

CrowdLearn: Crowd-sourcing the Creation of Highly-structured e-Learning Content

Darya Tarasowa, Ali Khalili, Sören Auer and Jörg Unbehauen
AKSW Research Group, Institute of Computer Science, University of Leipzig,
Postfach 100920, 04009 Leipzig, Germany

Keywords: e-Learning, LMS, Crowdsourcing, (semi-)Structured Learning Objects, SCORM.

Abstract: While nowadays there is a plethora of Learning Content Management Systems, the collaborative, community-based creation of rich e-learning content is still not sufficiently well supported. Few attempts have been made to apply crowd-sourcing and wiki-approaches for the creation of e-learning content. However, the paradigm is *only* applied to unstructured, textual content and cannot be used in SCORM-compliant systems. To address this issue we developed the CrowdLearn concept to exploit the wisdom, creativity and productivity of the crowd for the creation of rich, deep-semantically structured e-learning content. The CrowdLearn concept combines the wiki style for collaborative content authoring with SCORM requirements for re-usability. Therefore, it enables splitting the learning material into Learning Objects (LOs) with an *adjustable* level of granularity. In order to realize the CrowdLearn concept, a novel data model called *WikiApp* is devised. The WikiApp data model is a refinement of the traditional entity-relationship data model with further emphasis on collaborative social activities and structured content authoring. We implement and evaluate the CrowdLearn approach with SlideWiki – an educational platform dealing with presentations and assessment tests. The article also comprises results of a usability evaluation with real students.

1 INTRODUCTION

While nowadays there is a plethora of Learning Content Management Systems (LCMS), the collaborative, community-based creation of rich e-learning content is still not sufficiently well supported. Few attempts have been made to apply crowd-sourcing and wiki-approaches for the creation of e-learning content. *Wikiversity*¹, for example, is a Wikimedia Foundation project aiming to leverage standard wiki technology for the creation of hypertext e-learning content. *Peer 2 Peer University (P2PU)*² and *PlanetMath*³ are other examples which employ crowd-sourcing to create rich e-learning content. P2PU helps users to navigate the wealth of open education materials and supports the design and facilitation of courses. The PlanetMath is a project to aiming to become a central repository for mathematical knowledge on the web, with a pedagogical mission. However, we deem, that no real attempt has been made so far to truly apply the concepts behind wikis and crowd-sourcing to

develop a specifically tailored technology supporting the creation of highly-structured, SCORM-compliant e-learning content.

Sharable Content Object Reference Model (SCORM) (sco, 2011a) as one of the community-approved standards, requires the transformation of the learning material into the sequence of annotated *Sharable Content Objects* (SCOs). The granularity and sequencing of the SCOs should be determined by the content author depending on the audience needs and preferences (sco, 2011b). Ward Cunningham's *wiki* (Leuf and Cunningham, 2001) paradigm is mainly *only* applied to unstructured, textual content. This limitation makes it difficult or even impossible to use the wiki style of content authoring in the SCORM-compliant learning platforms. As a result, 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.

CrowdLearn exploits the wisdom, creativity and productivity of the crowd for the creation of rich,

¹<http://wikiversity.org/>

²<http://p2pu.org/>

³<http://planetmath.org/>

deep-semantically structured e-learning content. It combines the wiki style of collaborative content authoring with SCORM requirements for re-usability. Therefore, it enables splitting the learning material into *Learning Objects* (LOs) with an *adjustable* level of granularity. The CrowdLearn concept is based on five fundamental components (cf. 3): standard compliance, semantic structuring, enhanced possibilities for reuse and re-purpose, crowd-sourcing and social networking.

In order to implement and evaluate the CrowdLearn concept, we created a showcase application named *SlideWiki*. SlideWiki deals with the collaborative creation of original e-learning content such as presentations, slides, diagrams and self-assessment tests. During the implementation of the CrowdLearn concept, we faced several challenges (cf. 4). For enabling the high-level collaboration, all content should be versioned, similar to the wiki paradigm. SCORM has direct support for multiple-version content objects. However, we needed rules for triggering the creation of new revisions. Our findings on this issue are presented in 4.2. The next challenge was the SCORM requirement for the content to be structured. To solve this we developed the WikiApp data model (see 4.1) that organizes the relations between different content objects. Finally, the third challenge was to involve the learners in the process of content creation. We addressed this issue by providing support for social networking activities. Both content owners and students are able to participate in discussions about the learning material. While SCORM allows content engineers to do the sequencing, we allow it to be done by learners as well. As a consequence, semantically structured learning objects can be created and edited in a truly collaborative way. We further evaluated the CrowdLearn concept by conducting a comprehensive usability study with real students using SlideWiki (cf. 5).

2 RELATED WORK

Related work can be roughly divided into the following two categories:

Collaborative Creation of e-Learning Content. The importance of creating reusable and re-purposable e-learning objects is widely accepted by the e-learning community (Devedzic, 2006). However, most of the works address 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 (Pedreira et al., 2009): *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 (Kuo et al., 2008) or semantic structuring and annotation of video fragments (Barriocanal et al., 2011).

We should also mention the *Learning Objects Repositories* (LORs), that allow to produce structured reusable content. The first of them, *Connexions* (<http://cnx.org/>), presents the learning material as a combination of paragraphs, each of them could be easily edited or deleted. However, this structuring is done more for comfortable editing and does not have any functional benefits: the paragraphs cannot be reused or annotated independently. Thus, Connexions presents just an improved user interface for wiki-based system. The second example of structuring, that is more close to our idea, is *LeMill*⁴. LeMill provides a way of collaborative editing of the presentations by implementing presentations as a group of images which can be edited collaboratively. However, to edit a slide, a user has to replace it with another one. Also, it is impossible to have several subgroups of the slides within a presentation. The search through the slides (not presentations) is also not implemented. Thus, slides cannot be manipulated as independent learning objects.

The CrowdLearn concept differs from the existing approaches for managing e-learning content. It enables the construction of *semantically structured learning objects* from existing sources by combining, reordering and simple editing. By the term *semantically structured object* we mean that all the parts within the structure own all the attributes and methods of the object type, or, in the case of learning content, are complete and fully-functional LOs.

Wiki-based Collaborative Knowledge Engineering. The importance of wikis for collaborative knowledge engineering is widely acknowledged. In (Richards, 2009), 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 acquisi-

⁴<http://lemill.net/>

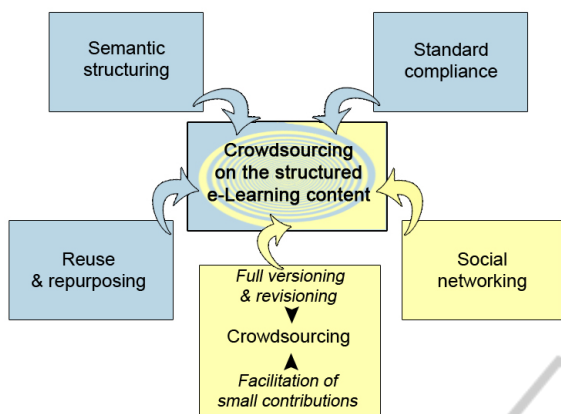


Figure 1: CrowdLearn concept.

tion 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, (Haake et al., 2005) introduces the concept of wiki templates that allow end-users to define the structure and appearance of a wiki page in order to facilitate the authoring of structured wiki pages. Similarly the hybrid wiki approach (Matthes et al., 2011) aims to solve the problem of using (semi-)structured data in wikis by means of page attributes. In our approach we apply the wiki paradigm to the creation and collaboration around (semi-)structured learning objects.

3 CrowdLearn CONCEPT

We see the CrowdLearn concept as an application of crowd-sourcing techniques to the e-learning content creation, re-purpose and reuse. As shown in 1, the concept is based on the following five strategies:

Standard-compliance. The costs associated with building high-quality e-learning content are high. One solution to decrease the costs is to author structured and reusable e-learning content that can be repurposed in different ways. To facilitate this, it should be possible to migrate content between different *Learning Management Systems* (LMSs). However, often content migration is not completely adequate and can thus result in loss of valuable content, meta-data or structure. Even if the transfer is possible, moving the content between systems can be more costly than just redeveloping that course in the new system. The strategy to overcome this challenge is the standard-compliance of both LMS and content. In that regard, we adopted the *SCORM standard* (sco, 2011a) and practical recommenda-

tions (sco, 2011b) and expanded the standard for the collaborative model.

Semantic Structuring. Instead of dealing with large learning objects (often whole presentations or tests), we decompose them into fine-grained *learning artifacts*. Thus, rather than a large presentation, user will be able to edit, discuss and reuse individual slides; instead of a whole test she/he will be able to work on the level of individual questions. This concept efficiently facilitates the reuse and re-purpose of the learning objects. To implement the concept, we employ the WikiApp approach, as discussed in 4.1.

Reuse and Re-purpose. The benefits of reuse and repurposing are: (1) increasing the cost efficiency of content creation, (2) increasing the quality of e-learning content and (3) supporting the evolution and adaptation to new requirements. To increase the economic efficiency of e-learning, content should be reused for a long period of time. Then, the development costs can be amortized over several years. However, the student expectations for higher quality e-learning experience increase, and new technology emerges so quickly that most courses need redeveloping every 3-4 year (Jones, 2002). Instead of the full redevelopment, the content can slightly evolve. In that case the courses remain competitive with regard to the provisioning of high quality e-learning content. The possibility to reuse and re-purpose is crucial for the e-learning content evolution. Also, re-purposing allows to increase the efficiency by teaching more learners with the same content.

Crowd-sourcing. There are already vast amounts of amateur and expert users which are collaborating and contributing on the Social Web. Harnessing the power of such crowds can significantly enhance and widen the distribution of e-learning content. Crowd-sourcing as a distributed problem-solving and production model is defined to address this aspect of collective intelligence (Howe, 2006). CrowdLearn as its main innovation combines the crowd-sourcing techniques with the creation of highly-structured e-learning content. E-learning material when combined with crowd-sourcing and collaborative social approaches can help to cultivate innovation by collecting and expressing (contradicting) individual's ideas. As Paulo Freire wrote in his 1968 book *Pedagogy of the Oppressed*, 'Education must begin with the solution of the teacher-student contradiction, by reconciling the poles of the contradiction so that both are simultaneously teachers and students...'. Therefore, crowd-sourcing in the domain of educational

material not only increases the amount of e-learning content but also improves the quality of the content.

Social Networking. The theoretical foundations for e-Learning 2.0 are drawn from *social constructivism* (Wang et al., 2012). It is assumed that students learn as they work together to understand their experiences and create meaning. In this view, teachers are knowers who craft a curriculum to support a self-directed, collaborative search and discussion for meanings. Supporting social networking activities in CrowdLearn enables students to proactively interact with each other to acquire knowledge. With the CrowdLearn concept we address the following social networking activities:

- Users can follow individual learning objects as well as other users activities to receive notification messages about their updates.
- Users can discuss the content of learning objects in a forum-like manner.
- Users can share the learning objects within their social network websites such as Facebook, Google Plus, LinkedIn, etc.
- Users can rate the available questions in terms of their difficulty.

Besides increasing of the learning process quality, social activities improve the quality of the created learning material. Even when answering a quiz, users can contribute 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 questions of assessment tests, or in other words through native learning activities.

4 CrowdLearn IMPLEMENTATION

We implement and evaluate the CrowdLearn concept with SlideWiki⁵ – a web-based crowd-learning platform. SlideWiki deals with two types of (semi-)structured learning objects: slide presentations and assessment tests. SlideWiki follows our proposed *WikiApp data model* to facilitate the creation, re-purpose and reuse of learning objects. In this section we first elaborate on the WikiApp data model and then discuss the technical implementation details of the SlideWiki application.

⁵<http://slidewiki.akswo.org>

4.1 WikiApp 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. A set of content objects connected by *part-of* relations can be arranged and manipulated in exactly the same manner, as an individual non-structured object. The model natively supports versioning and structuring of the different content objects.

The WikiApp data model comes with a *Domain Specific Language* (DSL)⁶ which allows the model-driven generation of CrowdLearn applications. We illustrate the WikiApp model in 2 and formally define it as follows:

Definition 1 (WikiApp data model). *The WikiApp data model \mathcal{WA} can be formally described by a triple $\mathcal{M} = (U, T, O)$ with:*

- U - a set of users.
- T - a set of content types with associated property types P_t having this content type as their domain.
- $O = \{O_{t \in T}\}$ with O_t being sets of content objects for each content type $t \in T$.

Each O_t consists of content objects $o_{t,i}$ with $i \in I_t$ being a suitable identifier set for the content objects in O_t . Each $o_{t,i}$ comprises a set of properties $P_{t,i} = Attr_{t,i} \cup Rel_{t,i}$. $Attr_{t,i}$ is a set of literal, possibly typed attributes, and $Rel_{t,i}$ is a set of relationships with other content objects; The only necessary attribute for all content objects is $c_{t,i}$, which contains the creation timestamp of the object $o_{t,i}$. $Rel_{t,i}$ can particularly include the following designated relationships to related objects:

- $part_{t,i} \subset O$ refers to set of the content objects contained in $o_{t,i}$;
- $base_{t,i} \in O_t$ refers to base content object from which the object $o_{t,i}$ was derived;
- $user_{t,i} \in U$ refers to a user being the owner of the $o_{t,i}$;

The WikiApp model assumes that all content objects are versioned using the timestamp $c_{t,i}$ and the base content object relation $b_{t,i}$. 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

⁶More information available at: <http://slidewiki.akswo.org/wikifier>

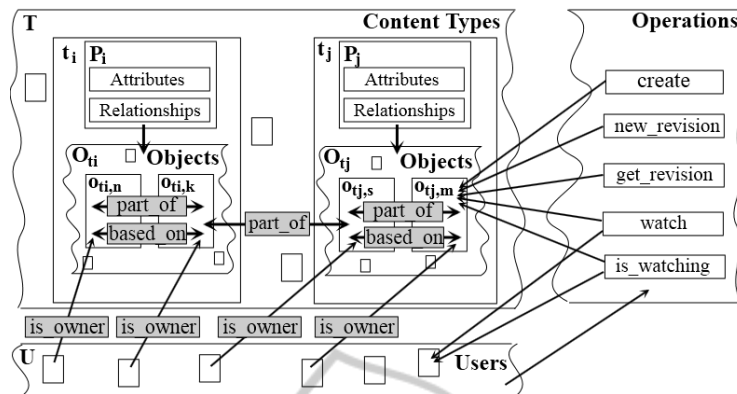


Figure 2: Conceptual view of the WikiApp data model.

this user being associated as their owner. In practice, however, usually only a subset of the content objects are required to be versioned. For auxiliary content (such as user profiles, preferences etc.) it is usually sufficient to omit a base content object relation. For reasons of simplicity of the presentation and space restrictions we have omitted a separate consideration of such content here. However, this is in fact just a special case of our general model, where the base content object relation $base_{t,i}$ is empty for a subset of the content objects.

The model is compatible with *both* the relational data model and the *Resource Description Framework* (RDF) data model (i.e. it is straightforward to map it to each one of these). When implemented as a relational data, content types correspond to tables and content objects to rows in these tables. Functional attributes and relationships as well as the *owner* and *base-content-object* relationships can be modeled as columns (the latter three representing foreign-key relationships) in these tables. The implementation of the WikiApp model in RDF is slightly more straightforward: content types resemble classes and content objects instances of these classes. Attributes and relationships can be attached to the classes via `rdfs:domain` and `rdfs:range` definitions and directly used as properties of the respective instances. For reasons of scalability we expect the WikiApp data model to be mainly used with relational backends. However, we also added a Linked Data interface using Triplify (Auer et al., 2009) (cf. 4.2).

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 im-

plementation is used for managing slides and presentations. It includes individual slides (consisting mainly of HTML snippets, SVG images and metadata), 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 assessment tests. It includes questions for the slide material (the question is assigned to all slide revisions), tests (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 answer the tests and have a look at 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 a new slide revision, that already has a list of questions, collected from other revisions. Secondly, we assigned assessment tests to concrete deck revisions. Thus the automatically created test saves the structure of the corresponding deck revision. This allows us to use module-based assessment to score the test results.

4.2 SlideWiki Implementation

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. The implementation comprises the main components: authoring, change management, import/export, linked data interface, e-

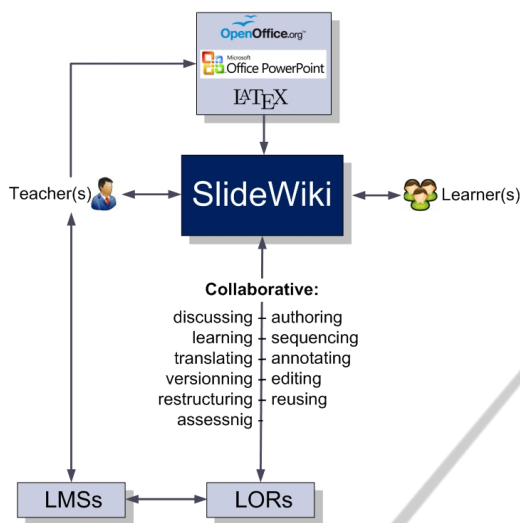


Figure 3: SlideWiki conceptual scheme.

assessment and translation. We briefly walk-through these components in the sequel.

Authoring. SlideWiki employs an inline HTML5 based WYSIWYG (What-You-See-Is-What-You-Get) text editor for authoring the presentation slides (cf. 5, image 1). 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 editor⁷ 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-edit⁸ with some predefined shapes which are commonly used in presentations. For logical structuring of presentations, SlideWiki utilizes a tree structure 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. Revision control is natively supported by WikiApp data model. We just define rules and restrictions to increase the performance. 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 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

⁷<http://aloha-editor.org/>

⁸<http://code.google.com/p/svg-edit/>

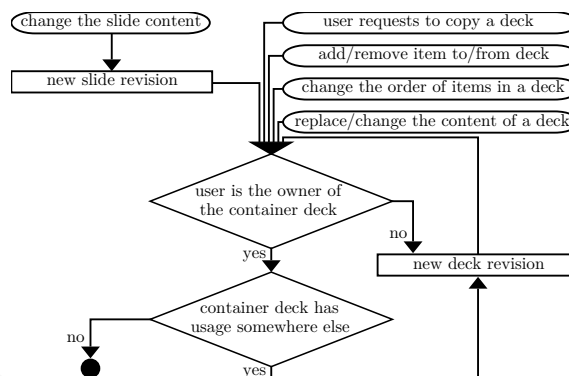


Figure 4: Decision flow during the creation of new slide and deck revisions.

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.

The decision flow is presented in 4. 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.

Import/Export. SlideWiki implementation addresses *interoperability* as its first class citizen. As shown in 3, SlideWiki supports import/export of the content from/to existing desktop applications and LORs thereby allowing users from other LMSs to access the created content. The main data format used in SlideWiki is HTML. However, there are other popular presentation formats commonly used by desktop application users, such as PowerPoint .pptx presentations, LaTeX and others. We implemented import of the slides from .pptx format and work on the LaTeX format support is in progress.

Linked Data Interface. While sharing and reusing educational data across institutional and national boundaries is a general goal for both the public and the private education sector, the last decade has seen a large amount of research dedicated to Web-scale interoperability. For example, LinkedEducation.org is

The figure consists of four numbered screenshots of the SlideWiki interface:

- 1**: A screenshot of the SlideWiki editor showing a tree structure on the left and a slide titled "E-Business E-Commerce" in the center. The slide content discusses electronic commerce and includes a list of bullet points.
- 2**: A screenshot of the "Edit Question for the slide" dialog box. It shows a question being edited and an "Add to a list" dropdown menu with options like "125" and "Manually created tests".
- 3**: A screenshot of a test titled "Test for 'Vorlesungsreihe Information'" in learning mode. It shows a question about SOA architecture styles with a list of checkboxes for correct answers: Schnittstellenorientiert, Nachrichtenorientiert, Datenorientiert, Ressourcenorientiert, and Visualisierungsorientiert.
- 4**: A screenshot of a scoring table for a test. It shows two tables: "BIS: ERP" and "Beispiel SAP". Each table has columns for "Question", "Difficulty", and "Points". Below the tables are summary statistics for different question types and a "your best is" score.

Figure 5: Four screenshots of SlideWiki features. **1** - Tree structure of the presentation, inline WYSIWYG editor; **2** - Editing of a question and manual assigning to a test using lists; **3** - Question in learning mode with correct answers displayed; **4** - Module-based scoring of an assessment test.

an open platform which promotes the use of Linked Data for educational purposes. In order to enable the export of SlideWiki content on Data Web as LORs, we employed the RDB2RDF mapping tool Triplify (Auer et al., 2009) to map SlideWiki content to RDF and publish the resulting data on the Data Web. The Triplify configuration for SlideWiki was created manually according to IEEE LOM standard and can be changed to support specific LORs. The SlideWiki Triplify Linked Data interface is available via: <http://slidewiki.aksw.org/triplify>.

e-Assessment. SlideWiki supports the creation of questions and self-assessment tests based on slide material. Each question has to be assigned to 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 tests. The *automatically created* tests include the last question revisions from all the slides within the current deck revision.

Manually created tests present a collection of chosen questions and currently cannot be manipulated as objects (cf. 5, image 2). 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.

For now, only multiple-choice (and multiple-mark) question type is implemented, however in the future we plan to expand the list of supported types. To score the results the student (or the teacher) can choose one of five implemented algorithms. Thus, dichotomous scoring gives points only when the answer was fully correct, other algorithms also count partially answered questions. Within other algorithms we implemented our balanced approach for scoring the multiple-mark questions, described in details in (?). All five algorithms use the dynamically accumulated difficulty d of the question as the number of points for

the fully correct response:

$$d = \frac{incorr}{all} \quad (1)$$

If the user prefers to use dichotomous scoring, the values of *incorr* and *all* mean, respectively, the accumulated number of *incorrect* answers and *all* answers of that question by any of users. In a case of partial scoring, *incorr* is determined as follows:

$$incorr = \sum_{i=1}^n \left(1 - \frac{p_i}{d_i}\right), \quad (2)$$

where n - number of attempts for the question, d_i - difficulty (or maximal points), that the question had at the moment, when the i^{th} attempt was made, p_i - points obtained in the i^{th} attempt

After the difficulty is determined, it's scaled to $(1, d_{max}]$, where d_{max} is the maximal weight, that a question can have. d_{max} is set up by the system administrator only for the users' comfort. In SlideWiki we set it up to 5.

Students can start a chosen test (manually created or automatically collected) in one of two possible modes: "learning" or "examination". In learning mode student can ask to show the slide, to which the question is assigned to remind the material, or simply show the correct answers (cf. 5, image 3). Thus, student should not spend time to find the material. However, after the user asked to show her/him either the slide or correct answers she/he will not get any points for the question. In examination mode these features are disabled.

After choosing the mode the user can also set up the amount of questions (all, all the difficult or concrete amount) and the order to show them (random or increase the difficulty). As the amount of questions can differ for the same test, we show the test results as a percentage of the maximum points for exactly this selection of question.

Our architecture allowed us to implement module-based scoring. Each module of the assessment test presents a sub-deck of the presentation and is scored individually. Then, all the "parent" modules are scored as a sum of "children" points and finally the whole test is scored as a sum of all the points for all the modules (cf. 5, image 4).

Translation. Our architecture allowed us to implement a translation feature backed by the Google Translate service. After the translation into one of 54 supported languages, the presentation can be edited independently from the original one. However, we store the information about the original version and now we are working on the possibility to partially update the translated presentation when the original one

was changed. Without such a possibility we faced with an important challenge. As we allow the owner of the revision to change it without creation of a new revision, it was an important issue: either we should allow the multiple translation of the same revision into the same language or not. For now we decided to allow it, however, this led us to the situation, that we would get several identical presentations with the content of bad quality, as it was translated automatically and was not edited manually. However, we could not disable the multiple translations, as in that case it would be for example impossible to get the translations of new slides, if they were added by the owner. Thus, the connection between original and translated presentations seems to be crucial. Also we think about supporting of the user and community thesauri.

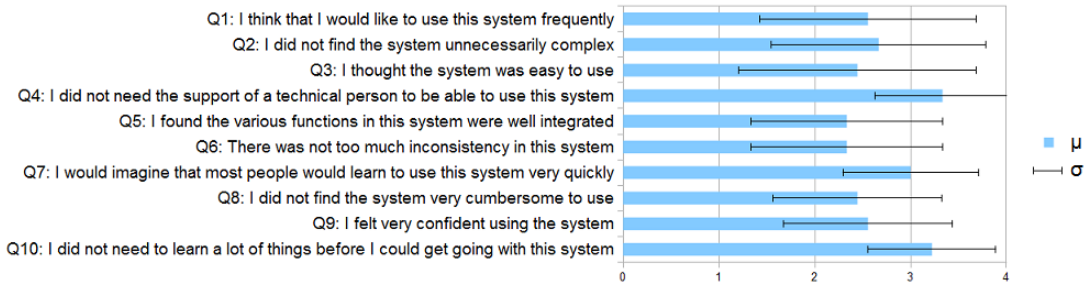
5 EVALUATION

To evaluate the real-life usability of SlideWiki, we used it for accompanying an information systems lecture at Chemnitz Technical University. We structured the slides within the lecture series and added questions for student self-assessment before the final exam. We informed them about the different e-learning features of SlideWiki, in particular, how to prepare for the exam using SlideWiki. The experiment was not obligatory but students actively contributed by creating additional questions and fixing mistakes. The experiment was announced to 30 students of the second year and 28 of them registered at SlideWiki.

The students were working with SlideWiki for several weeks, and we collected the statistics for that period. During that period, they created 252 new slide revisions which some of them were totally new slides, others were improved versions of the original lecture slides. Originally the whole course had 130 questions, and students changed 13 of them, fixing the typos or adding additional distractors to multiple choice questions. In total, students performed 287 self-assessment tests. The majority of these used the automatically and randomly created tests covering the whole course material. 20 tests included only difficult questions, 2 asked to show the questions with increasing difficulty. This showed us that the students liked the diversity of test organization. Students also liked the possibility to limit the number of questions – 80 attempts were made with such a setting. 8 students reached the 100% result for the whole course. On average, it took them 6 attempts before they succeeded.

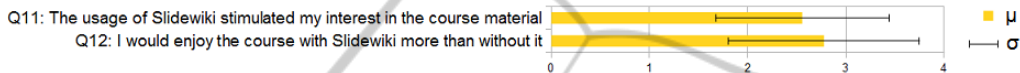
After the experiment we can claim, that more ac-

Questions recommended by SUS technique



Questions for measuring the quality of learning

SlideWiki overall



Assessment tool



Figure 6: Results of SlideWiki evaluation survey: mean μ and standard deviation σ .

tive SlideWiki users received better marks on the real examination. It shows that SlideWiki not only allows students to prepare for the examinations, but also engages them in active participation that helps to improve the quality of the learning. After the end of the semester, we asked the participants to fill out a questionnaire which consisted of three parts: usability experience questions, learning quality questions and open questions for collecting the qualitative feedbacks. We collected 9 questionnaires that were filled out completely. They show us emergent problems and directions for the future.

In the first part of the questionnaire we included questions recommended by *System Usability Scale* (SUS) (Lewis and Sauro, 2009) system to grade the usability of SlideWiki. SUS is a standardized, simple, ten-item Likert scale-based questionnaire⁹ 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 67.2 for SlideWiki which indicates a reasonable level of usability.

The second part of the questionnaire aimed to determine whether the SlideWiki helps to improve the quality of learning. It consisted of four questions with five options from “absolutely agree (1)” to “absolutely disagree (5)”. The evaluation results for these two parts are presented in 6.

Although the positive answers prevail, we were

⁹www.usabilitynet.org/trump/documents/Suschapt.doc

not satisfied by the fact that for many questions a third of participants chose the neutral value. It could be a signal, that students do not completely understand the question or are not 100% sure about the result. The last part of the questionnaire helped us to understand the reasons. We included four open questions:

1. What did you like most about Slidewiki?
2. What did you like least about Slidewiki?
3. What can we do to improve the Slidewiki’s usability?
4. What features would you add to Slidewiki?

Within the answers we found repeated complaints about several bugs, that interfered the working process. We consider this fact to be the main reason of neutral and contradictory values. However, we collected also positive opinions, especially about features and possibilities that SlideWiki allows. Three of the recipients mentioned that they mostly liked that SlideWiki is easy to use, four of them noted, that they liked the idea of collaborative work and sharing the presentations itself. Within the collected answers we also got important suggestions, which could be roughly divided into two groups:

- Suggestions about desired improvements of existing features such as displaying the test results graphically, supporting more import formats, improving the SVG editor etc.
- Suggestions about totally new features, several of those were later implemented, e.g. translation, templates for presentation structure, etc.

Also we collected a few suggestions about features

that were already implemented, but users were not aware of them. This encourages us to improve the documentation as well as to enhance the simplicity and clearness of the user interface. One of the students drew our attention to security issues.

The results of our evaluation showed that our concept is clear to the users, they like this way of learning, storing and sharing of the presentations. However, we need to improve the user interface, fix some minor bugs and spend more effort on privacy and security issues.

6 CONCLUSIONS AND FUTURE WORK

In this paper we presented the CrowdLearn concept that applies collaborative authoring and crowdsourcing techniques to the creation of (semi-) structured e-learning content. The concept is based on the SCORM concepts and uses the novel WikiApp data model to organize the content closely aligned with the standard. We implemented and evaluated the concept with SlideWiki – a social web e-learning application targeting slide presentations and e-assessments. While the evaluation results were promising, we still need to extend the concept in the future to address the requirements requested by users.

Beside the usability improvements, our first direction for future work is to implement a completely SCORM-compliant LMS and authoring tool, based on the SlideWiki. This will allow us to exchange the content with other SCORM-compliant LMSs. Also, in a real e-learning scenario, learners come from different environments, have different ages and educational backgrounds. These heterogeneities in user profiles are crucial to be addressed when enhancing the CrowdLearn concept. New approaches should provide the possibility to *personalize* the learning process. Thus, our second direction is providing the personalized content based on initial user assessments. The third direction for the future work is to support the annotation of learning objects using standard metadata schemes. We aim to implement the *LRMI*¹⁰ metadata schemes to facilitate end-user search and discovery of educational resources.

ACKNOWLEDGEMENTS

We would like to thank the AKSW research group members for their support during the evaluation of

¹⁰Learning Resource Metadata Initiative: www.lrmi.net/

SlideWiki. This work was supported by a grant from the European Union's 7th Framework Programme provided for the project LOD2 (GA no. 257943) and by a grant from the Saxon State Ministry for Higher Education, Research and the Arts (SMWK) and the Development Bank of Saxony (SAB) for the eScience project (grant no. 080951807).

REFERENCES

- (2011a). SCORM 2004 4th Edition Specification. Technical report, ADL.
- (2011b). SCORM Users guide for programmers. Technical report, ADL.
- Auer, S., Dietzold, S., Lehmann, J., Hellmann, S., and Aumueller, D. (2009). Triplify: Light-weight linked data publication from relational databases. In *WWW2009, Spain*. ACM.
- Barriocanal, E. G., Sicilia, M.-Á., Alonso, S. S., and Lyras, M. D. (2011). Semantic annotation of video fragments as learning objects. *Interactive Learning Environments*, 19(1):25–44.
- Devedzic, V. (2006). *Semantic Web and Education (Integrated Series in Information Systems)*. Springer, Secaucus, NJ, USA.
- Haake, A., Lukosch, S., and Schümmer, T. (2005). Wikitemplates: adding structure support to wikis on demand. In *Proceedings of the 2005 International Symposium on Wikis*, pages 41–51. ACM.
- Howe, J. (2006). The rise of crowdsourcing. *Wired Magazine*, 14(6).
- Jones, E. (2002). Implications of scorm and emerging e-learning standards on engineering education. *Proceedings of the 2002 ASEE Gulf-Southwest Annual Conference*.
- Kuo, Y.-H., Kinshuk, Q. T., Huang, Y.-M., Liu, T.-C., and Chang, M. (2008). Collaborative creation of authentic examples with location for u-learning. In *International Conference e-Learning 2008*, pages 16 – 20. IADIS.
- Leuf, B. and Cunningham, W. (2001). *The Wiki way: quick collaboration on the Web*. Addison-Wesley, London.
- Lewis, J. and Sauro, J. (2009). The Factor Structure of the System Usability Scale. In *Human Centered Design*, volume 5619 of *LNCS*, chapter 12, pages 94–103. Springer.
- Matthes, F., Neubert, C., and Steinhoff, A. (2011). Hybrid wikis: Empowering users to collaboratively structure information. In *ICSOFI (1)*, pages 250–259.
- Pedreira, N., Salgueiro, J. R. M., and Carballo, M. M. (2009). E-learning in new technologies. In *Encyclopedia of Artificial Intelligence*, pages 532–535. IGI Global.
- Richards, D. (2009). A social software/web 2.0 approach to collaborative knowledge engineering. *Inf. Sci.*, 179(15):2515–2523.
- Wang, X., Love, P. E., Klinc, R., Kim, M. J., and Davis, P. R. (2012). Integration of e-learning 2.0 with web 2.0. *ITcon - Special Issue eLearning 2.0: Web 2.0-based social learning in built environment*, 17:387–396.