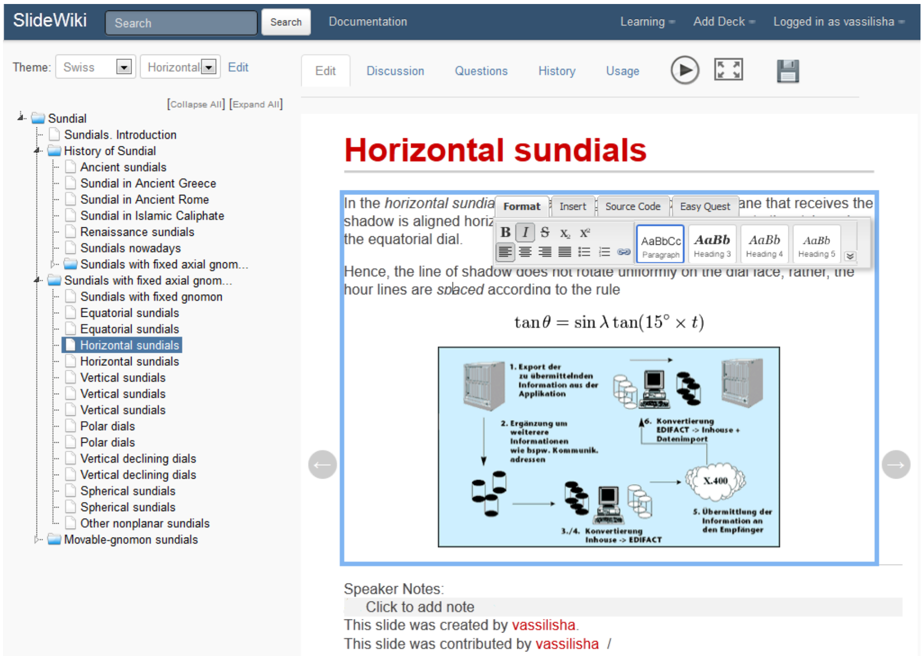

Figure 4: The screenshot of the SlideWiki application: Inline authoring of presentations

proliferation of deck revisions. This would, however, happen due to the fact, that every change of a slide would also trigger the creation of a new deck revision for all the decks the slide is a part of. Hence, we follow a more retentive strategy. We identified three situations which have to cause the creation of new revisions:

- The user specifically requests to create a new deck revision.
- The content of a deck is modified (e.g. slide order is changed, change in slides content, adding or deleting slides to/from the deck, replacing a deck content with new content, etc.) by a user which is neither the owner of a deck nor a member of the deck's editor group.
- The content of a deck is modified by the owner of a deck but the deck is used somewhere else.

In addition, when creating a new deck revision, we always need to recursively spread the change into the parent decks and create new revisions for them if necessary.

E-Assessment. SlideWiki supports the creation of questionnaires and self-assessment tests based on slide material. Each question has to be assigned with at least one slide. Important note here, that the question is assigned not to the slide revision, but to slide itself. Thus, when a new slide revision appears, it continues to include all the list of previously assigned questions. Questions can be combined into questionnaires. The *automatically created* questionnaires include the last question revisions from all the slides within the current deck revision. Manually created questionnaires present a collection of chosen questions and currently can not be manipulated as objects. Thus, in our implementation only questions and answers have to be placed under the version control. However, their structure is trivial and the logic of creating their new revisions is intuitive. We just restricted the number of new revisions to be created similarly with the decks: changes made by the question owner do not trigger a new revision creation.

Linked Data Interface. SlideWiki implementations can be easily equipped with a Linked Data interface. We employed the RDB2RDF mapping tool Triplify [1] to map SlideWiki content to RDF and publish the resulting data on the Data Web. Triplify is based on mapping HTTP-URI requests onto relational database queries. It transforms the resulting relations into RDF statements and publishes the data on the Web in various RDF serializations, in particular as Linked Data. Triplify neither defines nor requires to use a new mapping language, but exploits and extends certain SQL notions with suitable conventions for transforming database query results (or views) into RDF and Linked Data. The Triplify configuration for SlideWiki was created manually. The SlideWiki Triplify Linked Data interface is available via: `http://slidewiki.aksw.org/triplify`.

4 Evaluation

The SlideWiki concept was evaluated in several ways: Firstly, as a proof-of-concept we developed a comprehensive implementation, which is available at: `http://slidewiki.aksw.org`. The SlideWiki platform is currently used for accompanying an information systems lecture with more than 80 students. We performed a preliminary usability study as well as a performance evaluation, which are described in the sequel.

Synthetic benchmarking. To measure the system performance, we synthesized three datasets containing 100, 500 and 1000 presentations in .pptx format, with an average of 33 slides in each presentation. The presentations were obtained from the two websites: 2shared.com (900 presentations) and slideshare.com (100 presentations). We randomly chose presentations with a file size between 3MB and 20MB, in order to exclude empty and excessively large presentations. A notebook computer with Intel Core 2 Duo 2,66GHz CPU; 4GB RAM and Windows 7 64-bit was used for all the measurements. The results of the benchmarking are summarized in the Figure 5. One of the advantages of a relational database in comparison with a triple-store is better performance and scalability. To demonstrate this, we imported an RDF dump

Operation	SlideWiki			OntoWiki		
Data (decks)	**100**	**500**	**1000**	**100**	**500**	**1000**
(slides)	2,905	16,459	32,629	2,905	16,459	32,629
Import (average per slide)						
processing	771	934	951	-	-	-
database	1	2	3	-	-	-
total	772	936	954	-	-	-
Export (average per slide)						
total	54	156	209	-	-	-
Search for decks (average)						
database	6	8	10	5	7	12
processing	36	37	45	36	37	45
total	42	45	55	41	44	57
Search for slides (average)						
database	7	17	49	59	329	642
processing	118	238	376	118	238	376
total	125	255	425	177	667	1108
Move (average)						
deck	4	4	5	1	2	3
slide	3	4	5	3	4	5
Show deck content (average per slide)						
deck	10	35	59	12	40	81
Create new revision (average)						
deck	14	17	21	2	3	4
slide	1	1	2	3	5	7

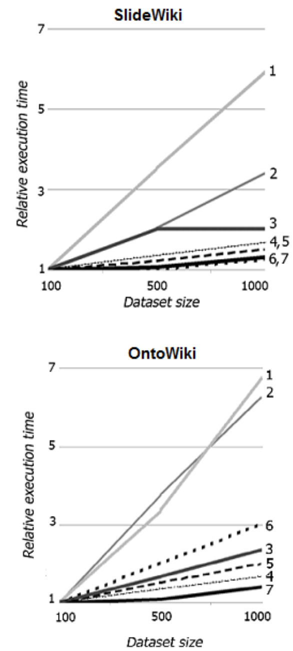

Figure 5: Benchmarking results: (left) Full comparison (right) Execution time relatively to 100-deck dataset. Legend: 1-*show deck*, 2-*search slides*, 3-*new slide revision*, 4-*move slide*, 5-*new deck revision*, 6-*move deck*, 7-*search deck*.

produced by Triplify, into OntoWiki [6] (with Virtuoso triple-store backend), and repeated the measurements with corresponding SPARQL queries. To compare the scalability of SlideWiki, the average execution time for each operation in the relational and RDF implementations was plotted relatively to the 100-presentations dataset in Figure 5. Based on the measurement results, we can conclude, that SlideWiki performance better in all operations, other than *move deck* and *new deck revision*. SlideWiki also scales significantly better, despite the fact, that we did not yet invest much effort in optimizing the performance. In particular, we will further optimize the performance of the least scaling operations *show deck* and *search slide*.

Usability Evaluation. In order to determine whether we succeeded to effectively hide SlideWiki's data model complexity, we performed a usability user study with 13 subjects. Subjects were drawn from the members of AKSW research group, the computer science department at the university of Leipzig. We first showed them a tutorial video of using different features of SlideWiki then asked each one to create a presentation with SlideWiki. After finishing the task, we asked the participants to fill out a questionnaire which consisted of three parts: demographic questions, feature usage questions and usability experience questions. We used the *System Usability Scale* (SUS) [10] to grade the usability of SlideWiki. SUS is a standardized, simple, ten-item Likert scale-based questionnaire[3] giving a global view of subjective assessments of usability. It yields a single number in the range of 0 to 100 which represents a composite measure of the overall usability of the system. The results of our survey showed a mean usability score of 69.62 for SlideWiki which indicates a reasonable level of usability. Of course, this is a very simplified view on usability and we expect even better results could be achieved by putting more effort into the SlideWiki development (the development of SlideWiki only consumed 5 man months). However, our goal was to demonstrate that SlideWiki implementations with good usability characteristics can be created with relatively limited effort. In addition to quantitative results, we also collected a number of user suggestions. For instance some users suggested improving the WYSIWYG editor for adding predefined shapes, providing autosave feature, supporting more import/export formats, defining user groups etc.

5 Related work

Collaborative creation of e-learning content. The importance of creating reusable and re-purposable e-learning objects is meanwhile widely accepted by the e-learning community [4]. However, it seems that most of the works addresses the learning object reuse problem rather by means of semantic meta-data annotations, content tagging and packaging than by creating richly structured, reusable learning objects from the ground. The importance of creating learning objects already with reuse in mind was, for example, stated by [13]: '*Content ... should be represented not as an object of study but rather as necessary elements towards a series of objectives that will be discovered in the course of various tests*'. There are only few approaches for the direct authoring of reusable content, such as, for example, learning examples creation [8] or semantic structuring and annotation of video fragments [3]. In this proposal we aimed to create an educational content *creation* platform grounded on reusable content authoring and large-scale community collaboration (or crowd-sourcing).

Wiki-based collaborative knowledge engineering. The importance of wikis for collaborative knowledge engineering is meanwhile widely acknowledged. In [14], for example, a knowledge engineering approach which offers wiki-style collaboration, is introduced aiming to facilitate the capture of knowledge-in-action which spans both explicit and tacit knowledge types. The approach extends a combined rule and case-based knowledge acquisition technique known as *Multiple Classification Ripple Down Rules* to allow multiple users to collaboratively view, define and refine a knowledge base over time and space. In a more applied context, [5] introduces the concept of wiki templates that allow end-users to define the structure and appearance of a wiki page in order to

[3] www.usabilitynet.org/trump/documents/Suschapt.doc

facilitate the authoring of structured wiki pages. Similarly the Hybrid wiki approach [11] aims to solve the problem of using (semi-)structured data in wikis by means of page attributes. We will apply the wiki paradigm and selected wiki-based knowledge engineering approaches for the creation and collaboration around e-learning artifacts and learning objects.

6 Conclusions and further work

Our approach addresses weaknesses of conventional text-oriented as well as semantic wikis. Compared to the text-oriented wikis, CrowdLearn concept deals with a structured content, resulting in substantially increased reuse, repurposing and collaboration capabilities. With regard to semantic wikis, CrowdLearn approach is more *user friendly* (since it can be tightly adapted to the respective domain) and *scalable* (since it can be used in conjunction with relational databases).

One of directions for the further work is a support of multi-linguality. Since all content is versioned and semantically structured, it is easier to (automatically) translate content and to keep track of changes in various multi-lingual versions of the same content object by comparing and merging the versions.

Learners come from different environments, have different ages and educational backgrounds. That makes their integration into one single group more complicated or even impossible. That is why, new approaches should give the possibility to *personalize* the learning process. That means, that modern e-learning platforms should present only the information that is really relevant for the learner, in an appropriate manner, and at the appropriate time [12]. Thus, our second direction is providing the personalised content based on initial user assessments.

References

[1] Sören Auer, Sebastian Dietzold, Jens Lehmann, Sebastian Hellmann, and David Aumueller. Triplify: Light-weight linked data publication from relational databases. In *WWW2009, Spain*. ACM, 2009.

[2] Sören Auer, Sebestian Dietzold, and Thomas Riechert. Ontowiki - a tool for social, semantic collaboration. In *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference*, pages 736 – 749. Springer, 2006.

[3] Elena García Barriocanal, Miguel-Ángel Sicilia, Salvador Sánchez Alonso, and Miltiadis D. Lytras. Semantic annotation of video fragments as learning objects. *Interactive Learning Environments*, 19(1):25–44, 2011.

[4] Vladan Devedzic. *Semantic Web and Education (Integrated Series in Information Systems)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[5] Anja Haake, Stephan Lukosch, and Till Schümmer. Wiki-templates: adding structure support to wikis on demand. In *Proceedings of the 2005 International Symposium on Wikis*, pages 41–51. ACM, 2005.

[6] Norman Heino, Sebastian Dietzold, Michael Martin, and Sören Auer. Developing semantic web applications with the ontowiki framework. In *Networked Knowledge - Networked Media*, volume 221 of *Studies in Computational Intelligence*, pages 61–77. Springer, Berlin / Heidelberg, 2009.

[7] Markus Krötzsch, Denny Vrandečić, Max Völkel, Heiko Haller, and Rudi Studer. Semantic Wikipedia. *Journal of Web Semantics*, 5(4):251–261, 2007.

[8] Yen-Hung Kuo, Qing Tan Kinshuk, Yueh-Min Huang, Tzu-Chien Liu, and Maiga Chang. Collaborative creation of authentic examples with location for u-learning. In *IADIS International Conference e-Learning 2008*, pages 16 – 20. IADIS, 2008.

[9] Bo Leuf and Ward Cunningham. *The Wiki way: quick collaboration on the Web.* Addison-Wesley, London, 2001.

[10] James Lewis and Jeff Sauro. The Factor Structure of the System Usability Scale. In *Human Centered Design*, volume 5619 of *LNCS*, chapter 12, pages 94–103. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2009.

[11] Florian Matthes, Christian Neubert, and Alexander Steinhoff. Hybrid wikis: Empowering users to collaboratively structure information. In *ICSOFT (1)*, pages 250–259, 2011.

[12] Robin D. Morris. Web 3.0: Implications for online learning. *Techtrends*, 55(1):42–46, 2006.

[13] Nieves Pedreira, José Ramón Méndez Salgueiro, and Manuel Martínez Carballo. E-learning in new technologies. In *Encyclopedia of Artificial Intelligence*, pages 532–535. IGI Global, 2009.

[14] Debbie Richards. A social software/web 2.0 approach to collaborative knowledge engineering. *Inf. Sci.*, 179(15):2515–2523, 2009.