International Journal of Computer (IJC)

ISSN 2307-4523 (Print & Online)

© Global Society of Scientific Research and Researchers

http://ijcjournal.org/

Ontology for Task and Quality Management in Crowdsourcing

Reham Alabduljabbar[a]*, Hmood Al-Dossari[b]

[a]*Information Technology Department, King Saud University, Riyadh, Saudi Arabia*

[b]*Information Systems Department, King Saud University, Riyadh, Saudi Arabia*

[a]*Email: ralabduljabbar@ksu.edu.sa*

[b]*Email: hzaldossari@ksu.edu.sa*

Abstract

This paper suggests an ontology for task and quality control mechanisms representation in crowdsourcing systems. The ontology is built to provide reasoning about tasks and quality control mechanisms to improve tasks and quality management in crowdsourcing. The ontology is formalized in OWL (Web Ontology Language) and implemented using Protégé. The developed ontology consists of 19 classes, 7 object properties, and 32 data properties. The development methodology of the ontology involves three phases including Specification (identifying scope, purpose and competency questions), Conceptualization (data dictionary, UML, and instance creation), and finally Implementation and Evaluation.

Keywords: Crowdsourcing; Quality control; Task ontology; Ontology engineering; OWL.

1. Introduction

The Humans may outperform machines in accomplishing some tasks that require very basic human skills, especially those tasks related to creativity, natural language processing, and image understanding [1]. Recently, crowdsourcing becomes a promising methodology to overcome such problems that remain difficult or impossible to automate. Crowdsourcing introduces a new way for organizations and individuals to leverage the humans' knowledge and intelligence towards accomplishing special tasks that are difficult to fulfill effectively with machines alone. For example, the crowd may be invited to tag a photo, translate written text, transcribe an audio file or perform usability testing. This can help organizations to extend their resources, improve productivity while reducing costs and minimizing time[2].

----

* Corresponding author.

Despite the growing interest and benefits of crowdsourcing, the quality control remains a valid concern[3]. In our review of the crowdsourcing literature [4], we have observed the following shortcomings: First, task and quality management are missing from current systems. Specifically, identification of which quality control mechanisms (QCM) are most appropriate for which types of tasks is not supported. That is, as illustrated by [5], a QCM that works well for some tasks might work poorly on another. For instance, a QCM that is suitable for evaluating an essay-writing task is not suitable for evaluating a transcription task. Second, there is no support for tasks similarities. Finally, there is a huge number of tasks existing in crowdsourcing systems in unstructured nature and in natural language, making it difficult for clustering or classifying tasks to improve and learn from such systems.

Crowdsourcing data should be organized in such a way that makes it understandable for both humans and machines and allows for standardization across different crowdsourcing systems. Usually, this task can be successfully done using ontologies. Ontologies are common representation models used for knowledge representation, distribution and reusing [6].

Hence, crowdsourcing can be semantically enriched to make the process of task and quality management more efficient. Incorporating ontological definitions of tasks and QCMs in crowdsourcing systems will provide many benefits. First, with such profiling, it is easier to reuse and automate the process of identification of the type of task. Second, a much clearer and consistent data could be obtained allowing a machine to infer new information about the data logically. Third, semantic entities will support the configuration of crowdsourcing tasks based on historical data and previous rating of QCMs. That is, based on a task performance with respect to certain QCM, this configuration can be reused.

To that end, and due to the vast variety of tasks on these systems, we proposed a task ontology-based model to be adopted by crowdsourcing systems with the goal of identifying which quality control mechanisms are most appropriate for which types of tasks [7]. The focus of this work is to build the ontology of tasks mentioned above which will act as a conceptual backbone for selecting a QCM to be used with a certain requested task.

The rest of the paper is organized as follows. Section 2, presents in details the methodology we followed and the steps we took to develop the ontology. Section 3, discusses the implementation of the ontology. Evaluation of the ontology is given in Section 4. Section 5, presents related works. Finally, the conclusions are given along with some remarks on the future directions of this work in Section 6.

2. Ontology Development Methodology

There is no single correct ontology-design methodology, and many methodologies have been proposed to design and build ontologies. The approach in [8] is followed to produce this ontology; the authors of [8] have adopted different methodologies including [9, 10, 11] to define a domain ontology. In general, they divided the development process into three main phases: specification, conceptualization and implementation as shown in Figure 1. The objective of the specification phase is to obtain the knowledge about the domain while the objective of the conceptualization phase is to organize and structure this knowledge using external representations [8].

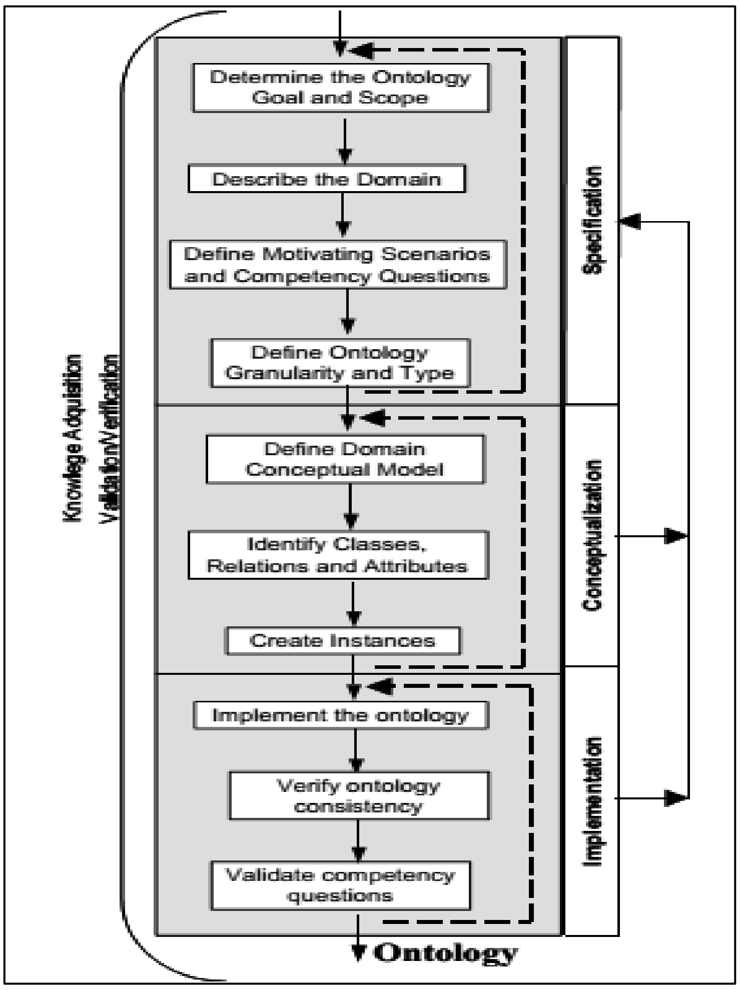

Determine the Ontology Goal and Scope
Describe the Domain
Define Motivating Scenarios and Competency Questions
Define Ontology Granularity and Type
Define Domain Conceptual Model
Identify Classes, Relations and Attributes
Create Instances
Implement the ontology
Verify ontology consistency
Validate competency questions
Ontology
Specification
Conceptualization
Implementation
Knowlege Acquisition
Validation/Verification

Figure 1: Ontology development methodology [8]

2.1 Specification: the ontology domain, purpose, and scope

As a first step, it is essential to define the domain and scope of the ontology. The scope bounds the ontology by indicating what to include and what to not. In our work, we aim to design and develop an ontology for crowdsourcing tasks and quality control mechanisms.

The ontology facilitates the structured capturing of the tasks and quality control mechanisms in an organized and meaningful way. Moreover, the ontology can be utilized by crowdsourcing systems to identify which QCM is most appropriate for a given task. Hence, this ontology considers the needs for creating concepts related to crowdsourcing tasks and QCMs. It does not consider the concepts related to other activities as task allocation, worker selections, and rewarding mechanisms.

2.2 Specification: considering reusing existing ontologies

Before starting the process of ontology modeling and building, it is important to check whether there exists any ontology appropriate for the purpose of our domain. Existing ontologies can be found in ontology libraries or by using semantic search engines. The following ontology libraries and search engines were examined in this work: Reference [20] ontology library, the DAML [21] ontology library, the Protégé [22] ontologies and the semantic search engine [23].

A lightweight ontology for Enterprise crowdsourcing was found to have some relation to our work [12]. However, the author of the Enterprise crowdsourcing ontology suggested that more investigations are needed on the part of the crowdsourcing task evaluation mechanism. In the Enterprise crowdsourcing ontology, the author developed a domain ontology for a special type of crowdsourcing task within an enterprise. We reuse and extend some parts of the Enterprise crowdsourcing ontology and adapt it fit our purpose. Also, we developed more general task crowdsourcing ontology by providing an ontological representation of tasks and quality control mechanisms.

2.3 Specification: competency questions

An important step in the specification phase is to write several competency questions (QC). Competency questions sharpen the understanding of the purpose and scope of the ontology [11]. These competency questions are questions that should be answered by the knowledge base built from the designed ontology. As an example, some of them are shown in Table 1:

Table 1: Examples of QC in the ontology

QC No.	QC
QC 1	Given a set of task attributes: (e.g. TaskType: CC, Action type: classification, Object type: image), what is the best QCM for evaluating it?
QC 2	Based on the analysis of historical data, which QCM is used for evaluating a certain type of tasks? For example: Which QCM has been used mostly to evaluate tasks of type CC?
QC 3	What tasks have similar or identical attributes to a certain task?
QC 4	What are the used Action Types for a certain Task Type in the system?
QC 5	What QCMs have already been used to evaluate a previously submitted task with similar or identical characteristics of a newly submitted task?
QC 6	What is the average rating for a given QCM with a certain task type?
QC 7	What is the most popular QCM being used with certain object type?

2.4 Specification: ontology granularity and type

Given the level of conceptualization and granularity [9] and the level of expressiveness [13], the ontology proposed here is a domain ontology with a light-weight level of expressiveness. Domain ontologies represent concepts that are particular for a certain domain, and lightweight ontologies provide small concepts and the relations between them.

2.5 Conceptualization: domain conceptual model and identification of classes, relations, and attributes

All concepts, relations, and attributes related to crowdsourcing tasks and QCMs were acquired from an extensive study of literature in the field of crowdsourcing. We also scanned several crowdsourcing systems such as Amazon Mechanical Turk [15,16,17] to get an idea of the domain. The methodology used to identify elements for defining the crowdsourcing task, and the QCMs from the literature follows two stages: First, the crowdsourcing definitions were reviewed to identify any defining element of the crowdsourcing task. Second, we conducted a review of the literature to find documentations that include crowdsourcing taxonomies[4]. In preliminary work, we have proposed a categorization of tasks that are currently used in crowdsourcing systems. Five types were proposed namely: Opinion-Based(OB), Data Processing (DP), Content Generation (CG), Content Conversion (CC) and Research-Based (RB). Further details can be found in [4]. To provide an advanced understanding of the domain, a data dictionary of the collected terms is created as a first intermediate representation. Then, it is converted to a conceptual model in the form of Unified Modeling Language (UML) diagram. Table 2 includes part of the data dictionary including descriptions of some concepts, relations, and attributes in the ontology.

From the data dictionary and to properly understand the conceptual aspects in the domain, a UML diagram was designed with the main relations among defined concepts. Figure 2, shows a fragment of the UML diagram.

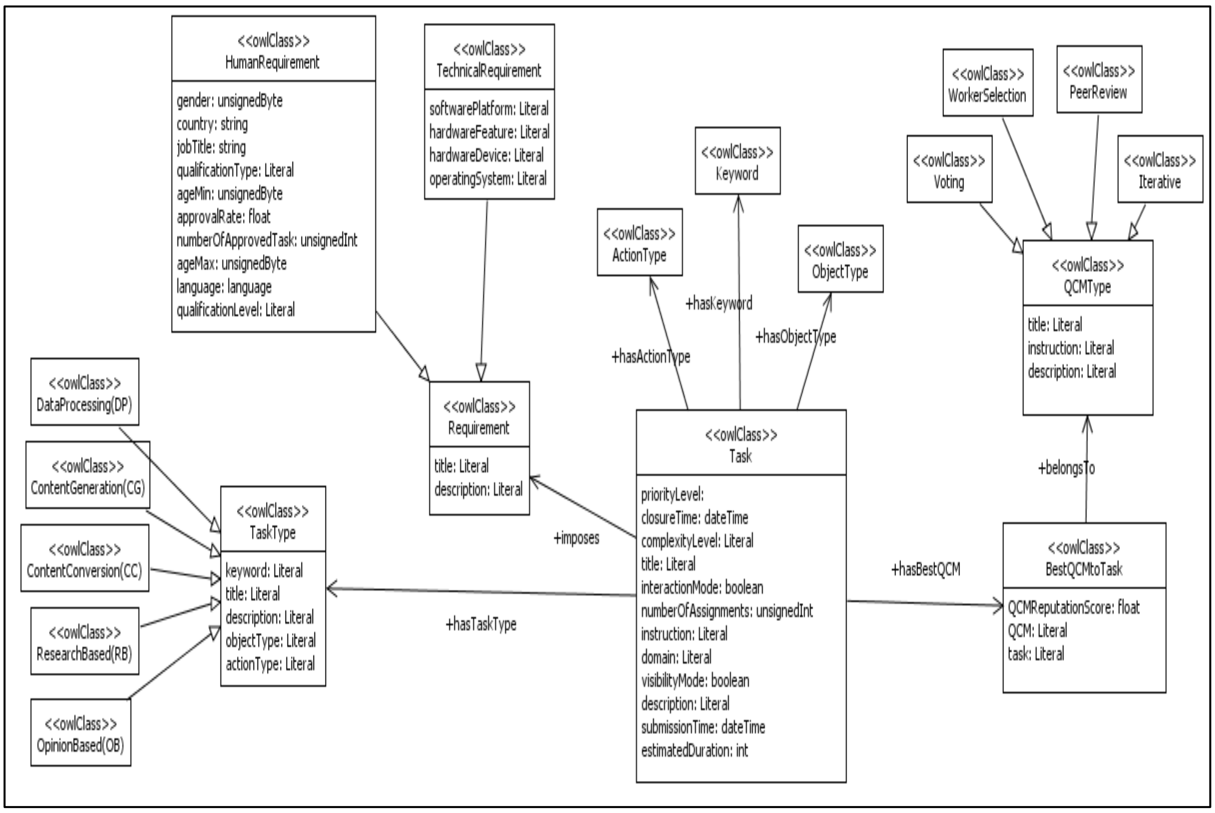


Figure 2: Ontology UML diagram

Table 2: Part of the data dictionary of the crowdsourcing task ontology

Name	Description	Type
Task	The core concept in the ontology, which represent one single task requested by a user and need to be solved by workers.	concept
TaskType	The type of crowdsourcing task which can be (OB, CG, DP, CC, RB).	concept
OpinionBased (OB)	Opinion-Based type of tasks.	concept
ContentGeneration (CG)	Content Generation type of tasks.	concept
DataProcessing (DP)	Data Processing type of tasks.	concept
ContentConversion (CC)	Content Conversion type of tasks.	concept
ResearchBased (RB)	Research-Based type of tasks.	concept
QCMType	The concept that holds the QCMs and can be of many types including (worker selection, iterative, voting and peer review)	concept
WorkerSelection	A QCM type where tasks are only allowed to be solved by a specified crowd upon the requesters’ choice.	concept
Iterative	A QCM type where the solution goes into an iterative process before sending it to requesters. That is a second worker improves the results of the initial submission, and a third worker improves the results of the second worker so on.	concept
Voting	A QCM type where the judgment of a majority of workers on the task solution is accepted.	concept
PeerReview	A QCM type where other workers can check and evaluate the solution.	concept
BestQCMtoTask	The class that holds the best QCM for each existing task in the ontology.	concept
Keyword	This class holds a list of keywords associated with the task.	concept
ActionType	This class holds the type of actions associated with tasks. An action type can be classification, summarization, tagging, etc.	concept
ObjectType	Holds the data type of the objects in the task. An object type can be image, text, audio, etc.	concept
QCMReputationScore	Holds the reputation score of the best suitable QCM for a certain task.	attribute
hasTaskType(Task, TaskType)	To represent the relation that each task has a task type.	relation
hasBestQCM (Task, BestQCMtoTask)	To represent the relation between certain task and the best QCM for evaluating it.	relation
belongsTo(BestQCMtoTask, QCMType)	To represent that a Best QCM belongs to the QCMType class	relation

The Task Class shows that a Task can be defined by many features including its Keywords, Task Type, Object Type and Action Type. As said earlier, Task Type class can be of types: OB, DP, CG, CC or RB. QCM Type class can be of types: Voting, Worker Selection, Peer Review and Iterative. These QCMs are the mostly used QCMs in the literature [4]. Bothe Task Types and QCM Types can be extended, and new types can be added.

The task ontology has a simple hierarchy of concepts as presented in the 'is-a' diagram in Figure 3.

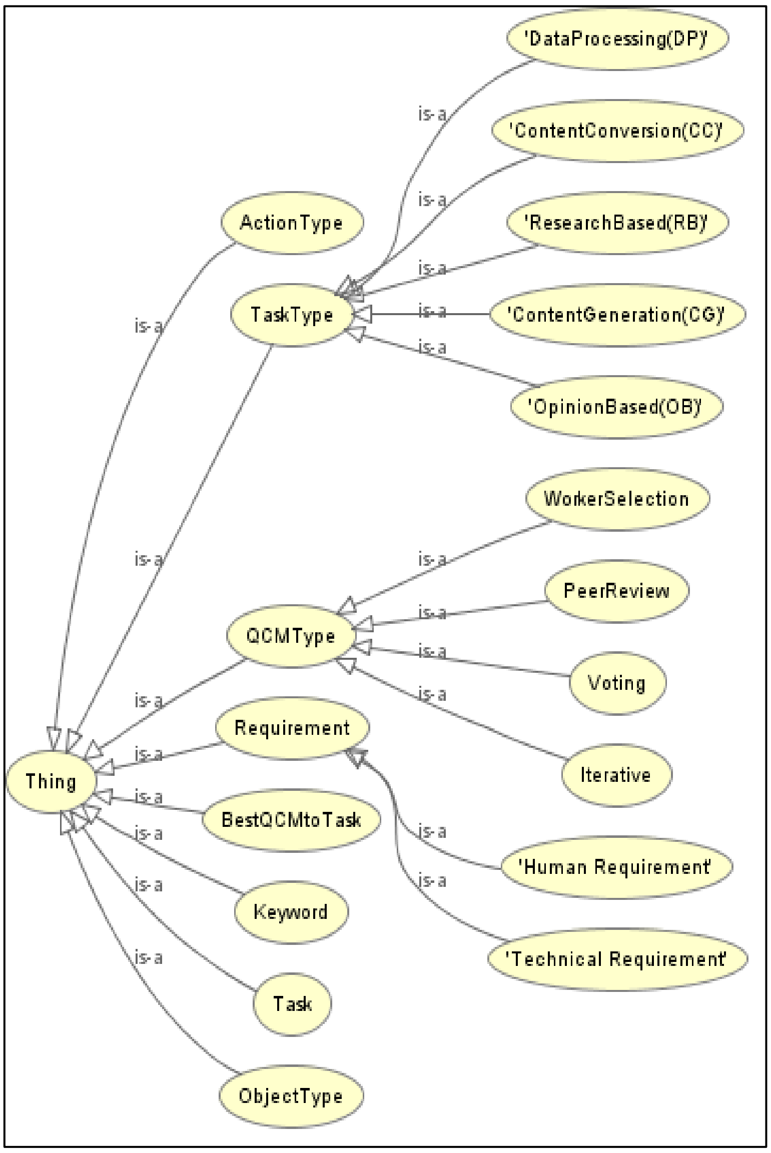


Figure 3: "is-a" diagram

2.6 Conceptualization: Instance definition

The last step in ontology modeling is to create individual instances of classes in the ontology. Each instance should be provided a definition of its name, the name of the concept it belongs to, and its attribute values if known [8]. Table 3, provides some instances of the instance table of our ontology.

Table 3: An excerpt of the instance table of our ontology

Instance	Concept	Attribute	Value
CC_124	CC	title	CC
		keyword	Transcribe, audio
		actionType	transcription
		objectType	Audio
Voting_146	Voting	title	Majority voting
		description	A QCM type where the judgment of a majority of workers on the task solution is accepted.
Task_111	Task	TaskType	CC
		ActionType	Describe
		ObjectType	Image
		Requirement	English, age above 20
		Keyword	Describe, image
Task_112	Task	TaskType	CC
		ActionType	transcription
		ObjectType	Audio
		Requirement	Arabic, age above 30, IT
		Keyword	Transcribe, audio
BestQCMtoTask_672	BestQCMtoTask	task	Task_111
		QCM	Voting_146
		QCMReputationScore	8.4
BestQCMtoTask _785	BestQCMtoTask	task	Task_112
		QCM	PeerReview_987
		QCMReputationScore	7.1

3. Ontology Implementation

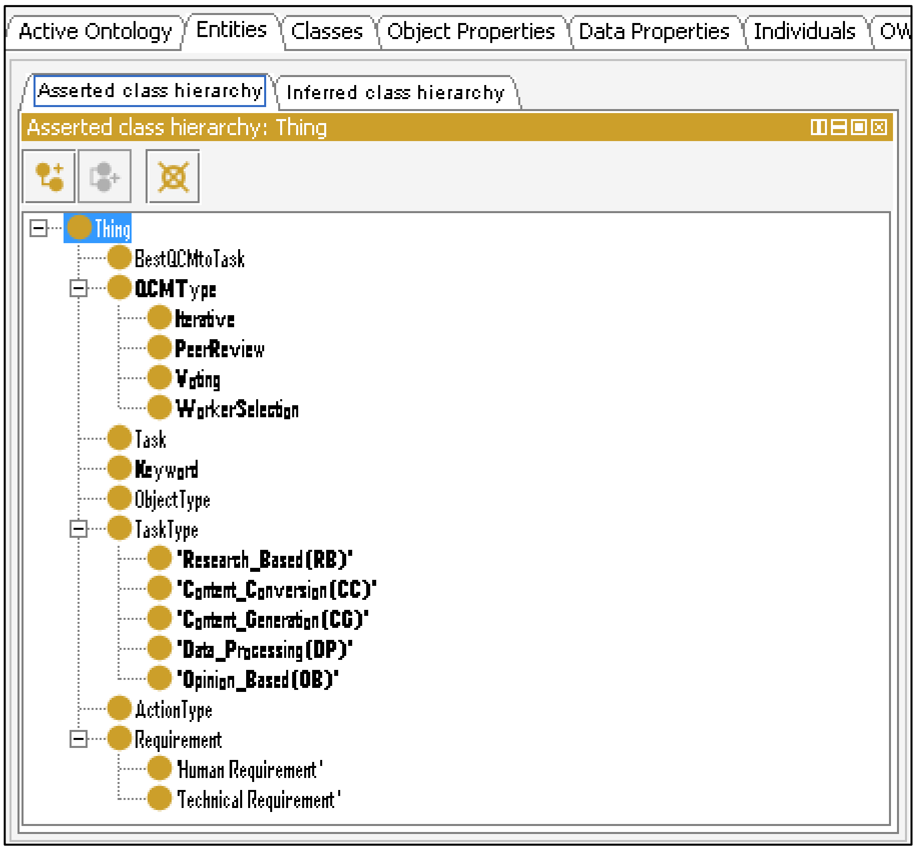
To bring the proposed ontology to reality, some development tools were used.

3.1 Web Ontology Language (OWL)

The description of the knowledge in the ontology requires explicit formal representation. The OWL (Web Ontology Language) descriptive language was used to specify formally how the knowledge will be stored. We chose OWL for its expressiveness and powerful representation. Also, OWL is a World Wide Web Consortium (W3C) standard ontology language and recommended for representing ontologies.

3.2 Protégé Editor

The implementation was done using a well-known ontology editor called Protégé. Protégé [18] is a free, open-source ontology editor and knowledge base framework developed at the Stanford Medical Informatics. Protégé ontologies can be exported into different formats including RDF Schema (RDFS) and Web Ontology Language (OWL). Protégé also provides the ability to check the consistency, validation, and verification of an ontology. Figure 4, depicts Protégé snapshot to show some of our ontology's entities.



```
Thing
    BestQCMtoTask
    QCMType
        Iterative
        PeerReview
        Voting
        WorkerSelection
    Task
    Keyword
    ObjectType
    TaskType
        'Research_Based(RB)'
        'Content_Conversion(CC)'
        'Content_Generation(CG)'
        'Data_Processing(DP)'
        'Opinion_Based(OB)'
    ActionType
    Requirement
        'Human Requirement'
        'Technical Requirement'
```

Figure 4: Protégé snapshot of the ontology entities.

3.3 Jena Library

To read ontology data that are stored in the OWL file, the Jena library was used [24]. This open-source library, written in the Java programming language, provides an API to load, save, query, and infer new information from the ontology.

The developed ontology consists of 19 classes, 7 object properties, and 32 data properties. In addition, some individuals were created for evaluation purposes, which can be extended later.

4. Ontology Evaluation

The evaluation of the ontology is demonstrated concerning the QC stated in Section 2.3 to check the convergence of the ontology to the CQ. This evaluation considers if the ontology solves the problem it was supposed to solve. A group of methods was implemented for the QC or requirements that were identified to enable the validation of the basic functionalities of the ontology. An overview containing the goals from these methods can be found below:

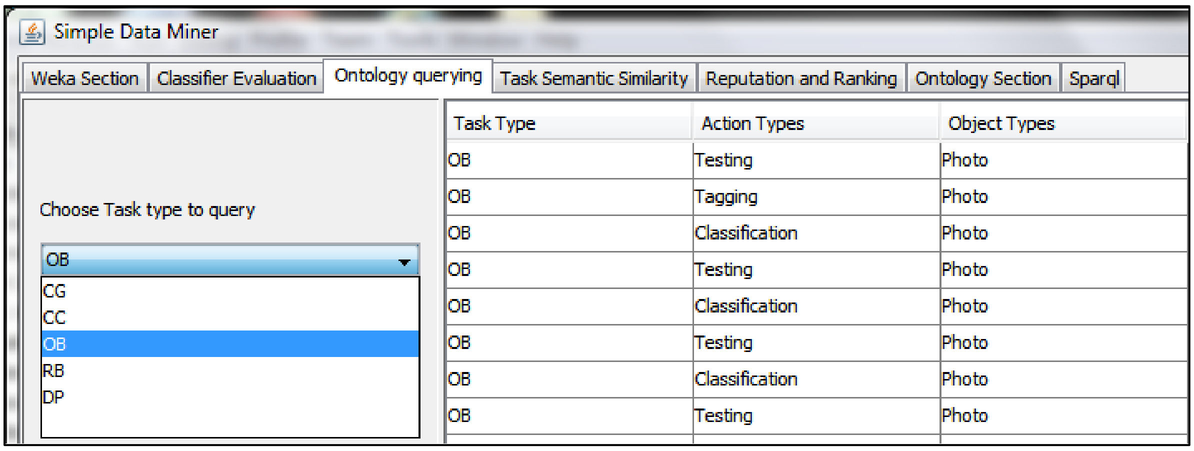
1. ***GetActionTypeList(TaskType):*** lists all action types for a certain task type.
2. ***GetObjectTypeList(TaskType):*** lists all object types for a certain task type.
3. ***GetTaskRequirement(TaskType, ActionType, ObjectType):*** retrieves the task requirements given the task type, action type, and object type.
4. ***GetSimiralTask(TaskType, ActionType,ObjectType***): uses semantic similarity measures to retrieve similar tasks to the requested one.
5. ***GetQCMReputation(QCM, TaskType):*** returns the reputation of a QCM with the requested task type.
6. ***GetTopQCM(TaskType, ActionType, ObjectType):*** is responsible for returning the QCM with the highest reputation for evaluating the requested task with its features.
7. ***GetTopQCM_TaskType(TaskType):*** is responsible for returning the QCM with the highest reputation for evaluating the requested task type.
8. ***GetTopQCM_ObjectType(ObjectType):*** is responsible for returning the QCM with the highest reputation begin used with the requested object type.

A prototype is developed to enable the execution of these methods. After executing these tests successfully, we used the QC and the methods implemented to verify if the basic functionalities of the ontology were accomplished. Table 4 lists this relationship. With this relationship between QC and the implemented methods and the results of testing, we conclude that the proposed ontology solves the problem it was supposed to and is adequate for the representation of tasks in crowdsourcing systems.

Table 4: QCs and related methods relationship

QC no.	Related Method
QC1	GetTopQCM(TaskType, ActionType, ObjectType)
QC2	GetTopQCM_TaskType(TaskType)
QC3	GetSimiralTask(TaskType, ActionType, ObjectType)
QC4	GetActionTypeList(TaskType)
QC5	GetSimiralTask(TaskType, ActionType, ObjectType)
	GetTopQCM(TaskType, ActionType, ObjectType)
QC6	GetQCMReputation(QCM, TaskType)
QC7	GetTopQCM_ObjectType(ObjectType)

Figure 5, provides a snapshot of the developed prototype, and it shows when the Opinion Based (OB) Task type is chosen, the previously used Action Types and Object Types are retrieved.



Task Type	Action Types	Object Types
OB	Testing	Photo
OB	Tagging	Photo
OB	Classification	Photo
OB	Testing	Photo
OB	Classification	Photo
OB	Testing	Photo
OB	Classification	Photo
OB	Testing	Photo

Figure 5: Snapshot of the developed prototype to query our ontology.

5. Related Works

This section describes previous work that uses ontologies for crowdsourcing activity representation. The most relevant work to ours is that proposed in [19] which is a skill ontology-based model for quality assurance. While the skill ontology-based model was used to identify and match the best worker to a given task, our task ontology-based model aims to identify and match the best-suited quality control mechanism to a given task. In our work, we argue that identifying the best worker is part of the whole process of the quality control mechanism. In other words, the skill of the worker will affect the performance of the quality control mechanism and will be implied in the given ratings by the requesters.

As said in Section 2.2, a lightweight ontology for Enterprise Crowdsourcing was proposed in [12]. However, it is a domain-specific ontology for a special type of crowdsourcing task within an enterprise.

Our work differs from the above because the focus is in the ontological representation of tasks and quality control mechanisms as well as identifying the best-suited quality control mechanism to a given task.

6. Conclusion and Future Work

This paper presented ontology of tasks and quality control mechanisms in the crowdsourcing domain. We explained how the concepts, properties, and relations were defined in our ontology. The ultimate goal is to use this ontology to improve task and quality management in crowdsourcing. The implementation of the ontology was done using the Protégé editor and the Jena library. The resulted ontology consists of 19 classes, 7 object properties, and 32 data properties.

This work is part of a greater model involving machine learning techniques and reputation systems to improve the overall quality of the crowdsourcing. Future work will concentrate on utilizing the ontology in the model proposed in [7].

References

[1] L. Ponciano, F. Brasileiro, N. Andrade, and L. Sampaio, "Considering human aspects on strategies for designing and managing distributed human computation," Journal of Internet Services and Applications, vol. 5, no. 1, p. 10, 2014.

[2] L. Litman, J. Robinson, and C. Rosenzweig, "The relationship between motivation, monetary compensation, and data quality among US- and India-based workers on Mechanical Turk.," Behavior research methods, Springer US, Jun. 2014.

[3] R. Buettner, "A Systematic Literature Review of Crowdsourcing Research from a Human Resource Management Perspective," in Proceedings of the 48th Hawaii International Conference on System Sciences, 2015.

[4] R. Alabdujabbar and H. Al-Dossari, "Towards a Classification Model for Tasks in Crowdsourcing," in the ACM proceedings of the second International Conference on Internet of Things, Data and Cloud Computing (ICC 2017), Cambridge city, United Kingdom. (In Press).

[5] M. Allahbakhsh, B. Benatallah, A. Ignjatovic, H. Motahari-Nezhad, E. Bertino, and S. Dustdar, "Quality Control in Crowdsourcing Systems: Issues and Directions," IEEE Internet Computing, vol. 17, no. 2, pp. 76–81, 2013.

[6] M. Gan, X. Dou, and R. Jiang, "From ontology to semantic similarity: Calculation of ontology-based semantic similarity," The Scientific World Journal, 2013.

[7] [7] R. Alabduljabbar and H. Al-Dossari, "A Dynamic Model for Quality Control in Crowdsourcing Systems," in Proceedings of the IKE'16 - The 15th International Conference on Information & Knowledge Engineering,July 2016, Las Vegas, USA.

[8] G. Brusa, M. Caliusco, and O. Chiotti, "A process for building a domain ontology: an experience in developing a government budgetary ontology," Proceedings of the second Australasian workshop on Advances in ontologies, vol. 72, no. c, pp. 7–15, 2006.

[9] A. Gómez-Pérez, M. Fernández-López, and O. Corcho, Ontological engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. London: Springer, 2004.

[10] M. Gruninger and M. S. Fox, "Methodology for the Design and Evaluation of Ontologies," Industrial Engineering, vol. 95, pp. 1–10, 1995.

[11] M. Uschold and M. Gruninger, "Ontologies: principles, methods and applications," The Knowledge Engineering Review, vol. 11, no. February, pp. 93–136, 1996.

[12] L. Hetmank, "A Lightweight Ontology for Enterprise Crowdsourcing," in Proceedings of the 22nd

European Conference on Information Systems (ECIS 2014), 2014, p. Paper 886.

[13] J. DE Bruijn and D. Fensel, “Ontology Definitions,” in Encyclopedia of Library and Information Science, Marcel Dekker, inc., 2005.

[14] M. Hepp, “Possible ontologies: How reality constrains the development of relevant ontologies,” IEEE Internet Computing, vol. 11, no. 1, pp. 90–96, 2007.

[15] MTurk, “MTurk: Amazon Mechanical Turk,” 2016. [Online]. Available: http://www.mturk.com/. [Accessed: 01-Jan-2016].

[16] Upwork, “Upwork,” 2016. [Online]. Available: https://www.upwork.com/. [Accessed: 24-Mar-2016].

[17] Freelancer, “Freelancer,” 2016. [Online]. Available: https://www.freelancer.com/. [Accessed: 24-Mar-2016].

[18] “Protégé.” [Online]. Available: http://protege.stanford.edu. [Accessed: 01-May-2016].

[19] K. El Maarry, W. Balke, H. Cho, S. Hwang, and Y. Baba, “Skill Ontology-Based Model for Quality Assurance in Crowdsourcing,” Database Systems for Advanced Applications, LNCS, vol. 8505, pp. 376–387, 2014.

[20] http://www.ksl.stanford.edu/software/ontolingua/ , Access: 10 June 2016.

[21] http://www.daml.org/ontologies/ , Access: 10 June 2016.

[22] http://protege.cim3.net/cgi-bin/wiki.pl?ProtegeOntologiesLibrary , Access: 10 June 2016.

[23] http://swoogle.umbc.edu/ , Access: 10 June 2016.

[24] http://jena.sourceforge.net/ , Access: 10 June 2016.