Toward a Crowdsourcing Platform for Knowledge Base Construction

Kazuhiro Kuwabara and Naoki Ohta
College of Information Science and Engineering
Ritsumeikan University
Kusatsu, Japan
emails: {kuwabara@is.ritsumei.ac.jp, n-ohta@fc.ritsumei.ac.jp}

***Abstract*—This paper proposes an approach to construct a knowledge base using crowdsourcing where the knowledge base is represented as linked data. With the crowdsourcing concept, the contents of a knowledge base are accumulated. We represent the process of knowledge base construction as a workflow. The ontology of the knowledge base's target domain is utilized in creating and executing workflows. Applications to the construction of knowledge bases in the domains of e-learning content and multilingual frequently asked questions (FAQs) are described as examples. We discuss our proposed approach from the viewpoint of a crowdsourcing platform that facilitates the use of the crowdsourcing concept to construct a knowledge base, and show how the domain ontology can be made use of.**

Keywords-crowdsourcing; linked data; ontology; knowledge base; e-learning.

I. INTRODUCTION

Crowdsourcing is attracting much attention as an approach to exploit the power of many people [3], [5], [16], [21]. For example, Wikipedia was basically created by volunteers on the Internet. Such web sites as Amazon Mechanical Turk provide crowdsourcing services.

In this paper, we propose an approach that exploits the crowdsourcing concept in knowledge base construction. With typical crowdsourcing, a given job is divided into smaller independent tasks. When such a division is difficult, coordinating the execution of tasks becomes an issue. For example, constructing a knowledge base in a company or in a group requires many tasks that involve creating a piece of knowledge. Since these tasks are not independent from each other, the crowdsourcing approach is not easily applied.

CrowdForge was proposed as a framework for applying the crowdsourcing concept to such complex tasks as writing a magazine article from a scientific journal paper [9]. In addition, TurKit, a scripting language, was proposed to specify the crowdsourcing's control flow as a script program [13]. However, it remains unclear how they can be applied to the construction of a knowledge base using domain ontology.

Here, we focus on an approach that utilizes the domain ontology to partition a task into sub-tasks and integrate their results in the construction of knowledge bases. Based on this approach, our framework can be customized to suit a particular domain for which a knowledge base is constructed by providing the domain dependent ontology. We assume that the contents of a knowledge base are represented as linked data [1], whose basic idea was proposed in the area of the semantic web to create the so-called Web of Data. Since linked data are inherently web-based, they are compatible with crowdsourcing.

This paper is structured as follows. The next section describes a crowdsourcing platform based on our proposed approach. Section III shows example scenarios using our proposed framework, and Section IV discusses related works and the features of the proposed framework. The final section concludes this paper.

II. CROWDSOURCING PLATFORM

A. Overview

The target of our proposed crowdsourcing platform is the construction of a knowledge base in a specific domain. We assume that the ontology of the target domain is provided, which is utilized to customize the platform.

The workflow of the knowledge base construction must maintain the quality of the knowledge base. The workflow is basically comprised of the division into tasks and the aggregations of the task results. The workflow's execution is monitored so that system administrators can grasp the progress of the knowledge base construction and intervene if necessary.

In addition, we consider a case where a human task and a program-based service coexist. For example, to construct a multilingual knowledge base, the contents of the knowledge base must be translated. Since many machine translation services are available, we can use such a service or a human translation service. From the viewpoint of knowledge base construction workflow, it is preferable that both the human and machine translation services be treated with the same programming interface. To achieve this, we introduce the concept of a software agent. Each task's interface is defined as an interaction with a software agent. If an individual task is to be executed by a human, then the software agent acts as an intelligent user interface to the human user. If a particular task is executed by a web service, the software agent acts as a wrapper function to that web service.

This resembles the orchestration of such web services as Business Process Execution Language (BPEL), which is extended so that a human task can be incorporated in the orchestration of services (BPEL4People [6]). BPEL4People focuses on tasks that can only be executed by humans, such as decision authorizations. In contrast, our tasks can be interchangeably executed either by a human or a web service to increase workflow flexibility.

Users of the proposed platform include not only contrib-

 ISBN: 978-1-61208-329-2

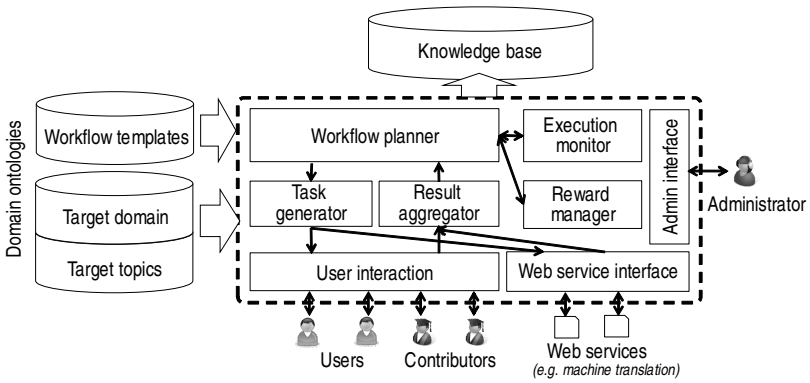

Figure 1. Overview of proposed crowdsourcing platform

utors to the knowledge base construction but also guest users who may just make comments or revision requests. In addition to them, we also assume administrator-type users who are regarded as owners of the job of constructing a knowledge base. We provide a function to monitor the progress of the knowledge base construction.

How to determine the rewards given to contributors and/or users is another issue. In the case of Wikipedia, volunteers basically contribute their time and knowledge to it. For Amazon Mechanical Turk, monetary compensation is standard, and determining rewards is important. To handle this reward issue, we define a reward manager in the platform.

Fig. 1 shows an overview of the proposed platform that incorporates the above functions. In addition, we borrow the idea of the blackboard model, a classical distributed problem solving model [7]. We focus on one that separates data and goal blackboards (Fig. 2).

The blackboard model consists of a blackboard and various knowledge sources. Each knowledge source is supposed to work on the data posted on the blackboard and write its results on it. The data on the blackboard define the level of abstraction, For example, in a typical blackboard application that interprets sensor signals, the lowest data level is the sensor's output signals, and the highest data level is the result of their interpretation. The knowledge source is assumed to process the sensor's output signals, and the results are written on the blackboard. Then, another knowledge source for a higher level works on the results of the lower level. The blackboard model with a separate goal blackboard was proposed to more easily control the execution of knowledge sources [12].

In our proposed crowdsourcing platform, a task is posted to the goal (task) blackboard, and the task's result is written on the data blackboard. The difference with the original blackboard model is that two kinds of tasks are considered. One further divides a task into sub-tasks and posts the generated sub-tasks on the task blackboard. The other executes the task itself and writes its results on the data blackboard.

B. *Workflow*

We use a workflow template to facilitate making a workflow. The following is the basic workflow of a knowledge base construction. The job owner posts a task to solicit a piece of knowledge about a particular topic. For example, if the target of the knowledge base is e-learning content, a posted task might create an exercise in the target domain. To maintain the quality of the created knowledge base, the contents must be checked and revised. We represent such a workflow with the template shown in Fig. 3. When the specific task to be executed is identified, the workflow template is instantiated and executed.

Consider another example of creating multilingual e-learning contents. The domain ontology can be divided into layers. The upper layer ontology describes the vocabulary that is common to e-learning content creation. The multilingual aspect is also described as a workflow for translating the content using a human or a machine service. The target domain

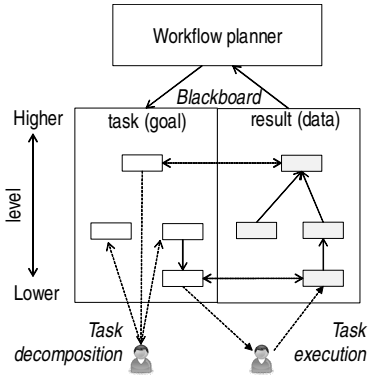

Figure 2. Using a blackboard metaphor

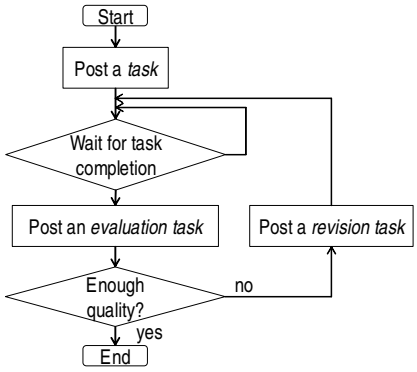

Figure 3. Workflow template for revising task results

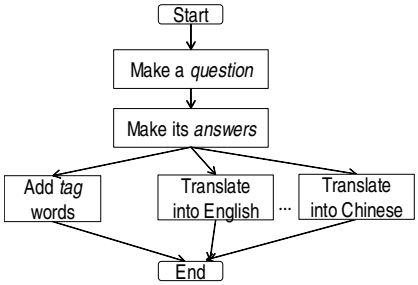

Figure 4. Workflow template for multilingual FAQs

of the e-learning environment is described as a lower layer of the domain ontology. For example, if the target e-learning content is about *artificial intelligence*, the topics that must be covered in this domain, such as *search algorithm*, will be described in the lower domain ontology. Tasks to create content for each topic are made using this domain ontology. As a result of the generated task, another task is also created for checking the generated contents.

III. Example Applications

Next, we consider two example applications: one creates a knowledge base of multilingual frequently asked questions (FAQs) in a domain of rental apartments [8], and the other creates content in an e-learning environment.

A. Multilingual FAQ system

In this application, the knowledge base contains questions that are often asked by international students living in Japan and answers in four languages. This application is intended to provide useful information to international students. Currently, the FAQs are stored in a linked data format. It is preferable that more FAQs are collected and stored in the system. Thus, it is necessary to provide a way to add to the FAQs an entry that consists of a question and its possible answers. This job can be divided into three sub-tasks: 1) adding a question, 2) adding its answer, and 3) adding translations. Its workflow template can be represented as shown in Fig. 4.

To cover as wide a domain area as possible, the job owner may want to make a task that obtains a question regarding a specific topic. In such a case, the workflow is instantiated to solicit a question and its answers in the specified topic and finally their translations in the target languages.

B. Content creation in the e-learning environment

As another example, we consider a task that creates content in an e-learning environment. In the proposed framework, a knowledge base is implemented as linked data. Thus, its contents are represented using a Resource Description Framework (RDF) [17]. In a RDF, information is represented as a set of triples, each of which consists of a subject, a predicate, and an object. In the following, we create a task that adds new exercises as e-learning content.

Assume that the target domain of the e-learning environment is *search algorithms*. An example domain ontology that describes the relationships among search algorithms can be represented, as shown in Fig. 5. The ontology itself is also represented in the RDF format.

A typical exercise can be represented, as shown in Fig. 6.

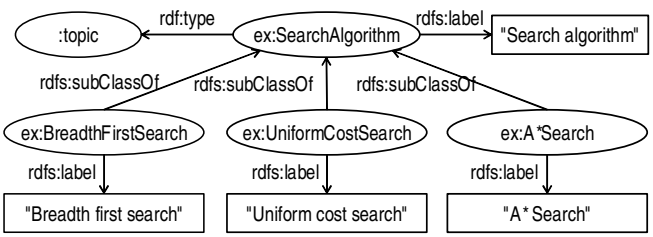

Figure 5. Example domain ontology

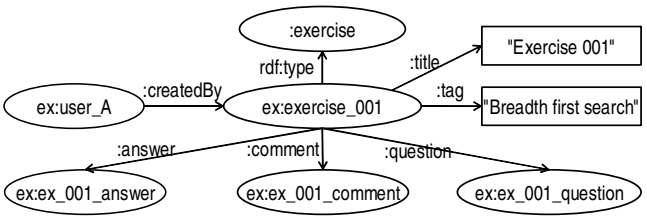

Figure 6. Example structure of exercise

This example represents an exercise that is related with *Breadth First Search*, as specified by its tag.

Next, consider a situation where some topics are not covered by existing exercises. First, such topics are specifically searched for using a SPARQL [19] query to the RDF database that stores the e-learning content. Then, if such a topic is found, a task is generated to create an exercise to cover it. Additionally, a task to revise the created exercise is also generated. These steps are determined by dividing a task of *making an exercise* into subtasks according to the workflow template (Fig. 7). The divided subtasks are executed in turn. The subtasks include a task executed by a machine (finding a topic, in this example) and a human intelligence task (making an exercise itself).

IV. Related Works and Discussion

A. Workflow execution

In crowdsourcing, a job is basically decomposed into small independent tasks that are assigned to individuals. Task decomposition itself can be performed by crowdsourcing. For example, a tool called *Turkomatic* collaboratively makes workflows with crowd workers and job requesters [11]. A method for dynamically controlling a crowdsourcing process is also proposed using the model of active rules [2]. In the proposed

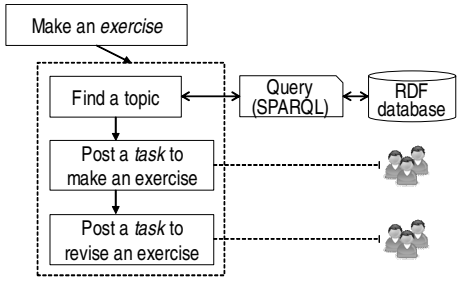

Figure 7. Dividing a task of making an exercise

platform, based on the blackboard model, we introduce two blackboards: one holds tasks and another separately holds task results. We plan to opportunistically execute task decomposition and task execution to adapt changes during a long job execution. This will be necessary for jobs like knowledge base construction that require incremental and possibly never-ending processes.

The crowdsourcing approach is used to build an ontology (for example, [4]), and we also apply crowdsourcing to revise the ontology of the target domain itself. By combining the process of the domain ontology revision and knowledge base construction, we can achieve greater flexibility.

As for a workflow's execution model, handling an exception that may occur explicitly is preferable. Human task execution is often error-prone and may fail. We plan to introduce a *meta-level* workflow to handle task failures so that the main workflow can be created by focusing on the execution of the task itself without paying attention to exception handling.

B. Distribution of rewards

A mechanism must be designed that rewards users for their contributions to the knowledge base. When a monetary reward is involved, determining the value for each task becomes an issue. To solve this problem, we will apply the concept of cooperative games [14]. Some methods have been proposed to distribute rewards in Internet environments where such malicious manipulations as false names or collusion are possible [15], [20]. It is a future issue to implement a mechanism that ensures the incentive of crowd workers, as a function of the reward manager in the proposed platform.

C. Monitoring knowledge base construction

We must also monitor how content is accumulated in knowledge base construction. A visualization mechanism is helpful for such a purpose. For example, CrowdScape [18] controls the *crowd* by visualizing the behaviors of workers. As for workflow execution, CrowdWeaver [10] visually manages complex crowd work.

In our proposed framework, we plan to exploit the domain ontology to visualize the progress of knowledge base content accumulation. For example, we plan to design a function to show how many topics are covered in the target domain by clarifying the correspondence between the contents and the topics described in the domain ontology. In this way, it is possible to provide a job owner with a feedback on the progress of the crowdsourced works. The job owner can intervene in the knowledge base construction, if necessary, to ensure the quality of the knowledge base.

V. Conclusion and Future Work

This paper described a crowdsourcing platform for the construction of a knowledge base and discussed its required functions. The domain ontology of the target domain of the knowledge base is utilized in preparing workflows and executing them. Adopting the blackboard metaphor allows the workflows to be executed opportunistically.

Currently, we are designing our platform as a web application, and are writing workflow templates for constructing knowledge bases in the example domains discussed in this paper. We plan to evaluate the proposed approach using this platform from the viewpoint of making use of the domain ontology.

References

[1] 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, pp. 1–22, 2009.

[2] A. Bozzon, M. Brambilla, S. Ceri, and A. Mauri, "Reactive crowdsourcing," Proc. the 22nd Int. Conf. on World Wide Web (WWW '13), May 2013, pp. 153–164.

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

[4] K. Eckert et al., "Crowdsourcing the assembly of concept hierarchies," Proc. of the 10th Annual Joint Conf. on Digital Libraries (JCDL'10), June 2010, pp. 139–148.

[5] J. Howe, Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business. Crown Business, 2009.

[6] D. Ings, et al. (eds), "WS-BPEL extension for people (BPEL4People) specification version 1.1," [Online] Available: http://docs.oasis-open.org/bpel4people/bpel4people-1.1.html, Aug. 2010 (accessed Jan. 15, 2014).

[7] V. Jagannathan, R. Dodhiawala, and L. S. Baum (eds), Blackboard Architectures and Applications. Academic Press, 1989.

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

[9] A. Kittur, B. Smus, S. Khamkar, and R. E. Kraut, "CrowdForge: crowdsourcing complex work," Proc. of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11), Oct. 2011, pp. 43–52.

[10] A. Kittur, S. Khamkar, P. André, and R. Kraut, "CrowdWeaver: Visually managing complex crowd work," Proc. of the ACM 2012 Conf. on Computer Supported Cooperative Work (CSCW '12), Feb. 2012, pp. 1033–1036.

[11] A. Kulkarni, M. Can, and B. Hartmann, "Collaboratively crowdsourcing workflows with Turkomatic," Proc. of the ACM 2012 Conf. on Computer Supported Cooperative Work (CSCW '12), Feb. 2012, pp. 1003–1012.

[12] V. R. Lesser and D. G. Corkill, "The distributed vehicle monitoring testbed: A tool for investigating distributed problem solving network," AI Magazine, vol. 4, no. 3, pp. 15–33, 1983.

[13] G. Little, L. B. Chilton, M. Goldman, and R. C. Miller, "TurKit: human computation algorithms on Mechanical Turk," Proc. of the 23nd Annual ACM Symposium on User Interface Software and Technology (UIST '10), Oct. 2010, pp. 57–66.

[14] J. von Neumann and O. Morgenstern, Theory of Games and Economic Behavior. Princeton University Press, 1944.

[15] N. Ohta, V. Conitzer, Y. Satoh, A. Iwasaki, and M. Yokoo, "Anonymity-proof shapley value: Extending shapley value for coalitional games in open environments," Proc. of the 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008), May 2008, pp. 927–934.

[16] J. Pedersen, et al., "Conceptual foundations of crowdsourcing: A review of IS research," 46th Hawaii Int. Conf. on System Sciences (HICSS), Jan. 2013, pp. 579–588.

[17] RDF Working Group, W3C, "Resource Description Framework (RDF)," [Online] Available: http://www.w3.org/RDF/, Feb. 2004 (accessed Jan. 15, 2014).

[18] J. Rzeszotarski and A. Kittur, "CrowdScape: interactively visualizing user behavior and output," Proc. of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12), Oct. 2012, pp. 55–62.

[19] SPARQL Working Group, W3C, "SPARQL 1.1 Overview," [Online] Available: http://www.w3.org/TR/sparql11-overview/, Mar. 2013 (accessed Jan. 15, 2014).

[20] M. Yokoo, V. Conitzer, T. Sandholm, N. Ohta, and A. Iwasaki, "Coalitional games in open anonymous environments," Proc. of the 20th National Conf. on Artificial Intelligence (AAAI), July 2005, pp. 509–514.

[21] M.-C. Yuen, I. King, and K.-S. Leung, "A survey of crowdsourcing systems," 2011 IEEE Third Int. Conf. on Privacy, Security, Risk and Trust (PASSAT), and 2011 IEEE Third Int. Conf. on Social Computing (SocialCom), Oct. 2011, pp. 766–773.

 ISBN: 978-1-61208-329-2