Content Co-creation and Refinement for Microlearning Settings

Kazuhiro Kuwabara

College of Information Science and Engineering
Ritsumeikan University
Kusatsu, Japan
e-mail: kuwabara@is.ritsumei.ac.jp

Boonsita Roengsamut

Graduate School of Information Science and Engineering
Ritsumeikan University
Kusatsu, Japan
e-mail: gr0186ii@ed.ritsumei.ac.jp

***Abstract*—This paper describes an approach to creating and refining e-learning contents. As mobile devices such as smart phones become popular, learning can be conducted without the constraints of time and place, and the learning activity may become fragmented into many small learning sessions. In such microlearning activity settings, e-learning contents tend to be small making it easier to create contents by the collaborative efforts of many people. In order to facilitate such collaborative creation activities, we make use of the framework of a domain ontology and propose an interactive refinement process for when a problem is found in its contents. In addition, by letting learners participate in the refinement process of the domain ontology, the learning activity is expected to be more effective.**

Keywords–e-learning; domain ontology; linked data; crowd-sourcing.

I. INTRODUCTION

As high-speed Internet environments become ubiquitous, Massive Open Online Courses (MOOCs) [1] are becoming increasing popular. Mobile devices such smart phones allow users to learn almost anytime and anywhere. Microlearning [2], where small contents are often used in a learning activity, is attracting much attention. For example, mobile phones were utilized for microlearning interactions to rehearse names and faces for a social event [3].

It usually takes considerable effort to create e-learning contents, involving many human resources. When the power of a large group of people is utilized, as is often seen in an Internet environment (e.g., the Wikipedia project), smaller e-learning contents are easier to handle.

In this paper, we focus on content creation by a large number of people. In order to produce high quality contents, we propose an approach to refining the contents. The refining process is conducted interactively between a user and a system, and is initiated when a problem is found in the contents. In order to extract necessary information from a user to fix the problem, the system produces a list of choices using the domain ontology, from which a user selects the appropriate one. By aggregating answers from many users, we can obtain useful information to refine either the contents or the domain ontology. When the user is a learner, this interaction process can also contribute to the learning experience.

In addition, in order to achieve an effective learning process, (micro-) contents need to be presented in accordance with the context of a learner. A *scenario* is introduced, which is created for each learner in the learning activity, and executed so that proper contents can be shown to the learner.

The remainder of the paper is structured as follows. The next section discusses some related works, and Section III presents the data model for the proposed e-learning contents. Section IV describes the refinement process of the domain ontology and e-learning contents, and Section V presents the prototype implementation. The final section concludes this paper with directions for future works.

II. RELATED WORKS

The hot topic of *crowdsourcing* is a method to harness collaborative human efforts [4]. CrowdLearn is an approach to apply the concept of crowdsourcing to the creation of educational contents [5]. It aims to create educational contents that follow the Sharable Content Object Reference Model (SCORM), a collection of standards and specifications for web-based e-learning. Contents are created collaboratively by using the SlideWiki [6] platform. This system has been further extended to handle personalization and multilingual contents [7]. The primary function of CrowdLearn is to make full-fledged e-learning content; however, this paper targets e-learning contents used in more casual settings.

Related to the ontology refinement in e-learning settings, there is a system for learning a concept map [8] using a game-like tool called *Termina* [9]. In a concept map, terms that represent a particular concept are linked to other terms representing other concepts. Making a learner aware of the concept map of the target domain provides an opportunity to learn the target domain in more depth. Since a concept map is related structurally to a domain ontology [10], refinement of the domain ontology by a learner can contribute to the learner's learning experience in a similar way.

III.DATA MODEL

We use Linked Data [11] as an underlying data model for presenting e-learning contents. Linked Data is based on the Resource Description Framework (RDF) [12], which is the standard in the Semantic Web. A Uniform Resource Identifier (URI) is assigned to a thing that is represented, and the data is represented as a triple consisting of the subject, predicate, and the object.

An example of a simple learning content is shown in Fig. 1. This figure shows a multiple choice type quiz targeted at studying the JavaScript programming language. It has tags that correspond to the contents of this quiz (`ex:for`, `ex:javascript`), which indicate that this quiz is related to the *for* command sentence in the JavaScript programming language.

Since we assume that the contents are basically of small

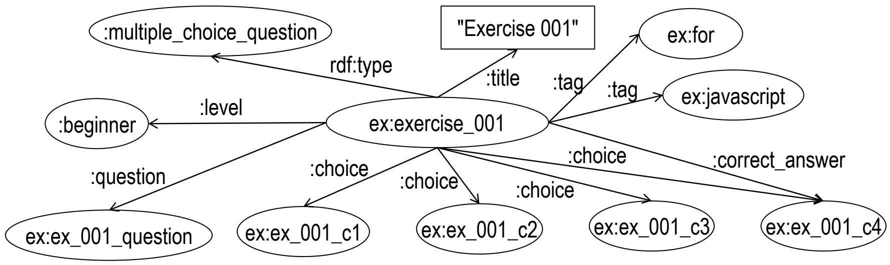

Figure 1. Example exercise

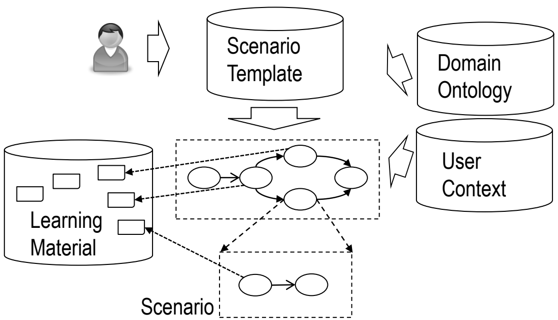

Figure 2. Scenario creation and execution

size and independent of each other, the context of the user (learner) needs to be considered when contents are presented to the user. The context of the user includes the kind of learning contents they have studied and to what extent they understand them. In order to tackle this problem, a scenario is created, according to which the learning contents are presented to the user. A given scenario is created based on a template scenario, along with the domain ontology and the user's context (Fig. 2).

Part of an example domain ontology is described as shown in Fig. 3. In this example, the concept of *control structure* is represented, under which concepts such as *if*, *while*, *for*, or *switch* are defined.

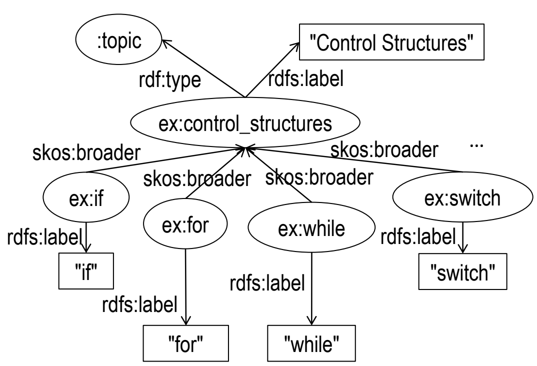

Figure 3. Part of example domain ontology

A scenario template is intended to be used to define an overall flow of presenting e-learning contents to a user. It is selected based on the intention and situation of the user. The selected scenario template is then instantiated with the domain ontology. For example, if the user intends to study the JavaScript programming language, its domain ontology is used. If the user selects a particular topic, say, *control structure*, the exercise related to the topics under *control structure* will be selected.

In addition, the context of the user, such as their level of understanding, is taken into consideration to finalize the scenario. For example, if the user is a beginner, the scenario will be finalized with a link to the beginner-level exercises.

As with a subroutine in a programming language, a scenario can be invoked from another scenario, creating a hierarchical structure of scenario execution. For example, the scenario that presents the e-learning contents of *control structure* to a user may invoke another scenario to present the contents corresponding to the topic of *if*. After its execution, another scenario presenting the contents of the topic of *for*, for example, may be invoked.

When a scenario is executed, all the scenarios that may be invoked are not necessarily instantiated from the beginning; a scenario is instantiated just before it is invoked. In the above example, the scenario for *for* will be finalized only when the topic of *for* is presented.

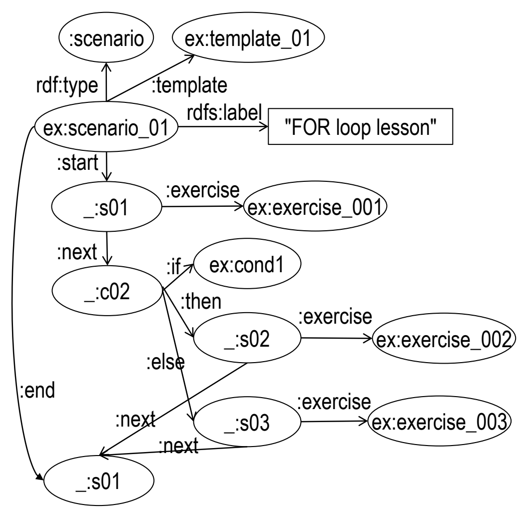

Figure 4. Example of an instantiated scenario

An instantiated scenario is a collection of links to the e-learning contents. A simple example scenario of a learning session regarding the *for* loop is shown in Fig. 4. Fig. 5 shows a serialized RDF representation in Turtle syntax of the example depicted in Fig. 4.

In this example scenario, the exercise represented by `ex:exercise_001` is presented first. The exercise presented to

```
ex:scenario_01 rdf:type :scenario ;
  :template ex:template_01 ;
  rdfs:label "FOR loop lesson" ;
  :start
    [ :exercise ex:exercise_001 ;
      :next [ :if ex:cond1 ;
              :then [ :exercise ex:exercise_002;
                      :next _:s01 ; ] ;
              :else [ :exercise ex:exercise_003;
                      :next _:s01 ; ] ;
            ];
      ];
  :end _:s01 .
```

Figure 5. Example scenario in the Turtle representation

the user next (`ex:exercise_002` or `ex:exercise_003`) depends on its result.

IV. Content and Ontology Refinement

Based on the data model presented above, the tag to the e-learning content is automatically attached as follows. First, keywords are extracted from the sentences in the contents, using a morphological analyzer. Then, the extracted keywords are compared with a label of the concepts in the domain ontology. A tag represents a concept defined in the domain ontology.

Tag information is viewed when learning contents are searched for. Thus, the correctness of the tag is important. When an incorrect tag is found by a user (a teacher or a student), a refinement process is initiated. This refinement process is basically an interactive process between the user and the system. When an error is detected, the system finds modifications that are needed to fix the problem (Fig. 6).

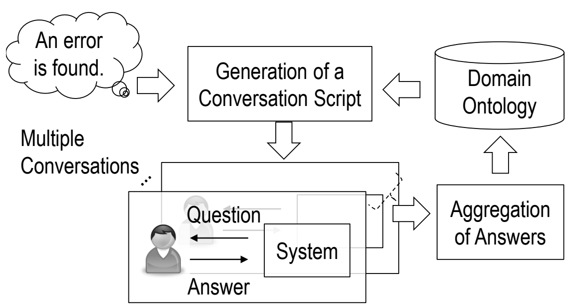

Figure 6. Refinement of the domain ontology

Let us suppose that the cause of the problem is an incorrect domain ontology. Since the domain ontology is also represented as an RDF graph, the problem stems from an incorrect or missing link. In order to identify the cause of the problem, the system tries to obtain the information necessary to fix the problem from a human user. This is done by first presenting a choice list to the user, from which the user is expected to select the appropriate answer. From the information provided by the user, the system recalculates the tag and presents it to the user. If the user confirms the revised tag, the process terminates; otherwise, the system tries to find another set of choices to present to the user. Fig. 7 represents this sequence in a Unified Modeling Language (UML) diagram.

Let us consider a simple example in the domain of the JavaScript programming language. Let us suppose that there is an exercise regarding a *ternary operator* (conditional operator), which is described using `?:`. This operator is similar to the *if-then-else* control structure in the sense that an expression written with a ternary operator can be expressed in the *if-then-else* control structure. If the exercise related to the ternary operator appears during the learning session of the *control structure*, it should be treated as an error. For the sake of the example, let us suppose that the concept of *ternary operator* is defined as a sub-concept of *control structure*. In this case, the cause of this error is an incorrect structure in the domain ontology. The best possible fix would be to change the upper concept of *ternary operator*. The system would present a list of concepts at the same level as *control structure* from which a user can choose the proper upper concept of *ternary operator*. When the user selects, say, *operators*, the concept of *ternary operator* will be changed to a sub-concept of *operators*. With this modification, an exercise tagged with *ternary operator* will no longer be shown under the topic of *control structure*.

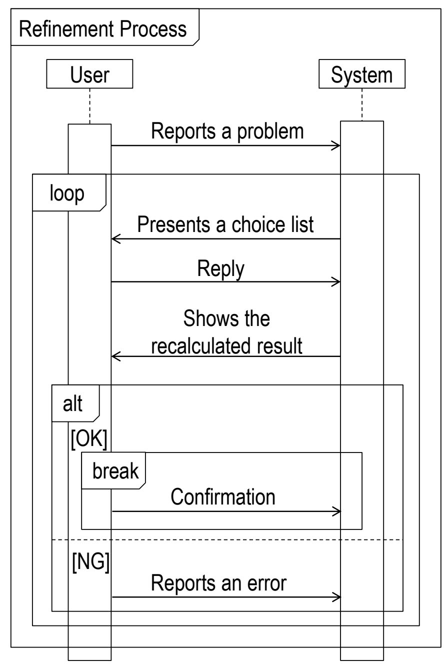

Figure 7. Interaction between a user and a system

If the user is a teacher, the update to the domain ontology is performed immediately. If the user is a student, the update is not immediately made to the domain ontology. Instead, a temporary area is set aside to store the update of the domain ontology. After the answers from multiple users are aggregated, and the majority of users concur, the change is made to the domain ontology itself.

V. Implementation

The system itself can be divided into two parts: content creation and refinement, and content presentation to the user. As mentioned before, the e-learning contents are represented using the RDF. In the prototype system we are currently implementing, we use the Fuseki server of the Apache Jena project [13] as a back-end RDF store. As for the front-end Web server, we use Node.js, which handles user authentication among other things (Fig. 8). Basically, e-learning contents are stored in the RDF store. The RDF query and update language SPARQL [14] is used to retrieve and update the contents from the Web Server using HTTP. The user interface is implemented using HTML and JavaScript. In order to push data from the Web server to the browser, we use WebSocket—more specifically the *socket.io* library—for easier implementation.

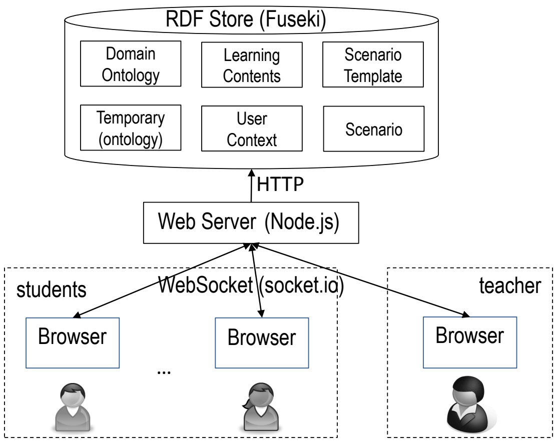

Figure 8. System implementation overview

As for the content creation and refinement, we consider the user and the system as a *software agent*, and define an agent interaction protocol [15] for the refinement of the domain ontology and e-learning contents. The interaction protocol is executed cooperatively by the programs at the Web server and the browser. Since WebSocket is used for communication between the Web server and a browser, bi-directional communication can easily be implemented.

As for the contents presentation, the system implements an engine that executes a scenario such as the one shown in Fig. 5. According to the user's context, a scenario template is selected and instantiated dynamically using a domain ontology. The instantiated scenario is stored in the RDF store and retrieved using SPARQL query language.

The execution of a scenario is conducted cooperatively by a Web server and a browser, and is specified as an interaction protocol between the system and the user. In microlearning settings, the execution of a scenario will likely often be interrupted as the situation of a user changes dynamically. The execution status will be stored at the Web server so that, when the user restarts the learning session, the interrupted execution of a scenario can be resumed. Alternatively, the user may choose to initiate a complete new learning session.

VI. CONCLUSION AND FUTURE WORK

This paper presents an approach to supporting the creation and refinement of e-learning contents. Since the main target of the proposed system is microlearning, we mainly consider creating smaller, independent contents. In order to provide a coherent learning experience while utilizing independent contents, we also consider a scenario by which the learning contents to be presented can be specified.

Currently, we are implementing the environment for educational content creation and playback described in Section V. As for the contents themselves, we are working on the domain of programming languages such as JavaScript. In addition to typical educational contents, we are also considering transferring other types of knowledge. For example, we have implemented a FAQ knowledge base for troubleshooting often needed in rental apartments [16]. The proposed framework is intended to take care of such kinds of knowledge contents in future applications.

Regarding future directions, we plan to diversify and extend the interaction protocols available to deal with refinement of domain ontologies and learning contents. Using the extended interaction protocols, we also plan to evaluate how effectively learner's participation contributes to the refinement process from the viewpoint of the learner's experience.

REFERENCES

[1] A. McAuley, B. Stewart, G. Siemens, and D. Cormier, "The MOOC model for digital practice," 2010. [Online]. Available from: http://www.edukwest.com/wp-content/uploads/2011/07/MOOC_Final.pdf [retrieved: Jan., 2015]

[2] T. Hug, Ed., Didactics of Microlearning: Concepts, Discourses and Examples. Waxmann Verlag GmbH, Germany, 2007.

[3] J. S. Beaudin and S. S. Intille, "Microlearning on a mobile device," Proceedings of UbiComp 2006 Extended Abstracts (Demo Program), 2006.

[4] A. Doan, R. Ramakrishnan, and A. Y. Halevy, "Crowdsourcing systems on the world-wide web," Commun. ACM, vol. 54, no. 4, April 2011, pp. 86–96.

[5] D. Tarasowa, A. Khalili, S. Auer, and J. Unbehauen, "CrowdLearn: Crowd-sourcing the creation of highly-structured e-learning content," The 5th International Conference on Computer Supported Education (CSEDU 2013), 2013, pp. 33–42.

[6] A. Khalili, S. Auer, D. Tarasowa, and I. Ermilov, "SlideWiki: Elicitation and sharing of corporate knowledge using presentations," The 18th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2012), Lecture Notes in Computer Science (LNCS) 7603. Springer, 2012, pp. 302–316.

[7] D. Tarasowa, S. Auer, A. Khalili, and J. Unbehauen, "Crowd-sourcing (semantically) structured multilingual educational content (CoSMEC)," Open Praxis, vol. 6, no. 2, 2014, pp. 159–170.

[8] J. D. Novak and A. J. Cañas, "The theory underlying concept maps and how to construct them," Technical Report IHMC CmapTools 2006-01 Rev 01-2008, 2008.

[9] F. Kneissl and F. Bry, "Fostering concept maps awareness as a means to learning," Third International Conference on Cloud and Green Computing (CGC), 2013, pp. 354–357.

[10] J. Jovanović, D. Gašević, C. Knight, and G. Richards, "Ontologies for effective use of context in e-learning settings," Educational Technology & Society, vol. 10, no. 3, 2007, pp. 47–59.

[11] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data - the story so far," International Journal on Semantic Web and Information Systems, vol. 5, no. 3, March 2009, pp. 1–22.

[12] RDF Working Group, W3C, "Resource description framework (RDF)," [Online]. Available from: http://www.w3.org/RDF/. [retrieved: Jan., 2015]

[13] "Fuseki: serving RDF data over HTTP." [Online]. Available from: http://jena.apache.org/documentation/serving_data/index.html [retrieved: Jan., 2015]

[14] "SPARQL 1.1 Query Language." [Online]. Available from: http://www.w3.org/TR/sparql11-query/ [retrieved: Jan., 2015]

[15] B. Bauer and J. Odell, "UML 2.0 and agents: how to build agent-based systems with the new UML standard," Engineering Applications of Artificial Intelligence, vol. 18, no. 2, 2005, pp. 141–157.

[16] S. Kinomura and K. Kuwabara, "Developing a multilingual application using linked data: A case study," The 5th International Conference on Computational Collective Intelligence Technologies and Applications (ICCCI 2013), Lecture Notes in Computer Science (LNCS) 8083, Springer, 2013, pp. 120–129.