TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Wirtschaftswissenschaften

Enhancing Automation and Interoperability in Enterprise Crowdsourcing Environments

Dissertation

zur Erlangung des akademischen Grades

Dr. rer. pol.

vorgelegt an der

Fakultät Wirtschaftswissenschaften

der

Technischen Universität Dresden

von

Lars Hetmank

Tag der Verteidigung:

1. September 2016

Gutachter:

Prof. Dr. rer. pol. habil. Eric Schoop

Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill

LIST OF ARTICLES

Enhancing Automation and Interoperability in Enterprise Crowdsourcing Environments (Summary)

Lars Hetmank

TABLE OF CONTENTS

LIST OF FIGURES

1 MOTIVATION

The last couple of years have seen a fascinating evolution. While the early Web predominantly focused on human consumption of Web content, the widespread dissemination of social software and Web 2.0 technologies enabled new forms of collaborative content creation and problem solving. These new forms often utilize the principles of collective intelligence, a phenomenon that emerges from a group of people who either cooperate or compete with each other to create a result that is better or more intelligent than any individual result (Leimeister, 2010; Malone, Laubacher, & Dellarocas, 2010). Crowdsourcing has recently gained attention as one of the mechanisms that taps into the power of web-enabled collective intelligence (Howe, 2008). Brabham (2013) defines it as "an online, distributed problem-solving and production model that leverages the collective intelligence of online communities to serve specific organizational goals" (p. xix). Well-known examples of crowdsourcing platforms are Wikipedia,[1] Amazon Mechanical Turk,[2] or InnoCentive.[3]

Since the emergence of the term *crowdsourcing* in 2006, one popular misconception is that crowdsourcing relies largely on an amateur crowd rather than a pool of professional skilled workers (Brabham, 2013). As this might be true for low cognitive tasks, such as tagging a picture or rating a product, it is often not true for complex problem-solving and creative tasks, such as developing a new computer algorithm or creating an impressive product design. This raises the question of how to efficiently allocate an enterprise crowdsourcing task to appropriate members of the crowd. The sheer number of crowdsourcing tasks available at crowdsourcing intermediaries makes it especially challenging for workers to identify a task that matches their skills, experiences, and knowledge (Schall, 2012, p. 2).

An explanation why the identification of appropriate expert knowledge plays a major role in crowdsourcing is partly given in Condorcet's jury theorem (Sunstein, 2008, p. 25). The theorem states that if the average participant in a binary decision process is more likely to be correct than incorrect, then as the number of participants increases, the higher the probability is that the aggregate arrives at the right answer. When assuming that a suitable participant for a task is more likely to give a correct answer or solution than an improper one, efficient task recommendation becomes crucial to improve the aggregated results in

[1] Wikipedia can be found at http://www.wikipedia.org/.
[2] Amazon Mechanical Turk can be found at https://www.mturk.com/.
[3] InnoCentive can be found at http://www.innocentive.com/.

crowdsourcing processes. Although some assumptions of the theorem, such as independent votes, binary decisions, and homogenous groups, are often unrealistic in practice, it illustrates the importance of an optimized task allocation and group formation that consider the task requirements and workers' characteristics.

Ontologies are widely applied to support semantic search and recommendation mechanisms (Middleton, De Roure, & Shadbolt, 2009). However, little research has investigated the potentials and the design of an ontology for the domain of enterprise crowdsourcing. The author of this thesis argues in favor of enhancing the automation and interoperability of an enterprise crowdsourcing environment with the introduction of a semantic vocabulary in form of an expressive but easy-to-use ontology. The deployment of a semantic vocabulary for enterprise crowdsourcing is likely to provide several technical and economic benefits for an enterprise. These benefits were the main drivers in efforts made during the research project of this thesis:

1. *Task allocation:* With the utilization of the semantics, requesters are able to form smaller task-specific crowds that perform tasks at lower costs and in less time than larger crowds. A standardized and controlled vocabulary allows requesters to communicate specific details about a crowdsourcing activity within a web page along with other existing displayed information. This has advantages for both contributors and requesters. On the one hand, contributors can easily and precisely search for tasks that correspond to their interests, experiences, skills, knowledge, and availability. On the other hand, crowdsourcing systems and intermediaries can proactively recommend crowdsourcing tasks to potential contributors (e.g., based on their social network profiles).
2. *Quality control:* Capturing and storing crowdsourcing data increases the overall transparency of the entire crowdsourcing activity and thus allows for a more sophisticated quality control. Requesters are able to check the consistency and receive appropriate support to verify and validate crowdsourcing data according to defined data types and value ranges. Before involving potential workers in a crowdsourcing task, requesters can also judge their trustworthiness based on previous accomplished tasks and hence improve the recruitment process.
3. *Task definition:* A standardized set of semantic entities supports the configuration of a crowdsourcing task. Requesters can evaluate historical crowdsourcing data to get suggestions for equal or similar crowdsourcing tasks, for example, which incentive or

evaluation mechanism to use. They may also decrease their time to configure a crowdsourcing task by reusing well-established task specifications of a particular type.

4. *Data integration and exchange:* Applying a semantic vocabulary as a standard format for describing enterprise crowdsourcing activities allows not only crowdsourcing systems inside but also crowdsourcing intermediaries outside the company to extract crowdsourcing data from other business applications, such as project management, enterprise resource planning, or social software, and use it for further processing without retyping and copying the data. Additionally, enterprise or web search engines may exploit the structured data and provide enhanced search, browsing, and navigation capabilities, for example, clustering similar crowdsourcing tasks according to the required qualifications or the offered incentives.

The remainder of this summary article is structured as follows. Section 2 introduces the research domain of the thesis. Section 3 presents the chosen research methodology including the author's theoretical position, the research strategy, and the overall research process. The structure of the thesis as well as the five articles that are part of the thesis are explained in section 4. Then, section 5 outlines the ontology for enterprise crowdsourcing as the main research contribution. Finally, the paper concludes with a brief summary, highlights the strengths and reveals current limitations of the designed artifact, and suggests aspects for further research.

2 RESEARCH DOMAIN

Before introducing the research methodology of this thesis, the subject matter under study is outlined. The subject matter under study lies within the intersection of three research areas: enterprise crowdsourcing, Semantic Web, and knowledge management.

2.1 ENTERPRISE CROWDSOURCING

Enterprise crowdsourcing as a specific type of crowdsourcing aims to outsource organizational tasks traditionally performed by predetermined employees to an large undefined group of people who come from both inside and outside the company (Jayakanthan & Sundararajan, 2011). Gassenheimer, Siguaw, and Hunter (2013) stress that compared to other types of crowdsourcing, which focus on satisfying social or scientific demands, enterprise crowdsourcing aims to serve business purposes.

The target group of an enterprise crowdsourcing activity varies depending on how critical and confidential the organizational task is (Hetmank, 2014b). It ranges from an internal crowd of employees contracted within an organization (intra-corporate crowdsourcing), over a crowd of freelancers, partners, suppliers, and customers associated with the organization (inter-corporate crowdsourcing), to a loosely coupled crowd of the general public domain (corporate crowdsourcing).

Applications of enterprise crowdsourcing that harness the power of the crowd can be largely divided into externally hosted crowdsourcing intermediaries, such as Amazon Mechanical Turk[4] (micro-task platform) or InnoCentive[5] (idea-generation and problem-solving platform), and internally hosted, often self-developed and task-specific crowdsourcing platforms, such as IBM's People Cloud (knowledge acquisition in the area of IT inventory management and IT support services) or ScribeCrowd (technical documentation) (Lopez, Vukovic, & Laredo, 2010; Vukovic, Salapura, & Rajagopal, 2013).

A considerable amount of literature has been published on challenges that enterprises face when introducing crowdsourcing in order to solve labor- or knowledge-intensive tasks (Erickson & Trauth, 2013; Maiolini & Naggi, 2011; Pedersen et al., 2013; Simula, 2013; Vukovic, 2009). These challenges fall mainly into the following categories:

- management of crowdsourcing tasks (including breaking down a task into subtasks, recommending a task to suitable users, selecting a task, and integrating a task into existing business processes),
- configuration of incentive mechanisms (including reward and pricing models),
- setup of quality assurance and control mechanisms (including feedback systems as well as methods and tools for evaluating crowdsourcing users and contributions),
- aggregation of large numbers of crowdsourcing contributions toward a common solution,
- consideration of data privacy and security concerns (including aspects of confidentiality and trustworthiness), and
- change of the organizational culture that supports crowdsourcing initiatives.

This thesis investigates the potential of the Semantic Web to address the above-mentioned challenges.

[4] Amazon Mechanical Turk can be found at https://www.mturk.com/.
[5] InnoCentive can be found at http://www.innocentive.com/.

2.2 SEMANTIC WEB

The Semantic Web as an extension of the existing Web brings semantics and structure to the content of the web pages (Berners-Lee, Hendler, & Lassila, 2002). From the beginning, the Web has advanced as a platform of documents aimed for human consumption. The Semantic Web provides guidelines, standards, tools, and methods that allow machines to process decentralized data and information encoded into web pages (Islam, Abbasi, & Shaikh, 2010). Structured data as well as a set of inference rules for conducting automated reasoning are the prerequisites for the Semantic Web to work.

The HyperText Markup Language (HTML) is still the predominant language used to create documents on the Web. However, HTML is mainly intended to specify the appearance of a web page and describes how a web browser should arrange headings, text, tables, and images. Achieving the vision of the Semantic Web requires languages for expressing machine-understandable metadata for web documents. The extensible markup language (XML) allows hierarchical structuring of data within documents albeit without specifying the actual meaning of the structure (Bray, Paoli, Sperberg-McQueen, Maler, & Yergeau, 2008). Adding meaning to the content of the Web can be realized by applying the resource description framework (RDF). RDF provides a minimalist triple-based knowledge representation language for the Web (Cyganiak, Wood, & Lanthaler, 2014). Similarly to sentence structures, RDF offers a way to encode semantics in subject-predicate-object triples. Each subject, predicate, and object is uniquely identified by a universal resource identifier (URI). URIs are the basic elements to interlink resources on the Web. There are several mechanisms to encode machine-readable data within HTML documents, for example, RDFa, microformats, or microdata (Adida, Birbeck, McCarron, & Herman, 2013; Hickson, 2013; Khare & Çelik, 2006).

Another essential building block of the Semantic Web is ontology languages that are built upon RDF and used to construct ontologies. In information systems and computer science, Studer, Benjamins, and Fensel (1998) define an ontology as "a formal, explicit specification of a shared conceptualization" of a domain of interest (p. 184), which is in this thesis the domain of enterprise crowdsourcing. With regard to the Semantic Web, ontologies provide the necessary semantic vocabularies to annotate web pages in a machine-interpretable form (Grimm, Abecker, Völker, & Studer, 2011). These so-called Semantic Web vocabularies can be encoded, for example, with the simple ontology language *RDF schema* (RDFS) (Brickley

& Guha, 2014) or with the more expressive *Web Ontology Language* (OWL) (Hitzler, Krötzsch, Parsia, Patel-Schneider, & Rudolph, 2012).

2.3 KNOWLEDGE MANAGEMENT

Knowledge management comprises a wide range of activities, such as acquiring, creating, storing, distributing, using, and maintaining knowledge of an organization (Fong & Choi, 2009). These knowledge activities affect knowledge resources that are not only manifested in codified digital objects but also in people, organizational procedures, and guidelines (Bick et al., 2012). Both knowledge activities and the involved knowledge resources are central, especially in knowledge-intensive firms where a large number of people are employed to work mainly with creative and complex tasks (Alvesson, 2004). The features of social software and the bottom-up approach of crowdsourcing offer a new way to solve these tasks efficiently and cost-effectively.

In the past, enterprises have long focused on methods and tools for knowledge storing and preservation, such as debriefing, lessons learned, microarticles, or documentation of best practice cases (Lee & Lan, 2007; Maier, 2007, p. 284). Enterprise crowdsourcing, however, follows a different approach to support knowledge management, especially in knowledge-intensive large-scale organizations that are challenged with effectively managing and exploiting employees' internal knowledge as well as external knowledge of the public domain (Skopik, Schall, & Dustdar, 2012). Enterprise crowdsourcing allows increasing the productivity of knowledge-intensive tasks by identifying rare experts and using their free resources (Jayakanthan & Sundararajan, 2011). Further, harnessing the collective intelligence and workforce during a crowdsourcing process plays an important role in advancing the organizational knowledge base and hence provides a prerequisite for organizational learning within the company (Boder, 2006). Finally, while knowledge workers are engaged in complicated and complex problem-solving activities, applying mechanisms of crowdsourcing generates new knowledge and thus encourages innovation (Ribiere & Tuggle, 2010).

3 RESEARCH METHODOLOGY

This section presents the selected research methodology including the author's theoretical position, the research strategy, and the research process. Selecting a research methodology depends not only on the problem statement and the derived research objectives but also on the researcher's *theoretical position* (Becker, Holten, Knackstedt, & Niehaves, 2003). Thus, before undertaking a research project, researchers should clarify their theoretical position, which comprises an epistemological, an ontological, and a linguistic perspective. They should also define the criteria of truth that judge the accuracy of statements and the usefulness of the developed artifact for research and practice. In this thesis, it is presumed that truth is reached by consistently and cohesively embedding new knowledge in already existing recognized knowledge (coherence theory of truth) as well as by achieving consensus among acknowledged experts (consensus theory of truth) (Österle, Winter, & Brenner, 2010).

From an *epistemological* point of view, which describes the relation between the knower and the artifact under investigation, the author argues that reality cannot be perceived directly and that multiple socially constructed realities exist. The assumption of a subjective reality does not require determining whether an objective real world exists. As a consequence, the author of this thesis holds an open *ontological* position and neither refuses nor confirms the existence of an objective reality. A subjective perception of the reality and the constructed artifact based on this makes it likewise difficult for other researchers to grasp the intended meaning of the artifact from a *linguistic* point of view. However, applying a well-structured research methodology reduces the ambiguity while subjectively interpreting the meaning of the designed artifact.

A *research methodology* embraces a set of principles (rules, guidelines), processes (procedures, practices), and methods (instruments) that are applied in a *research project* when studying a *research domain* (Longman, 2009; Nunamaker & Chen, 1990, p. 632). IT artifacts are central in information systems and comprise constructs, models, methods, and instantiations (March & Smith, 1995, p. 253). Developing an ontology is a *design activity*, and the ontology itself as a model for the domain of enterprise crowdsourcing constitutes an IT artifact. As a consequence, the methodology that is employed in this thesis and that supports the design activity follows the *design science research* (DSR) in information systems. The DSR methodology provides guidance to create and evaluate innovative IT artifacts that serve human purposes and enhance organizational effectiveness (Hevner, March, Park, & Ram, 2004, p. 77; March & Smith, 1995).

According to Iivari (2014), two different *DSR strategies* in information systems can be pursued within a research project. The first strategy begins with a general problem or class of problems that are indirectly informed by the practice to build a conceptual IT meta-artifact. A real implementation of the artifact as a proof of concept may be used for the purpose of evaluation. In contrast, the second strategy begins with a specific problem faced directly by a certain client in practice and implements a real system mainly as a source of inspiration to derive the general DSR problem during the research project. The principal motivation of this thesis was to develop a universal ontology for a wide range of applications. Since following the second strategy encounters considerable uncertainty about the innovative artifact that qualify as a general DSR solution, this research project employed the first strategy to guide the overall research process.

The *research process* that was adopted in this thesis follows the suggestion of Peffers, Tuunanen, Rothenberger, and Chatterjee (2007). They identified six activities to support the design science process: (1) problem identification and motivation, (2) definition of the objectives for a solution, (3) design and development, (4) demonstration, (5) evaluation, and (6) communication. In the following, the embodiment of each of these six activities are outlined in the scope of this research project:

Activity 1: Problem identification and motivation

The specific nature of enterprise crowdsourcing focuses on complex and knowledge-intensive rather than on simple tasks and thus requires workers with appropriate skills, experiences, and knowledge to solve these tasks. Unfortunately, most of the current crowdsourcing systems rely on an amateur crowd to solve mainly simple tasks and offer less assistance to match tasks in regard to the workers' expectations, interests, and capabilities. Often the potential contributors are challenged to identify an appropriate task from a vast number and with a limited set of selection criteria. A further challenge results from the different types of crowdsourcing systems and intermediaries and the difficulty to interoperate with other business applications. These systems often apply different semantics to describe crowdsourcing tasks and offer only system-specific application programming interfaces. To reduce the efforts for implementing interfaces to and publishing tasks on multiple crowdsourcing platforms simultaneously, a common and system-independent language is required.

Activity 2: Define the objective for the solution

The research objective of this thesis was to improve the *automation* and *interoperability* in enterprise crowdsourcing environments by developing a semantic vocabulary in form of an ontology. In the first place, the ontology was designed to support the task allocation to potential participants taking into account not only human and technical requirements but also other criteria, such as incentive schemes or evaluation mechanisms. In addition, the goal was to provide a controlled vocabulary that software developers and architects can consult and adopt when building their own crowdsourcing system and integrating it in an existing information system architecture. Finally, the ontology was developed as a Semantic Web vocabulary and thus will form the basis for future standardization efforts. To guide the research project the following principal research question (RQ) was stated:

RQ: How must a semantic vocabulary be adequately designed and encoded to support automation and interoperability in enterprise crowdsourcing environments?

To adequately answer the overall research question, several subordinate questions required answers. These subordinate questions were addressed in five distinct research articles, which are part of this thesis and are introduced in section 4.

Activity 3: Design and development

The performance of an IT artifact depends on the environment in which it operates (March & Smith, 1995). A major challenge at the beginning of the research project resulted from the lack of a clear and shared understanding among researchers of what a crowdsourcing environment characterizes. As a consistent and comprehensive understanding is a prerequisite to design a commonly accepted ontology, two systematic literature reviews were conducted on that matter. The first literature review elaborated from a system-oriented perspective which components and functions are parts of a crowdsourcing system (Article 1). The second literature review investigated from a theoretical perspective the characteristics of what enterprise crowdsourcing constitutes (Article 2).

The ontology itself resulted from two build-evaluate cycles (Article 3 and Article 4). The design included three levels of abstractions starting from the contextual and conceptual layer (conceptual foundation), over the logical layer (data dictionary and schema), to the physical layer (implementation and instantiation). In order to derive a shared set of classes and properties, several sources of knowledge were considered, including a review of recent crowdsourcing literature, current crowdsourcing applications, and already existing semantic vocabularies (Article 5).

Activity 4: Demonstration

Prototyping was applied in this thesis as an iterative approach for improving the awareness of the nature of the problem and based on this for designing the artifact. An advantage of this approach is that you can search for a solution before understanding the problem in full detail (Oates, 2011, p. 114). After developing a conceptual prototype and showing its potential usefulness, the artifact was adopted to meet further requirements (Article 3). In continuative research, a data dictionary and a data schema were derived that were implemented in OWL later on (Article 4). As a first proof of demonstration, three use-case scenarios and corresponding instances of the ontology showed how the ontology can be used to query enterprise crowdsourcing data on the Semantic Web (Article 5).

Activity 5: Evaluation

As a designed IT artifact can affect organizational effectiveness positively or negatively, the ontology must be evaluated (March & Smith, 1995, p. 252). The author carried out evaluations at two different maturity levels of the ontology. At the first maturity level, a scenario-based evaluation showed the value of enriching the crowdsourcing process with semantics (Article 3). After the first set of semantic entities evolved to a more sophisticated ontology, additional methods were adopted at a second maturity level of the ontology. These methods comprised transforming informal to formal competency questions, examining the ontology to see whether it meets the defined functional and non-functional requirements, comparing the ontology with existing semantic vocabularies and standards, as well as calculating ontology metrics (Article 5).

Activity 6: Communication

Due to the novelty of the topic, the research contributions were disseminated in peer-reviewed conference articles. The first article of conducting a systematic literature review on components and functions of crowdsourcing systems appeared at the International Conference on Wirtschaftsinformatik (WI 2013) (Article 1). A set of constitutional properties and distinguishing features of enterprise crowdsourcing was presented at the 22nd European Conference on Information Systems (ECIS 2014) (Article 2). The vision of enhancing automation and interoperability in enterprise crowdsourcing environments based on a set of semantic entities was first introduced at the 21st European Conference on Information Systems (ECIS 2013) (Article 3). Finally, the development process of the ontology was published in two articles. The initial article provided a brief overview of the essential semantic entities of the ontology and was presented at the Multikonferenz Wirtschaftsinformatik (MKWI 2014) (Article 4). Additionally, an extended version of this article in form of a technical report was issued to explain the rigorous ontology engineering process and the artifact evaluation (Article 5). The implementation of the ontology was presented during a prototype and poster session at the 22nd European Conference on Information Systems (ECIS 2014) (Hetmank, 2014a; Appendix, p. 30). To reach not only an academic but also a broader audience, the schema definition of the ontology as well its documentation has been published online at http://purl.org/csm/.

4 STRUCTURE OF THE THESIS

The cumulative doctoral thesis comprises five research articles that are part of the overall research process and that finally led to the construction of an ontology for the enterprise crowdsourcing domain in form of a prototype implementation (Figure 1).

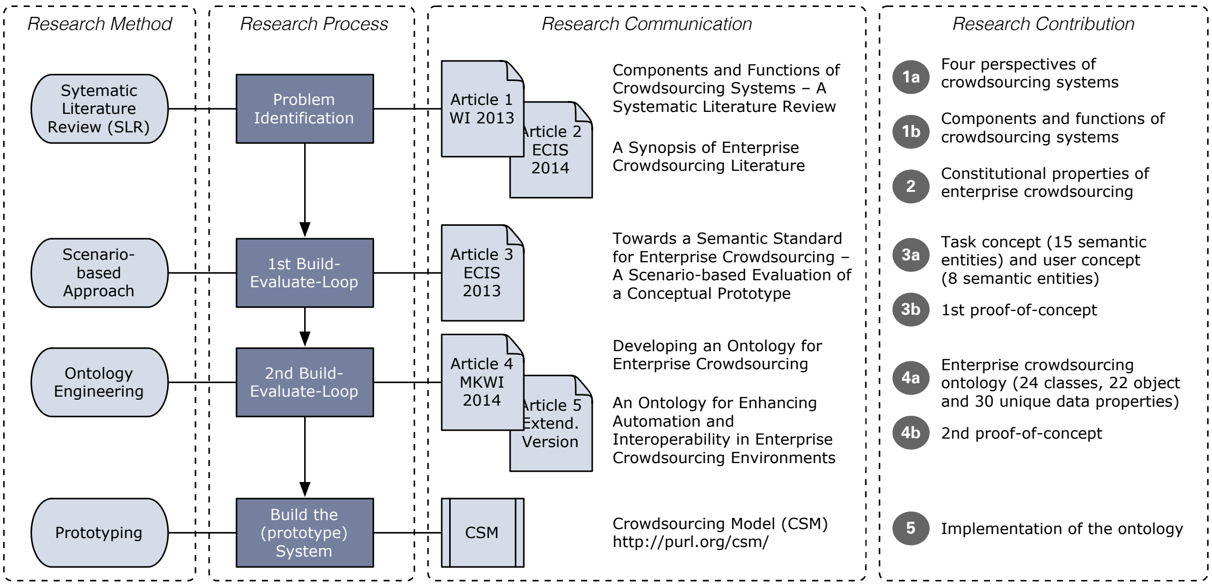

Figure 1: Research process and structure of the cumulative thesis

The thesis started with descriptive research that aimed to understand the nature of crowdsourcing (Article 1 and 2). Later, while developing the ontology, the focus was on prescriptive research (Article 3, 4, and 5). The first two articles of the thesis supported the problem identification and motivation (DSR Activity 1) and allowed the researcher to narrow down and to define the objective for the solution (DSR Activity 2). The design and development (DSR Activity 3) as well as the evaluation (DSR Activity 5) of the ontology itself consisted of two build-evaluate cycles. Article 3 applied a scenario-based approach to derive an initial set of semantic entities. As a systems-development methodology,[6] *ontology engineering* was then employed in articles 4 and 5 to gradually extend the set of semantic entities. Each partial result of the designed artifact was demonstrated and communicated at several conferences (DSR Activity 4 and 6).

In the following, the research objectives, questions, methods, and results of each research article that are part of the thesis are introduced:

Article 1: Components and Functions of Crowdsourcing Systems – A Systematic Literature Review (Hetmank, 2013a)

The first research article strived to gain a better understanding of what crowdsourcing systems are and what typical design aspects are considered in the development of such systems. It has been published and was presented at the 11th International Conference on Wirtschaftsinformatik in Leipzig (WI 2013). The work focused on answering the following two research questions (RQ):

RQ 1: How and in what detail are crowdsourcing systems defined in current research literature? What design aspects do they cover?

RQ 2: What type of components and functions of a crowdsourcing system can be conceptualized?

Based on a systematic literature review of crowdsourcing systems, two major theoretical research contributions were derived toward setting up a conceptual framework for supporting the development of a semantic vocabulary. As a first contribution, 17 definitions of crowdsourcing systems were identified and categorized into four perspectives: organizational, technical, functional, and human-centric. Each perspective included an essential set of design aspects. To improve the understanding of the technical perspective, the second contribution of this work was the development of a framework of typical components and functions that should be implemented in crowdsourcing systems.

Article 2: A Synopsis of Enterprise Crowdsourcing Literature (Hetmank, 2014b)

The second research article focused on enterprise crowdsourcing as one of the recent derivatives of crowdsourcing. As there is no clear and broad consensus on what the term

[6] The systems-development methodology, which leads the researcher through the phases of analysis, design, implementation, and testing, should not be confused with the overall research methodology, which is a combination of research strategies and methods used in a research project (Oates, 2011, p. 112).

enterprise crowdsourcing constitutes, this article aimed to explore and analyze the current body of literature in order to get a consolidated view of the different perspectives and applications and to identify the key characteristics of enterprise crowdsourcing. The article has been published and was presented at the 22nd European Conference on Information Systems in Tel Aviv (ECIS 2014). The article sought to answer the following research question:

RQ 3: What are the constitutional properties that make enterprise crowdsourcing unique compared to other types of crowdsourcing?

A systematic literature review was conducted to survey different explanations of the term and to derive the constitutional characteristics of enterprise crowdsourcing. Moreover several crowdsourcing applications were illustrated that enterprises deployed to aid either primary or support activities of the value-added chain. This work helped to understand the peculiarities of applying crowdsourcing in an enterprise context.

Article 3: Towards a Semantic Standard for Enterprise Crowdsourcing – A Scenario-based Evaluation of a Conceptual Prototype (Hetmank, 2013b)

The third research article was concerned with identifying an initial set of semantic entities for describing two of the main concepts in enterprise crowdsourcing activities: the crowdsourcing task and the user. The article was issued in the conference proceedings of the 21st European Conference on Information Systems in Utrecht (ECIS 2013). The article focused on the research question:

RQ 4: Which semantic entities are potential candidates for describing crowdsourcing tasks and users to meet the challenges that are currently prevailing in enterprise crowdsourcing systems?

Guided by a list of key challenges and supported by the results of the systematic literature reviews of articles 1 and 2, this work resulted in a first conceptual prototype of a semantic vocabulary for enterprise crowdsourcing. To demonstrate the general applicability of the designed prototype, three different scenarios that may occur in real business environments were constructed around the proposed semantic vocabulary: evaluate product design proposals (Scenario 1), translate a technical specification (Scenario 2), and build a company-wide virtual library (Scenario 3).

Article 4: Developing an Ontology for Enterprise Crowdsourcing (Hetmank, 2014d)

The fourth article focused on introducing an ontology with an extended set of semantic entities that facilitates the structured recording of enterprise crowdsourcing data. The paper was presented at the Multikonferenz Wirtschaftsinformatik in Paderborn (MKWI 2014). The work was guided by the following research question:

RQ 5: What are the essential semantic entities that form a universal and lightweight yet powerful ontology to enhance automation and interoperability in enterprise crowdsourcing environments?

The main contribution of this article was to illustrate the semantic entities of the designed ontology for capturing, storing, and linking crowdsourcing data. The ontology engineering itself was only briefly introduced. The ontology included 24 classes as well as 22 object properties, and 30 datatype properties to describe the main aspects of the crowdsourcing model.

Article 5: An Ontology for Enhancing Automation and Interoperability in Enterprise Crowdsourcing Environments (Hetmank, 2014c)

The last article is an extended version of article 4 in the form of a technical report. While article 4 describes the structure and the content of the ontology, this article is concerned more with the ontology engineering approach itself and how the ontology was created. In addition to RQ 5 of article 4, the technical report was mainly motivated by the following research questions:

RQ6: Which functional, non-functional, and reasoning requirements must an ontology for the domain of enterprise crowdsourcing comply?

RQ7: How should an ontology be designed based on the ontology engineering approach to support these requirements?

During the activity of building the ontology the following sources were applied to derive a set of semantic entities for the enterprise crowdsourcing domain:

- preliminary considerations that resulted from the scenario-based evaluation of the early prototype,
- existing knowledge of the enterprise crowdsourcing domain that is codified in frameworks, taxonomies, and models,

- an extensive and elaborative system analysis of 15 different crowdsourcing applications, and
- an investigation of Semantic Web vocabularies of related application domains.

The analyses of the various sources led to a data dictionary and a data schema, which were the basis for implementing the ontology using the Web Ontology Language (OWL). Whereas the semantic entities of the ontology were presented in full in article 4, the elaborate and multifaceted evaluation of the artifact was described in more detail in article 5. The following methods were used to evaluate the artifact:

- transforming informal to formal competency questions that are part of three use case scenarios,
- comparing the designed ontology to other semantic vocabularies, and
- presenting the potential expressive power of the ontology by calculating different ontology metrics.

5 RESEARCH CONTRIBUTION

This section explains the key research contributions of the thesis. Compared to conventional industry-based design and creation work, a DSR project must introduce new knowledge to the existing knowledge base. Research contributions in the form of IT artifacts can generally range from specific implementations (Level 1), over more abstract models, methods and principles (Level 2), to well-developed design theories (Level 3) (Gregor & Hevner, 2013). The results of the thesis contributed to IS research in multiple ways, mainly at the first two levels of abstraction. First, the thesis argued in favor of an ontology for the new emerging domain of enterprise crowdsourcing, which has previously not been automated based on Semantic Web technologies. Second, by applying diverse sources of knowledge, a crowdsourcing model (CSM) in form of a lightweight and extensible ontology was derived. Third, the model was implemented in OWL to demonstrate the technical feasibility and viability of the artifact.

Before summarizing the main contributions for research and practice, a broad overview of the CSM ontology is given. The unified modeling language (UML) class diagram of the ontology illustrates the interplay between the CSM semantic entities (Figure 2).

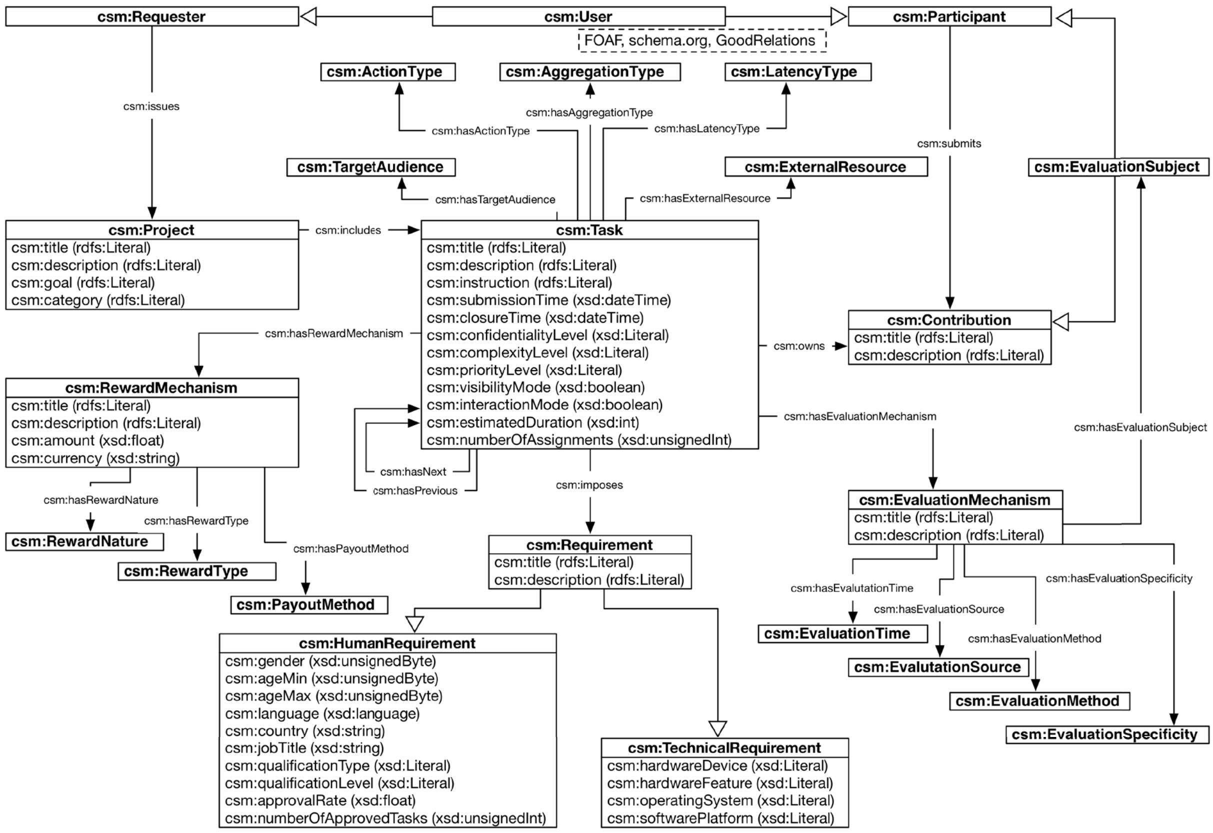

Figure 2: UML class diagram of the CSM ontology

The CSM ontology considers two roles for an enterprise crowdsourcing activity: the requester (csm:Requester) and the participant (csm:Participant). Both are modeled as subclasses of a universal user class (csm:User) that acts as a link between the ontology and the concepts of other vocabularies. For current purposes, the person and organization concept of the FOAF and schema.org vocabulary (foaf:Person, foaf:Organization, schema.org/Person, schema.org/Organization) as well as the business entity concept of the Good Relations vocabulary (gr:BusinessEntity) are associated to benefit from the reuse of their properties. Each crowdsourcing project (csm:Project) that the requester issues includes one or more crowdsourcing tasks (csm:Task). The sequence of multiple crowdsourcing tasks within a project can be determined by using the csm:hasNext and csm:hasPrevious properties. The task concept comprises several data properties to specify the task characteristics, such as the instruction (csm:instruction), the submission and closure time (csm:submissionTime, csm:closureTime), or the confidentiality and priority level (csm:confidentialityLevel, csm:priorityLevel). Some aspects of the crowdsourcing task are designed as independent classes to enhance the semantics of the overall ontology. These classes are, for instance, the reward mechanism, the evaluation mechanism, the human requirement, or the technical requirement (csm:RewardMechanism, csm:Evaluation-Mechanism, csm:HumanRequirement, csm:TechnicalRequirement).

The CSM ontology claims to provide three major contributions for researchers and practitioners in the field of enterprise crowdsourcing:

- *The ontology as a frame of reference for software development:* The data dictionary and the data schema of the ontology serve as a frame of reference to develop application-specific enterprise crowdsourcing systems. Software architects may adopt the proposed semantic entities of the ontology that best suit their application domain.

- *The ontology as a facilitator for linking the crowdsourcing system with other business functions and applications:* Within the company, the ontology acts as a lingua franca to integrate existing enterprise crowdsourcing solutions with other business applications. Thus, it connects the crowdsourcing system with applications, such as enterprise dictionaries, skill databases, and competency management systems.

- *The ontology as a foundation toward a standard vocabulary for the Semantic Web:* At the scale of the Semantic Web, the ontology provides a good starting point to establish a standard for advertising and posting tasks to the crowd on the Web that is commonly understood, accepted, and supported by a wide range of applications and functions. These comprise not only crowdsourcing applications but also search engines, web crawlers, or intelligent software agents. An accepted and standardized Semantic Web vocabulary would allow publishing a crowdsourcing task on multiple intermediary crowdsourcing platforms and thus reaching a wider audience.

6 CONCLUSION

The primary focus of this thesis was on exploring the possibilities of Semantic Web technologies for enhancing automation and interoperability in enterprise crowdsourcing environments. At present, only a few researchers have made initial efforts in the direction of enriching crowdsourcing environments with semantics. Based on the data model of provenance (PROV) standardized by the World Wide Web Consortium (W3C), Celino (2013) developed an ontology, namely the human computation ontology, for representing the provenance of user-generated geospatial data. Hassan, O'Riain, and Curry (2013) created the lightweight Semantically Linked Users and Actions (SLUA) ontology for describing users, tasks, actions, rewards, and human capabilities in micro-task crowdsourcing platforms. Both ontologies have their limitations in expressing detailed reward schemes and evaluation mechanisms as well as in defining human and technical requirements. Thus, a more comprehensive ontology was designed in this research project to address the particular needs of enterprise crowdsourcing.

Although this thesis led to the first well-grounded ontology for the enterprise crowdsourcing domain, there are some limitations that leave room for follow-up research. First, the designed artifact has been mainly communicated to researchers whose feedback and comments influenced the design of the ontology. However, to further improve the ontology, other relevant audiences, namely practicing professionals, should be more closely involved in upcoming design-evaluate cycles. Specifically, measuring the economic impact while introducing the ontology in an existing crowdsourcing application may inform future efforts to improve the ontology. Research methods for carrying out real-world evaluation may be included, such as case studies, focus groups, or surveys.

Second, the Semantic Web requires carefully designed and commonly accepted standards based on a shared conceptualization. Although the developed ontology provides a solid foundation for a Semantic Web vocabulary, future efforts are needed to initialize and foster the standardization process. In this regard, difficulties may occur in the fact that the concepts and properties of the ontology may not be stable and change over time as "new procedures and understanding emerge" (Shadbolt, Berners-Lee, & Hall, 2006). Thus, managing and endorsing the ontology requires committed practice communities.

Third, prospects for enhancing the semantics of the ontology should be investigated. These comprise the consideration of additional concepts, properties, and individuals, or the introduction of axioms that define further information or restrictions about these concepts, properties, and individuals. An interesting experiment for upcoming research in this regard would be to bring the next improvement cycle of the ontology to the qualified crowd itself. There are several starting points for investigations to improve the semantics of the ontology. The modeling of complex crowdsourcing workflows is currently limited within the ontology as it only provides mechanisms for defining previous and subsequent tasks. This is the strength of business-process modeling languages, such as the Web Services Business Process Execution Language (WS-BPEL) or the XML Process Definition Language (XPDL). Thus, future research should examine how these process modeling languages can be integrated into the proposed ontology. Moreover, researchers have already noted that a deeper understanding is required of how to relate technical dependencies of products and services to social dependencies of the collaborating crowd members (Skopik et al., 2012). These insights may also inform future improvement cycles of the ontology. Another aspect for enhancement may be the representation of context information passed to future contributors and considered by them to create better and more informed solutions, especially in iterative crowdsourcing processes (Zhang, Horvitz, Miller, & Parkes, 2011).

Fourth, in future research projects, functionalities should be developed that export CSM data from existing crowdsourcing platforms according to the semantic entities of the ontology. Additionally, semantic search engines should be built that enable more complex search queries about Web pages annotated with CSM data and that lead to more accurate search results when searching for a crowdsourcing task. Finally, recommender systems should be designed that analyze CSM data for predicting the preference of a potential contributor for an issued crowdsourcing task.

Although follow-up research remains from a content-related as well as from a methodological perspective, this thesis achieved the research objective by proposing a first comprehensive ontology for the domain of enterprise crowdsourcing. Researchers and practitioners are now encouraged to critically assess and, where necessary, adapt and improve the designed ontology.

REFERENCES

Adida, B., Birbeck, M., McCarron, S., & Herman, I. (2013). RDFa Core 1.1 - Second Edition: syntax and processing rules for embedding RDF through attributes (W3C Recommendation 22 August 2013). World Wide Web Consortium (W3C). Retrieved from http://www.w3.org/TR/2013/REC-rdfa-core-20130822/

Alvesson, M. (2004). *Knowledge work and knowledge-intensive firms*. New York: Oxford University Press.

Becker, J., Holten, R., Knackstedt, R., & Niehaves, B. (2003). *Forschungsmethodische Positionierung in der Wirtschaftsinformatik: epistemologische, ontologische und linguistische Leitfragen* (No. 93). *Wirtschaftsinformatik*. Institut für Wirtschafts-informatik.

Berners-Lee, T., Hendler, J., & Lassila, O. (2002). The Semantic Web: a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American Special Online Issue*, 25–30.

Bick, M., Hetmank, L., Kruse, P., Maier, R., Pawlowski, J. M., Peinl, R., Schoop, E., Seeber, I. & Thalmann, S. (2012). Manifesto for a standard on meaningful representations of knowledge in social knowledge management environments. In D. C. Mattfeld & S. Robra-Bissantz (Eds.), *Multikonferenz Wirtschaftsinformatik 2012 - Tagungsband der MKWI 2012* (pp. 1359–1375). Braunschweig: GITO Verlag.

Boder, A. (2006). Collective intelligence: a keystone in knowledge management. *Journal of Knowledge Management*, *10*(1), 81–93. doi:10.1108/13673270610650120

Brabham, D. C. (2013). *Crowdsourcing*. Cambridge, Mass.: MIT Press.

Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., & Yergeau, F. (2008). Extensible Markup Language (XML) 1.0 (Fifth Edition, W3C Recommendation 26 November 2008). World Wide Web Consortium (W3C). Retrieved from http://www.w3.org/TR/2008/REC-xml-20081126/

Brickley, D., & Guha, R. V. (2014). RDF Schema 1.1 (W3C Recommendation 25 February 2014). World Wide Web Consortium (W3C). Retrieved from http://www.w3.org/TR/rdf-schema/

Celino, I. (2013). Human computation VGI provenance: Semantic Web-based representation and publishing. *Geoscience and Remote Sensing, IEEE Transactions on*, *51*(11), 5137–5144. doi:10.1109/TGRS.2013.2252015

Cyganiak, R., Wood, D., & Lanthaler, M. (2014). RDF 1.1 Concepts and abstract syntax (W3C Recommendation 25 February 2014). World Wide Web Consortium (W3C). Retrieved from http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/

Erickson, L. B., & Trauth, E. M. (2013). Getting work done: evaluating the potential of crowdsourcing as a model for business process outsourcing service delivery. In

Proceedings of the 2013 Annual Conference on Computers and People Research (pp. 135–140). New York, NY, USA: ACM. doi:10.1145/2487294.2487323

Fong, P. S. W., & Choi, S. K. Y. (2009). The processes of knowledge management in professional services firms in the construction industry: a critical assessment of both theory and practice. *Journal of Knowledge Management*, *13*(2), 110–126. doi:10.1108/13673270910942736

Gassenheimer, J., Siguaw, J., & Hunter, G. (2013). Exploring motivations and the capacity for business crowdsourcing. *AMS Review*, *3*(4), 205–216. doi:10.1007/s13162-013-0055-8

Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly*, *37*(2), 337–A6.

Grimm, S., Abecker, A., Völker, J., & Studer, R. (2011). Ontologies and the Semantic Web. In J. Domingue, D. Fensel, & J. A. Hendler (Eds.), *Handbook of Semantic Web Technologies* (pp. 507–570). Berlin: Springer.

Hassan, U. ul, O'Riain, S., & Curry, E. (2013). SLUA: towards semantic linking of users with actions in crowdsourcing. In *12th Interantional Semantic Web Conference*. Sydney.

Hetmank, L. (2013a). Components and functions of crowdsourcing systems – a systematic literature review. In *11th International Conference on Wirtschaftsinformatik* (pp. 55–69). Leipzig.

Hetmank, L. (2013b). Towards a semantic standard for enterprise crowdsourcing – a scenario-based evaluation of a conceptual prototype. In *21st European Conference on Information Systems*. Utrecht.

Hetmank, L. (2014a). A lightweight ontology for enterprise crowdsourcing. In *22nd European Conference on Information Systems* (pp. 1–8). Tel Aviv.

Hetmank, L. (2014b). A synopsis of enterprise crowdsourcing literature. In *22nd European Conference on Information Systems*. Tel Aviv.

Hetmank, L. (2014c). An ontology for enhancing automation and interoperability in enterprise crowdsourcing environments (technical report). Retrieved from http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-155187

Hetmank, L. (2014d). Developing an ontology for enterprise crowdsourcing. In *Multikonferenz Wirtschaftsinformatik 2014*. Paderborn.

Hevner, A., March, S., Park, J., & Ram, S. (2004). Design science in information systems research. *Mis Quarterly*, *28*(1), 75–105.

Hickson, I. (2013). HTML Microdata (W3C Working Group Note 29 October 2013). World Wide Web Consortium (W3C). Retrieved from http://www.w3.org/TR/microdata/

Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P. F., & Rudolph, S. (2012). OWL 2 Web Ontology Language Primer (Second Edition, W3C Recommendation 11 December

2012). World Wide Web Consortium (W3C). Retrieved from http://www.w3.org/TR/owl2-primer/

Howe, J. (2008). *The rise of crowdsourcing: why the power of the crowd is driving the future of business*. New York: Three Rivers Press.

Iivari, J. (2014). Distinguishing and contrasting two strategies for design science research. *European Journal of Information Systems*.

Islam, N., Abbasi, A. Z., & Shaikh, Z. A. (2010). Semantic Web: choosing the right methodologies, tools and standards. In *Proceedings of the International Conference on Information and Emerging Technologies (ICIET)* (pp. 1–5). doi:10.1109/ICIET.2010.5625736

Jayakanthan, R., & Sundararajan, D. (2011). Enterprise crowdsourcing solutions for software development and ideation. In *Proceedings of the 2nd International Workshop on Ubiquitous Crowdsouring* (pp. 25–28). New York, NY, USA: ACM. doi:10.1145/2030100.2030108

Khare, R., & Çelik, T. (2006). Microformats: a pragmatic path to the Semantic Web. In *Proceedings of the 15th International Conference on World Wide Web* (pp. 865–866). New York, NY, USA: ACM. doi:10.1145/1135777.1135917

Lee, M., & Lan, Y. (2007). From Web 2.0 to conversational knowledge management: towards collaborative intelligence. *Journal of Entrepreneurship Research*, *2*(2), 47–62.

Leimeister, J. (2010). Collective intelligence. *Business & Information Systems Engineering*, *2*(4), 245–248. doi:10.1007/s12599-010-0114-8

Longman, P. (2009). *Dictionary of Contemporary English*. Pearson/Longman.

Lopez, M., Vukovic, M., & Laredo, J. (2010). PeopleCloud service for enterprise crowdsourcing. In *Proceedings of the International Conference on Services Computing (SCC)* (pp. 538–545). doi:10.1109/SCC.2010.74

Maier, R. (2007). *Knowledge management systems: information and communication technologies for knowledge management* (3rd ed.). Berlin: Springer.

Maiolini, R., & Naggi, R. (2011). Crowdsourcing and SMEs: opportunities and challenges. In A. D'Atri, M. Ferrara, J. F. George, & P. Spagnoletti (Eds.), *Information Technology and Innovation Trends in Organizations* (pp. 399–406). Physica-Verlag.

Malone, T. W., Laubacher, R., & Dellarocas, C. (2010). *Harnessing crowds: mapping the genome of collective intelligence* (No. 4732-09). *MIT Sloan School Working Paper 4732-09* (Vol. 1, pp. 1–21).

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, *15*(4), 251–266. doi:10.1016/0167-9236(94)00041-2

Middleton, S. E., De Roure, D., & Shadbolt, N. R. (2009). Ontology-based recommender systems. In S. Staab & R. Studer (Eds.), *Handbook on Ontologies* (pp. 779–796). Springer Berlin Heidelberg.

Nunamaker, J. F., & Chen, M. (1990). Systems development in information systems research. In *Proceedings of the 23rd Annual Hawaii International Conference on System Sciences* (Vol. iii, pp. 631 –640 vol.3). doi:10.1109/HICSS.1990.205401

Oates, B. J. (2011). Researching information systems and computing. SAGE.

Österle, H., Winter, R., & Brenner, W. (2010). Gestaltungsorientierte Wirtschaftsinformatik: Ein Plädoyer für Rigor und Relevanz. *St. Gallen Ness & Information Systems Engineering*, *1*(1), 1–4.

Pedersen, J., Kocsis, D., Tripathi, A., Tarrell, A., Weerakoon, A., Tahmasbi, N., Xiong, J., Deng, W., Oh, O. & de Vreede, G.-J. (2013). Conceptual foundations of crowdsourcing: a review of IS research. In *Proceedings of the Hawaii International Conference on System Science (HICCS-46)*. Grand Wailea, Maui, Hawaii.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*(3), 45–77.

Ribiere, V. M., & Tuggle, F. D. (Doug). (2010). Fostering innovation with KM 2.0. *Vine*, *40*(1), 90–101. doi:10.1108/03055721011024955

Schall, D. (2012). *Service-oriented crowdsourcing: architecture, protocols and algorithms*. New York: Springer.

Shadbolt, N., Berners-Lee, T., & Hall, W. (2006). The Semantic Web revisited. *IEEE Intelligent Systems*, *21*(3), 96–101. doi:10.1109/MIS.2006.62

Simula, H. (2013). The rise and fall of crowdsourcing? In *Proceedings of the 46th Hawaii International Conference on System Sciences (HICSS)* (pp. 2783–2791). doi:10.1109/HICSS.2013.537

Skopik, F., Schall, D., & Dustdar, S. (2012). Discovering and managing social compositions in collaborative enterprise crowdsourcing systems. *International Journal of Cooperative Information Systems*, *21*(4), 297–341.

Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: principles and methods. *Data & Knowledge Engineering*, *25*(1–2), 161–197. doi: 10.1016/S0169-023X(97)00056-6

Sunstein, C. R. (2008). *Infotopia: how many minds produce knowledge*. Oxford: Oxford Univ. Press.

Vukovic, M. (2009). Crowdsourcing for enterprises. In *Proceedings of the 2009 Congress on Services - I* (pp. 686–692). Los Angeles: IEEE. doi:10.1109/SERVICES-I.2009.56

Vukovic, M., Salapura, V., & Rajagopal, S. (2013). Crowd-enabled technical writing services. In *Proceedings of the International Conference on Services Computing (SCC)* (pp. 635–642). doi:10.1109/SCC.2013.84

Zhang, H., Horvitz, E., Miller, R., & Parkes, D. (2011). Crowdsourcing general computation. In *Proceedings of the 2011 ACM Conference on Human Factors in Computing Systems*. Vancouver: Association for Computing Machinery.

APPENDIX

TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Business and Economics | Chair of Business Information Systems, esp. Information Management

CSM ONTOLOGY

ENHANCING AUTOMATION AND INTEROPERABILITY IN **ENTERPRISE CROWDSOURCING** ENVIRONMENTS

RESEARCH OBJECTIVE

The overall objective of this research project is to develop a **lightweight** and **extensible ontology** for capturing, storing, utilizing, and sharing **enterprise crowdsourcing** data that is grounded on Semantic Web technologies and Linked Data principles.

The deployment of an ontology for enterprise crowdsourcing aims to provide several benefits for an organization:

- Provide a controlled vocabulary for the enterprise crowdsourcing domain to reduce ambiguity and to ensure consistency.
- Increase the overall transparency of the entire crowdsourcing activity and thus allow a more sophisticated quality control.
- Deploy enhanced semantic and intelligent search, navigation, and browsing capabilities in crowdsourcing systems.
- Form smaller task-specific crowds that perform tasks at lower cost and in shorter time than larger crowds.
- Judge the candidate's trustworthiness based on previous accomplished tasks.
- Evaluate historical crowdsourcing data to support the configuration of similar tasks.
- Improve interoperability between crowdsourcing platforms and other business applications.

RESEARCH METHODOLOGY

The development process of the ontology follows the **ontology engineering** approach.

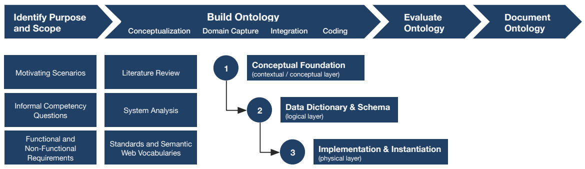

Figure 1: Development process of the ontology based on Grüninger and Fox (1995), Uschold and King (1995), Gómez-Pérez et al. (2005)

Five key **requirements** are derived to guide the design of the ontology.

Requirement		Central question
Task specification		Which semantic entities support requesters in their efforts of specifying a crowdsourcing task that will draw an audience?
Task allocation	Task distribution (requester-oriented)	Which semantic entities aid requesters in proposing a crowdsourcing task to an appropriate and available user with the required qualifications?
	Task selection (participant-oriented)	Which semantic entities improve the self-selection of a crowdsourcing task that fits best to the participants' knowledge, skill, and experience?
Team building	Team identification (requester-oriented)	Which semantic entities are applied to identify existing teams, working groups, or online communities of a particular knowledge domain in social networks that are suitable for a certain crowdsourcing task?
	Team formation (participant-oriented)	Which semantic entities help to foster the self-formation process of the participants?
Transaction transparency and quality control		Which semantic entities support the evaluation process of a crowdsourcing user or contribution and are essential to improve the transparency of a crowdsourcing activity according to specific roles?
Interoperability, data integration, and data exchange		Which semantic entities are required to support the interoperability between the crowdsourcing system and other business and social software applications?

Table 1: Requirements and central questions

PROTOTYPE IMPLEMENTATION

The designed **crowdsourcing model (CSM)** describes the key conceptual entities of enterprise crowdsourcing: user, project, task, requirement, reward mechanism, evaluation mechanism, and contribution.

The **Web Ontology Language (OWL)** is applied to formalize the prototype of the ontology (**http://www.purl.org/csm/**).

For the implementation of the ontology, the open source ontology development editor **Protégé** was applied.

The ontology contains **24 classes**, **22 object properties**, and **30 data properties** to describe the key aspects of a typical enterprise crowdsourcing activity. Additionally, several common individuals are suggested, which may be extended in future versions of the ontology. All semantic entities of the ontology are supplemented with additional annotation information.

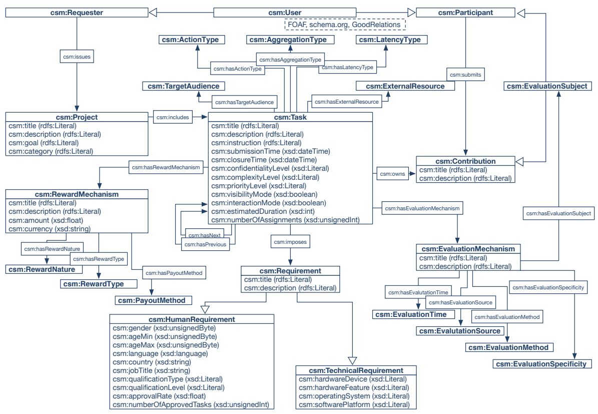

Figure 2: Semi-formal specification of the ontology (own illustration)

TU Dresden
Faculty of Business and Economics
Chair of Business Information Systems, esp. Information Systems | Prof. Schoop

Lars Hetmank, PhD Student
Mobile: +49 176 24 77 39 65

E-Mail: lars.hetmank@tu-dresden.de

The Crowdsourcing Model (CSM)

Specification can be found at:

http://purl.org/csm/

Components and Functions of Crowdsourcing Systems – A Systematic Literature Review

Lars Hetmank

Abstract. Many organizations are now starting to introduce crowdsourcing as a new model of business to outsource tasks, which are traditionally performed by a small group of people, to an undefined large workforce. While the utilization of crowdsourcing offers a lot of advantages, the development of the required system carries some risks, which are reduced by establishing a profound theoretical foundation. Thus, this article strives to gain a better understanding of what crowdsourcing systems are and what typical design aspects are considered in the development of such systems. In this paper, the author conducted a systematic literature review in the domain of crowdsourcing systems. As a result, 17 definitions of crowdsourcing systems were found and categorized into four perspectives: the organizational, the technical, the functional, and the human-centric. In the second part of the results, the author derived and presented components and functions that are implemented in a crowdsourcing system.

Note. This research article was published as Hetmank, L. (2013). Components and Functions of Crowdsourcing Systems – A Systematic Literature Review. In *11th International Conference on Wirtschaftsinformatik* (pp. 55–69). Leipzig.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

The research of crowdsourcing is a vigorous research area that has been steadily increasing over the last several years (Zhao & Zhu, 2012) and there is still an ongoing need for scientific engagement in this field (Hammon & Hippner, 2012; Leimeister, Huber, Bretschneider, & Krcmar, 2009). Crowdsourcing is a powerful mechanism for outsourcing tasks, which are traditionally performed by a specialist or small group of experts, to a large group of humans (Greengard, 2011). It is used for a variety of applications, such as evaluating ideas, creating knowledge repositories, or developing new products collaboratively. The main advantage of crowdsourcing lies in the way how it significantly changes the business processes by harnessing skills, knowledge or other resources of a distributed crowd to achieve an outcome at lower cost and in shorter time (Vukovic & Bartolini, 2010). Besides using existing external crowdsourcing solutions, such as Amazon Mechanical Turk or Innocentive, many organizations are now starting to develop their own crowdsourcing systems (CSS). However, the development of a CSS as well as its integration into an existing information and communication technology environment is a risky and difficult undertaking, which has to be planned thoroughly based on a profound theoretical foundation. Thus, to support the requirements engineering and architectural design of CSSs, the main objectives of this paper are first to provide a better understanding of what CSSs are from the technical point of view, and second to identify components and functions that are considered when designing a CSS. To this end, the author conducts a systematic literature review to revise current research efforts in the field of CSSs. The results from this article are an attempt to move the procedure of developing CSSs from an ad hoc manner to a planned routine that is based on a list of typically implemented components and functions.

The remainder of this article is structured as follows: The second section gives an overview of related conceptual work in the domain of crowdsourcing. The research method used in this study is described in the subsequent section. In section four, definitions of CSSs are categorized and typical components and functions of CSSs are presented. Finally, the author critically reflects on the results, depicts limitations of the work and highlights future research directions.

2 STATE-OF-THE-ART

Theoretical examinations in the domain of crowdsourcing have been conducted in a variety of directions and fields of research. One of the first attempts in scientific literature to define crowdsourcing as a new model for problem solving was made by Brabham (2008). Since then a lot of various *crowdsourcing definitions* have been proposed. Recently, Estellés-Arolas and González-Ladrón-de-Guevara (2012) analyzed existing definitions of crowdsourcing and created an integrated definition that considers several specific aspects of the crowd, the initiator and the underlying process.

The *process perspective* on crowdsourcing was examined in detail by Geiger, Seedorf, Schulze, Nickerson, and Schader (2011) who developed a taxonomic framework for crowdsourcing processes. The authors identified four dimensions that describe how crowdsourcing processes can be configured, ranging from pre-selection of contributors, accessibility of contributors, aggregation of contributors to remuneration for contribution.

Several authors have drawn their attention to *crowdsourcing taxonomies*. Rouse (2010), for example, decomposed the term "crowdsourcing" into several subtypes. These subtypes form a crowdsourcing taxonomy that is based on the nature of the task (simple, moderate or sophisticated tasks), the distribution of the benefits (individualistic, community or mixed), and the forms of motivation. Another typology of crowdsourcing practices is illustrated by Schenk and Guittard (2011). Two aspects are relevant for their typology. The first aspect focuses on the value of the individual's contribution, which may either only be valuable when combined with other contributions (integrative crowdsourcing) or already be valuable by addressing a specific problem of the initiator directly (selective crowdsourcing). The second aspect addresses, similar to Rouse's taxonomy, the type of the issued tasks (simple, complex and creative tasks).

According to a well-established model of the computer supported cooperative work (CSCW) domain that proposes a classification based on the distribution over time and space, Erickson (2011) derived his own four-quadrant crowdsourcing model, in which he suggests four modes of crowdsourcing: audience-centric (same time and place), event-centric (same time and different places), geocentric (different times and same place) and global crowdsourcing (different times and places). Yuen, King, and Leung (2011) surveyed various crowdsourcing literatures and allocated them into four categories: the type of application (voting system, information sharing system, game, or creative system), the used algorithm, the performance (user participation, quality management and cheating detection) and the datasets available. The most recent, sophisticated classification of CSSs was proposed by Doan, Ramakrishnan, and Halevy (2011). They defined nine dimensions to classify existing CSSs: the nature of collaboration, the type of target problem, the design of incentive mechanism, the type of contribution, the approach to combine solutions, the method to evaluate users, the degree of manual effort, the role of human users, and the type of architecture (standalone versus piggyback).

Several well-established *conceptual frameworks* have been proposed to guide decision-makers, software architects and project managers through the design process of CSSs. Kazman and Chen (2009), for instance, argue that prior life-cycle models in software development, such as the waterfall model or the spiral model, do not meet properly the requirements of commons-based peer production and the service-oriented nature of crowdsourcing. Thus, they suggest a new system-development model called the metropolis model that offers a new logic of thinking and propose several principles to design CSSs. Malone, Laubacher, and Dellarocas (2010) specify a further conceptual framework. Their proposed framework contains four building blocks that are important in designing collective intelligence systems. They classify the four building blocks, also called "genes," by addressing the following four questions: What is being done? Who is doing it? Why are they doing it? and How is it being done?

While there have been a number of valuable studies regarding (i) the definition of crowdsourcing (Brabham, 2008; Estellés-Arolas & González-Ladrón-de-Guevara, 2012), (ii) the characterization of the crowdsourcing process (Geiger et al., 2011), (iii) the development of a crowdsourcing taxonomy (Doan et al., 2011; Erickson, 2011; Rouse, 2010; Schenk & Guittard, 2011; Yuen et al., 2011), and (iv) the introduction of a conceptual framework that supports the designing of CSSs (Kazman & Chen, 2009; Malone et al., 2010), little has been investigated to define a CSS and its technical design precisely. However, a clear theoretical understanding supports a structured development process of CSSs. Therefore, an extensive literature review was conducted that on the one hand aimed for categorizing existing definitions of CSSs and on the other hand gave insights of typical design aspects of a CSS.

3 RESEARCH METHOD

To improve the understanding on functional and technical requirements of CSSs, a systematic literature review (SLR) was conducted, which will be described in the following section. A SLR provides a well-structured and repeatable procedure to identify, evaluate and interpret existing literature relevant to a specific research question (Kitchenham, 2007). The main goal of a SLR is not only to methodically aggregate scientific studies in a certain research domain but also to support the development of evidence-based guidelines for practitioners (Kitchenham et al., 2009).

The procedure of the literature review including all created results was carefully documented in a review protocol and contains four steps: (i) plan systematic literature review, (ii) conduct search of articles, (iii) screen papers and (iv) extract data (Figure 1).

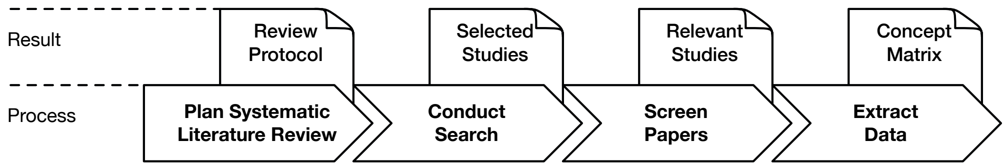

Figure 1: Systematic literature review procedure

3.1 PLANNING THE SYSTEMATIC LITERATURE REVIEW

In the planning stage of the literature review several steps were taken. First, the research interest of the paper was stated in the form of two research questions. Second, after formulating the research questions an appropriate search strategy was derived.

Research Questions. The main goal of the SLR was to investigate the research area of crowdsourcing from a system point of view. Therefore, the literature review addresses the following research questions (RQ):

RQ1: How and in which detail are CSSs defined in current research literature?
What design aspects do they cover?

RQ2: What type of components and functions of a CSS can be conceptualized?

Search Strategy. The search strategy comprises the determination of the population, the selection of search resources, the identification of search strings, and the definition of inclusion and exclusion criteria.

Population. The author searched for peer-reviewed conference proceedings and journal papers since 2006 when the term *crowdsourcing* was first coined by Howe (2006). For getting a general overview, there was no need to cover the broad range of publication types. Hence, books, dissertations, newspaper articles, unpublished works or non-scientific articles were not considered. The databases used below focus on English scientific papers (except SpringerLink). For that reason, articles that were not published in English were removed from the initial population. Finally, only full papers that could be accessed through the database subscription of the library were included.

Search Resources. With respect to search resources, all databases that contained articles of the relevant population as well as were accessible through the library subscription, such as ACM Digital Library, Ebscohost (Academic Search Complete and Business Source Complete), Emerald, IEEE Xplore Digital Library, Sage Journals, ScienceDirect, SpringerLink and Wiley, were used.

Search Terms. From the RQs, *crowdsourcing system* was derived as a first search term. After screening several papers that discuss crowdsourcing systems, two other related terms were found in the same context: *crowdsourcing application* and *crowdsourcing platform.* However to support the decision of the chosen search terms, several other test queries were conducted (Table 1). First, the term *crowdsourcing* was applied to all databases considering all document metadata fields. In this case, the total amount of publications reached 1699 entries. To limit the set of articles, the same term was used again, but with the restriction that only keywords were taken into account. The population of the paper was reduced to 337, an amount that could be handled in a reasonable amount of time. Finally, the initial choice of search terms: *crowdsourcing system*, *crowdsourcing application* and *crowdsourcing platform* (both in singular and plural form) resulted in 220 research papers in total. After checking the relevance of several abstracts of the prior results, the initial variant was chosen, which was most appropriate to address the RQs stated above.

Table 1: Number of publications found by applying diverse databases and search terms

Database / Search string[1] and restrictions	Crowdsourcing		Crowdsourcing system(s) Crowdsourcing application(s) Crowdsourcing platform(s)
	all fields	**keyword**	**all fields**
ACM Digital Library	843	184	139
Ebscohost	66	17	4
Emerald	55	5	3
IEEE Xplore Digital Library	138	83	14
SAGE Journals	73	11	8
ScienceDirect	203	18	18
Springerlink	166	15[2]	22
Wiley	155	4	12
Total amount of publications found	**1699**	**337**	**220**

Inclusion Criteria. The literature review includes peer-reviewed journal articles and conference contributions that:

- define or at least propose a description of what CSSs are (RQ 1),
- address design issues of CSSs (RQ 2), or
- classify or give an overview of CSSs (RQ 2).

Exclusion Criteria. Articles that used CSSs, such as Amazon Mechanical Turk for evaluation research purposes, but that do not discuss any design issues were excluded.

3.2 CONDUCTION OF THE SEARCH

The selection of relevant studies was processed in two stages. At first, the abstract, introduction and conclusion of all relevant studies were reviewed. This approach has proved to be necessary for literature of information technology and software engineering, in which the abstracts are too poor to rely solely on them (Brereton, Kitchenham, Budgen, Turner, & Khalil, 2007). An article was included in the set of relevant studies if it met one of the inclusion criteria and was not rejected by the exclusion criterion. Simultaneously, each paper was

[1] The databases were queried on July 18, 2012.
[2] Since SpringerLink does not provide a keyword search, the search was restricted to the title and the abstract of the publications.

classified according to publication type and research approach (Figure 2) (Wieringa, Maiden, Mead, & Rolland, 2006).

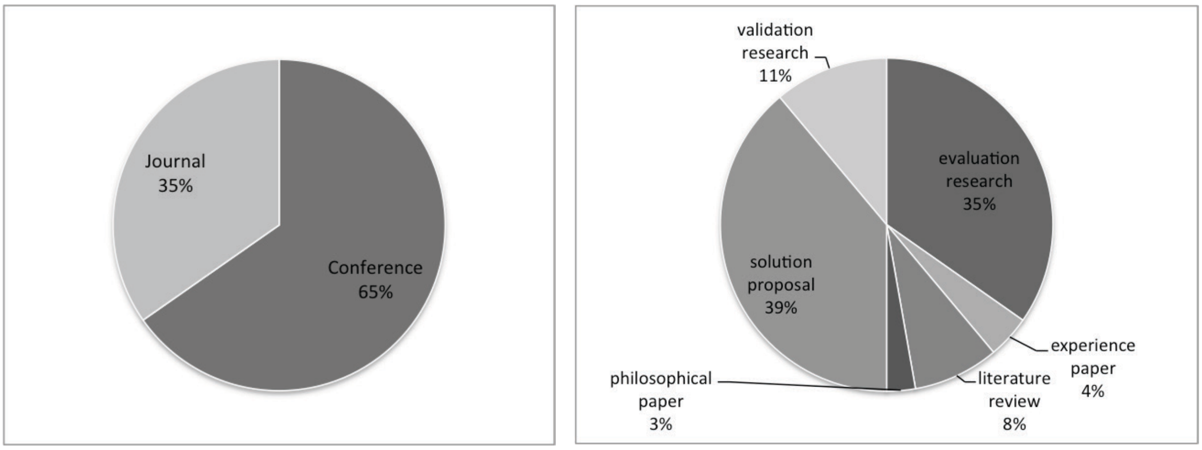

Figure 2: Article distribution regarding publication type (left) and research type (right)

After having identified all relevant studies, in sum 72, all articles were carefully read in order to find and record all definitions, descriptions and uses of the term *crowdsourcing system, crowdsourcing application* and *crowdsourcing platform*. With the aid of content analysis, all definitions were grouped in different perspectives of CSSs (Bortz & Döring, 2009). Furthermore, keywords were collected which either addressed a component or a function of a CSS. Iteratively, specific keywords were aggregated to more generic terms. Finally, a concept matrix was created that maps all relevant literature to one or more of the derived generic components and function terms.[3]

4 RESULTS

In this section, the author presents the results that were obtained by the literature review. The author first addresses the question of existing definitions of CSSs and then draws attention to several design aspects of components and functions of a CSS.

4.1 CROWDSOURCING SYSTEM DEFINITIONS (RQ 1)

By analyzing the primary studies, the author found 17 different kinds of definitions that relate to any of the terms: *crowdsourcing system*, *crowdsourcing application*, or *crowdsourcing platform* (Table 2).

[3] See also http://larshetmank.com/documents/wi2013_css_concept_matrix.pdf for more details of the concept matrix; the terms finally found are represented in Figure 3.

Table 2: Collected definitions

Article	Definition of crowdsourcing system and its assigned perspective (O, T, P, H)
DiPalantino and Vojnovic, 2009, p. 119	… exhibit a similar structure – a task is described, a reward and time period are stated, and during the period users compete to provide the best submission. At the conclusion of the period, a subset of submissions are selected, and the corresponding users are granted the reward. (P)
Doan et al., 2011, p. 87	… if it enlists a crowd of humans to help solve a problem defined by the system owners, and if in doing so, it addresses the following four fundamental challenges: How to recruit and retain users? What contributions can users make? How to combine user contributions to solve the target problem? How to evaluate users and their contributions? (P)
Franklin, Kossmann, Kraska, Ramesh, and Xin, 2011, p. 62	… creates a marketplace on which requesters offer tasks and workers accept and work on the tasks. (O)
Fraternali, Castelletti, Soncini-Sessa, Ruiz, and Rizzoli, 2012, p. 69	… has a Web interface that can be used by two kinds of people: work providers can enter in the system the specification of a piece of work they need …; work performers can enrol, declare their skills, and take up and perform a piece of work. The application manages the work life cycle: performer assignment, time and price negotiation, result submission and verification, and payment. In some cases, the application is also able to split complex tasks into microtasks that can be assigned independently …. In addition to the web interface, some platforms offer Application Programming Interfaces (APIs), whereby third parties can integrate the distributed work management functionality into their custom applications. (T, P)
Hirth, Hoßfeld, and Tran-Gia, 2013, p. 2919	Every employer needs a mediator to access the worker crowd. This mediator is called a crowdsourcing platform … (O)
Hirth, Hoßfeld, and Tran-Gia, 2011, p, 323	… offers an interface for the employer to submit his tasks and an interface for the crowd workers to submit the completed tasks. These platforms also provide a reward system which allows the employer to pay for the completed tasks. (O, T)
Hoßfeld, Hirth, and Tran-Gia, 2011, p. 142	… distributes the work submitted by an employer among the human worker resources and acts as mediator between worker and employer. (O)
Jayakanthan and Sundararajan, 2011, p. 25	… enterprise crowdsourcing applications which aim to utilize the capabilities of members within the organization itself – particularly the employees within a large company. (H)
Karger, Oh, and Shah, 2011, p. 284	… establish a market where a "taskmaster" can submit batches of small tasks to be completed for a small fee by any worker choosing to pick them up. (O)
Lofi, Selke, and Balke, 2012, p. 109	… an effective tool making human skills and intelligence accessible to machines. (H)
Ross, Irani, Silberman, Zaldivar, and Tomlinson, 2010, p. 2864	… that allows users to distribute work to a large number of workers. This work is broken down into simple, one-time tasks that workers are paid to complete. (P)
Treiber, Schall, Dustdar, and Scherling, 2011, p. 1	… distribute problem-solving tasks among a group of humans. (only weakly associated to P)
Venetis, Garcia-Molina, Huang, and Polyzotis, 2012, p. 989	… must post tasks for the humans, collect results, and cleanse and aggregate the answers provided by humans. (P)
Vukovic, 2009, p. 687	… is a trusted broker ensuring that providers successfully complete the task requests and that requestors pay for the charges. … issues authentication credentials for requestors and providers when they join the platform, stores details about skill-set, history of completed requests, handles charging and payments, and manages platform misuse. … can execute crowdsourcing requests in a number of different modes, by advertising them on the marketplace, allowing providers to bid for them, or in the form of a competition, where requestor identifies criteria to be used for selection of the winning submission. … may further allow requestors and providers to team-up. (O, T, P)
Zhai et al., 2011, p. 879	… collaborative cyberinfrastructure that can aggregate scattered resources, including both human brainpower and machine computational capacities. (H)

Zhang and van der Schaar, 2012, p. 2140	… systems where small tasks (typically on the order of minutes or seconds) and performed in exchange for rewards awarded to the users who performed them. (P)
Zhao and Zhu, 2012, p. 13	… are man-made socio-technical systems to support interaction and connectivity between people and technology in workplaces, and to reflect interaction between society's complex infrastructures and human behaviors. (H)

All definitions vary in the level of detail and address different aspects of CSSs. After labeling the definitions and integrating them to more general groups, four perspectives of CSSs were identified (Bortz & Döring, 2009):

- The *organizational perspective* (O) highlights the role of the CSS as an agent, which distributes the crowdsourcing tasks that are issued by the requesters (system owner, employer) to the potential recipients (crowd, human worker). Only definitions that explicitly state this role by using terms, such as mediator, marketplace, interface, or trusted broker, are associated to this perspective.
- The *technical perspective* (T) focuses on technical aspects of the CSS. These definitions enumerate software components, technical functions, or data objects that are generally implemented in a CSS, such as user interface, user authentication, user profiles, including skills and expertise, history tracking, payment mechanisms, quality control, workflow support or application programming interfaces (API).
- The *process perspective* (P) details actions that are usually performed to data objects or users of the CSS. As compared to the organizational perspective, the process perspective goes beyond the issue of submitting, distributing and accepting a crowdsourcing task and describes more clearly what happens inside the black-box of a CSS. Some of these actions or process steps are, for example, define task, set time period, state reward, recruit user, split task, assign task, provide contribution, combine submissions, select solution, evaluate user, or pay user.
- The *human-centric perspective* (H) emphasizes that human brainpower and collective intelligence are the main drivers of a CSS. In this perspective, the interaction between the users and the collaborative nature of the CSS plays a central role.

The labeling and categorization process revealed that the found definitions vary in detail and none of them covers all of the four derived perspectives. For example, whereas the definition of Vukovic (2009) addresses at least the organizational, the technical, and the process perspective, the definition of Treiber et al. (2011) is only weakly associated to the process perspective. As the quality of the development process and further theoretical contributions rely deeply on a profound definition, future research should sharpen the definition of CSSs regarding all perspectives. One first effort to detail the technical perspective is presented in the next section.

4.2 CROWDSOURCING COMPONENTS AND FUNCTIONS (RQ 2)

To further improve the understanding of CSSs, the author drew the attention to typical components and functions that may be implemented. Out of the concept matrix, as a result from the literature review, the author could derive four components: user management, task management, contribution management, and workflow management. In this section, I depicted for each component several functions that should be addressed when developing a CSS (Figure 3).

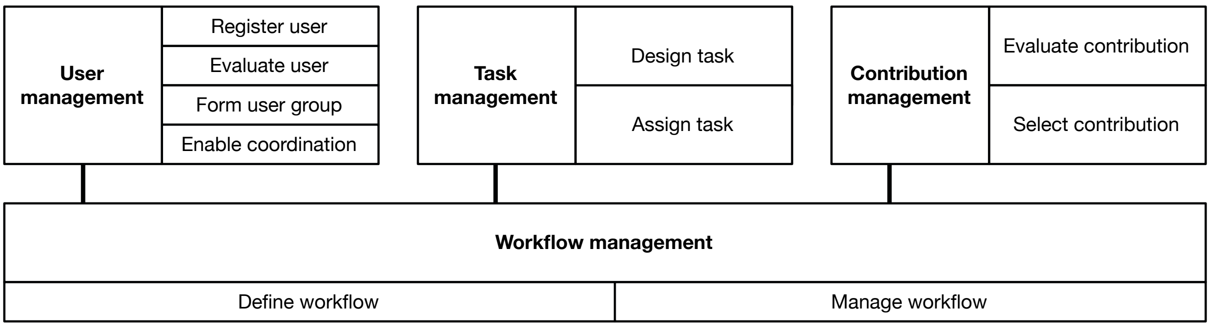

Figure 3: Components and functions of crowdsourcing systems

User Management. The first component that is worth considering in a CSS is user management that contains functions to register users, evaluate users, to form user groups for different purposes, and to establish coordination mechanisms among the users:

- *Register User.* A user profile may record both the user identity of the worker and of the requester. To improve the trust between workers and requesters, the crowdsourcing identity may also be associated with public profiles on social network sites (Klinger & Lease, 2011).
- *Evaluate User.* Users may be evaluated before they start the first task (ex-ante) or after they have finished a task (ex-post). The former applies entry questions, pre-qualification tasks or gold standard data to determine the expertise or skill level of a worker (Corney et al., 2010). The latter considers acceptance and rejection decisions of historic contributions (Mashhadi & Capra, 2011). Sometimes a certain user's answer will be directly compared to the answers of the other users responding to the same question (Karger et

al., 2011). The evaluation of a user may either be done automatically by the CSS or manually by the requester of the task. Additionally, ranking scores that presents the skill level, the reputation or the quality of the worker may be employed (Archak, 2010; Ipeirotis, Provost, & Wang, 2010; Schall, 2012).

- *Form User Group.* Different types of users are motivated differently and hence need specific incentive mechanisms (Heipke, 2010). Crowdsourcers can form either open groups that can be seen as partners of the underlying project or closed groups that get paid for their work and have mostly no benefit from the outcome (Heipke, 2010). Different types of tasks may require different amounts of people. Sometimes, only one individual per task is needed; in other cases a closed group which has specialized skills is necessary to solve the problem and again in some cases the whole open community is asked to find a solution (Fraternali et al., 2012).

- *Enable Coordination.* A CSS needs appropriate mechanisms to facilitate collaboration and coordination (Gao, Barbier, & Goolsby, 2011). On the one hand, the crowd may interact to solve the issued task collaboratively. On the other hand, direct links between the provider of the task and the crowd may be established in both directions to give feedback to the intermediate results of the crowd (from provider to crowd), and to ask for more details regarding the task specification of the provider (from crowd to provider) (Liu, Lehdonvirta, Alexandrova, & Nakajima, 2012). In this regard, the utilization of social software may support human interaction as it provides functionalities to manage personal identities, maintain relationships, share information or collaboratively document knowledge.

Task Management. The task management handles the incoming submissions of tasks and their distribution to the crowd that will solve the task. It should provide at least the following functions:

- *Design Task.* The quality of the contributions highly depends on the task design. Cheat submissions can be prevented if the task is defined appropriately (implicit crowd filtering). Thus, an important aspect is the formulation of the right question and the corresponding instructions and constraints (Corney et al., 2010). Furthermore, the type (e.g., straightforward, novel), the size, the reward or incentive scheme (DiPalantino & Vojnovic, 2009; Liu, Alexandrova, & Nakajima, 2011), the submission time, the latency (e.g., immediate, waitable) (Bernstein, Brandt, Miller, & Karger, 2011; Liu et al., 2010), the degree of confidentiality and the designated crowd should be carefully defined (Eickhoff & Vries, 2013; Hirth et al., 2011). Additionally, the requester's user profile and other contextual information, such as the location or time may be automatically assigned to the task specification. This information may support the interpretation of the task by the crowd. To further assist the task definition procedure, a CSS may also provide information about previous projects to the requester (Shao, Shi, Xu, & Liu, 2012) or knowledge that is gained by applying social network analysis techniques to the existing crowd network (Fraternali et al., 2012). Another important issue when designing a task lies in the question of how a task should be modularized in subtasks or vice versa bundled in a compound task, so it can be efficiently processed by the crowd (Kazai, Kamps, Koolen, & Milic-Frayling, 2011). Finally, a requester may configure if the contributions of the solver can be seen by the other users or not (Aparicio, Costa, & Braga, 2012).

- *Assign Task.* Allocating the right task to the right person at the right time is a key issue for the success of crowdsourcing projects. A task may either be sent to a single person, to a selected group or to the whole crowd. Intelligent task routing, where workers are selected based on the task specification and the user profile, becomes important when a large number of tasks have to be handled (Govindaraj, K.V.M., Nandi, Narlikar, & Poosala, 2011). Two aspects have to be considered when assigning a task to the crowd. The first one denotes to the question of if the worker has sufficient skills and knowledge to accomplish the task, and the second one aims for choosing an appropriate point of time when the worker can or is willing to work (Liu et al., 2010).

Contribution Management. The contribution management heavily relates to quality control and contains functions that evaluate, pre-process, combine and select solutions of the crowd:

- *Evaluate Contribution.* Evaluation plays a central role in providing feedback to the task solver in order to increase quality as well as in selecting the best result from a large set of solutions. Several aspects have to be considered when designing an effective feedback or evaluation mechanism (Dow, Kulkarni, Klemmer, & Hartmann, 2012). First, the source has to be specified, which may be the solver himself (self-assessment), a person from the crowd or the proposer of the task (external assessment). Next, the specificity of evaluation may be a simple accept or reject answer, a filled assessment form with predefined questions or a custom response as free text. Finally, when considering the time aspect, feedback can be given simultaneously while the workers are still involved in the task, or asynchronously after the task is completed.

- *Select Contribution.* Several methods may be used to detect cheat submissions and to sustain quality of the final result, such as the majority decision or the control group approach proposed by Hirth et al. (2013). The majority decision approach assigns the task to multiple users who submit their individual result to the CSS, and finally selects the result that was mostly returned. In contrast, the control group approach assigns the task only to one worker who completes the task. Afterwards, the CSS sends to the control group multiple validation tasks with the request to rate the submitted solution. The solution will be accepted if the majority of the control group decides it is correct. There exist several other crowdsourcing algorithms (e.g., sort, join, max) that model the performance of a CSS and have to be carefully designed (Marcus, Wu, Karger, Madden, & Miller, 2011; Venetis et al., 2012). Furthermore, various data processing techniques, such as data mining or machine learning algorithms may be applied to pre-process, select and combine results that are often noisy and comprise redundant data (Barbier, Zafarani, Gao, Fung, & Liu, 2012).

Workflow Management. A workflow management component is of crucial importance when designing complex tasks with global requirements and constraints (H. Zhang et al., 2012), and helps to secure contribution quality (Zhai et al., 2011). A workflow management system comprises the following functions:

- *Define Workflow.* A workflow coordinates among the inputs and the outputs of independent human or machine functions in order to get an optimal result (H. Zhang et al., 2012). Workflows are either defined by the requester of the task or the crowd itself (Kulkarni, Can, & Hartmann, 2012).

- *Manage Workflow.* The definition of crowdsourcing tasks requires experimentation of different influence parameters such as latency, the delay between issuing and commencing the task, the price of the work done, the quality of workers and contributions, and time that is needed to complete a specific task (Kittur, Khamkar, André, & Kraut, 2012). There are often several iterations required to find an efficient crowdsourcing workflow that combines the issued task, the contributions of the crowd and powerful crowdsourcing algorithms. A graphical representation of the workflow may support the creation process by serving as a mental model of a task flow (Kittur et al., 2012).

5 MAIN INSIGHTS, LIMITATIONS, AND FUTURE RESEARCH

The purpose of this paper was to gain a better understanding of what CSSs are and what typical design aspects have to be considered in the development of such systems. Therefore, this study aimed first to give an overview of how the term CSS is defined in scientific literature, and second, to derive typical components and functions of CSSs. After reviewing several definitions of CSSs, the author identified four perspectives on CSSs: the organizational, the technical, the functional, and the human-centric. In the second part of the results, the author drew attention to design aspects of generic components and functions that are usually incorporated by CSSs.

Several main insights were gained during the SLR and categorization process. First of all, the found definitions of CSSs are heterogeneously defined in the literature. They cover different aspects as the mapping between the definitions and the four perspectives showed it. They also vary in detail within each of the perspectives. For example, within the technical perspective, none of the definitions described the broad range of functions and software components that are implemented in a CSS. Therefore, future research should focus on the development of an integrated CSS definition that covers all the needed aspects for a structured development process. Moreover, while tacking a closer look at the technical perspective of CSSs by categorizing the found literature according to typical functions and components that are implemented in a CSS, it was noticed that there exists a high dependency between the identified elements that are currently not well represented, for example, the evaluation of a contribution directly affects the rating of the user and determines the reward. Hence, an accurate and complete description of a CSS has also to consider these interdependencies, which needs further investigation.

When critically reflecting this work, two issues are worth mentioning. First, the current diversity of CSSs, which are found in practice and described in research literature, makes it difficult to derive a unified list of components and functions that are usually implemented in a CSS. Nevertheless, the recent strong interest of the companies in CSSs requires not only knowing how crowdsourcing works and where it is applied, but also how it is technically implemented. Therefore, the components and functions proposed in this work may be used as a checklist and may guide decision makers, software developers and managers to better crowdsourcing solutions. Second, the result heavily relies on theoretical scientific literature and thus momentarily lacks insights from practice. Therefore, the found components and the incorporated functions should be contrasted to business case studies and real practical examples, and be refined or adjusted where applicable. However, the results of this paper are a decent starting point to get a deeper understanding of the technical nature of CSSs.

With the aid of the results of this work, the next step in future research will encompass the design of a semantic model for corporative knowledge-intensive problem solving in crowdsourcing environments. The model will focus on aspects to enhance automation in CSSs, to improve data portability between different CSSs, and to connect CSSs with other business application software.

REFERENCES

Aparicio, M., Costa, C. J., & Braga, A. S. (2012). Proposing a system to support crowdsourcing. In *Proceedings of the Workshop on Open Source and Design of Communication* (pp. 13–17). New York, NY, USA: ACM. doi:10.1145/2316936.2316940

Archak, N. (2010). Money, glory and cheap talk: analyzing strategic behavior of contestants in simultaneous crowdsourcing contests on TopCoder.com. In *Proceedings of the 19th international conference on World wide web* (pp. 21–30). New York, NY, USA: ACM. doi:10.1145/1772690.1772694

Barbier, G., Zafarani, R., Gao, H., Fung, G., & Liu, H. (2012). Maximizing benefits from crowdsourced data. *Computational & Mathematical Organization Theory*, 1–23.

Bernstein, M. S., Brandt, J., Miller, R. C., & Karger, D. R. (2011). Crowds in two seconds: enabling realtime crowd-powered interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (pp. 33–42). New York, NY, USA: ACM. doi:10.1145/2047196.2047201

Bortz, J., & Döring, N. (2009). *Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler*. *Springer-Lehrbuch* (4th ed.). Heidelberg: Springer.

Brabham, D. C. (2008). Crowdsourcing as a model for problem solving: an introduction and cases. *Convergence: The International Journal of Research into New Media Technologies*, *14*(1), 75–90. doi:10.1177/1354856507084420

Brereton, P., Kitchenham, B. a., Budgen, D., Turner, M., & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, *80*(4), 571–583. doi:10.1016/j.jss.2006.07.009

Corney, J. R., Torres-Sánchez, C., Jagadeesan, A. P., Yan, X. T., Regli, W. C., & Medellin, H. (2010). Putting the crowd to work in a knowledge-based factory. *Advanced Engineering Informatics*, *24*(3), 243–250. doi:10.1016/j.aei.2010.05.011

DiPalantino, D., & Vojnovic, M. (2009). Crowdsourcing and all-pay auctions. In *Proceedings of the 10th ACM Conference on Electronic Commerce* (pp. 119–128). New York, NY, USA: ACM. doi:10.1145/1566374.1566392

Doan, A., Ramakrishnan, R., & Halevy, A. Y. (2011). Crowdsourcing systems on the World-Wide Web. *Communications of the ACM*, *54*(4), 86–96. doi:10.1145/1924421.1924442

Dow, S., Kulkarni, A., Klemmer, S., & Hartmann, B. (2012). Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work* (pp. 1013–1022). New York, NY, USA: ACM. doi:10.1145/2145204.2145355

Eickhoff, C., & Vries, A. de. (2013). Increasing cheat robustness of crowdsourcing tasks. *Information Retrieval*, *16*(2), 121–137.

Erickson, T. (2011). Some thoughts on a framework for crowdsourcing. In *Workshop on Crowdsourcing and Human Computation* (pp. 1–4).

Estellés-Arolas, E., & González-Ladrón-de-Guevara, F. (2012). Towards an integrated crowdsourcing definition. *Journal of Information Science*, *38*(2), 189–200. doi:10.1177/0165551512437638

Franklin, M. J., Kossmann, D., Kraska, T., Ramesh, S., & Xin, R. (2011). CrowdDB: answering queries with crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data* (pp. 61–72). New York, NY, USA: ACM. doi:10.1145/1989323.1989331

Fraternali, P., Castelletti, A., Soncini-Sessa, R., Ruiz, C. V., & Rizzoli, A. E. (2012). Putting humans in the loop: social computing for water resources management. *Environmental Modelling & Software*, *37*, 68–77. doi:10.1016/j.envsoft.2012.03.002

Gao, H., Barbier, G., & Goolsby, R. (2011). Harnessing the crowdsourcing power of social media for disaster relief. *Intelligent Systems, IEEE*, *26*(3), 10–14. doi:10.1109/MIS.2011.52

Geiger, D., Seedorf, S., Schulze, T., Nickerson, R., & Schader, M. (2011). Managing the crowd: towards a taxonomy of crowdsourcing processes. In *Proceedings of the 17th Americas Conference on Information Systems* (pp. 1–11).

Govindaraj, D., K.V.M., N., Nandi, A., Narlikar, G., & Poosala, V. (2011). MoneyBee: towards enabling a ubiquitous, efficient, and easy-to-use mobile crowdsourcing service in the emerging market. *Bell Labs Technical Journal*, *15*(4), 79–92. doi:10.1002/bltj.20473

Greengard, S. (2011). Following the crowd. *Communications of the ACM*, *54*(2), 20–22. doi:10.1145/1897816.1897824

Hammon, L., & Hippner, H. (2012). Crowdsourcing. *Wirtschaftsinformatik*, *54*(3), 165–168.

Heipke, C. (2010). Crowdsourcing geospatial data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *65*(6), 550–557. doi:10.1016/j.isprsjprs.2010.06.005

Hirth, M., Hoßfeld, T., & Tran-Gia, P. (2011). Anatomy of a crowdsourcing platform – using the example of Microworkers.com. In *Proceedings of the 5th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)* (pp. 322–329). doi:10.1109/IMIS.2011.89

Hirth, M., Hoßfeld, T., & Tran-Gia, P. (2013). Analyzing costs and accuracy of validation mechanisms for crowdsourcing platforms. *Mathematical and Computer Modelling*, *57*(11–12), 2918–2932. doi:10.1016/j.mcm.2012.01.006

Hoßfeld, T., Hirth, M., & Tran-Gia, P. (2011). Modeling of crowdsourcing platforms and granularity of work organization in future internet. In *Proceedings of the 23rd International Teletraffic Congress* (pp. 142–149).

Howe, J. (2006). The rise of crowdsourcing. *Wired Magazine*, (14), 1–5. Retrieved from http://www.clickadvisor.com/downloads/Howe_The_Rise_of_Crowdsourcing.pdf

Ipeirotis, P. G., Provost, F., & Wang, J. (2010). Quality management on Amazon Mechanical Turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation* (pp. 64–67). New York, NY, USA: ACM. doi:10.1145/1837885.1837906

Jayakanthan, R., & Sundararajan, D. (2011). Enterprise crowdsourcing solutions for software development and ideation. In *Proceedings of the 2nd International Workshop on Ubiquitous Crowdsouring* (pp. 25–28). New York, NY, USA: ACM. doi:10.1145/2030100.2030108

Karger, D. R., Oh, S., & Shah, D. (2011). Budget-optimal crowdsourcing using low-rank matrix approximations. In *Proceedings of the 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)* (pp. 284–291). doi:10.1109/Allerton.2011.6120180

Kazai, G., Kamps, J., Koolen, M., & Milic-Frayling, N. (2011). Crowdsourcing for book search evaluation: impact of hit design on comparative system ranking. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 205–214). New York, NY, USA: ACM. doi:10.1145/2009916.2009947

Kazman, R., & Chen, H.-M. (2009). The metropolis model a new logic for development of crowdsourced systems. *Communications of the ACM*, *52*(7), 76–84. doi:10.1145/1538788.1538808

Kitchenham, B. (2007). *Guidelines for performing systematic literature reviews in software engineering*.

Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology*, *51*(1), 7–15. doi:10.1016/j.infsof.2008.09.009

Kittur, A., Khamkar, S., André, P., & Kraut, R. (2012). CrowdWeaver: visually managing complex crowd work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work* (pp. 1033–1036). New York, NY, USA: ACM. doi:10.1145/2145204.2145357

Klinger, J., & Lease, M. (2011). Enabling trust in crowd labor relations through identity sharing. In *Proceedings of the American Society for Information Science and Technology* (Vol. 48, pp. 1–4). doi:10.1002/meet.2011.14504801257

Kulkarni, A., Can, M., & Hartmann, B. (2012). Collaboratively crowdsourcing workflows with Turkomatic. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (pp. 1003–1012). New York, NY, USA: ACM. doi:10.1145/2145204.2145354

Leimeister, J. M., Huber, M., Bretschneider, U., & Krcmar, H. (2009). Leveraging crowdsourcing: activation-supporting components for IT-based ideas competition. *Journal of Management Information Systems*, *26*(1), 197–224. doi:10.2753/MIS0742-1222260108

Liu, Y., Alexandrova, T., & Nakajima, T. (2011). Gamifying intelligent environments. In *Proceedings of the 2011 International ACM Workshop on Ubiquitous Meta User Interfaces* (pp. 7–12). New York, NY, USA: ACM. doi:10.1145/2072652.2072655

Liu, Y., Lehdonvirta, V., Alexandrova, T., & Nakajima, T. (2012). Drawing on mobile crowds via social media. *Multimedia Systems*, *18*(1), 53–67.

Liu, Y., Lehdonvirta, V., Kleppe, M., Alexandrova, T., Kimura, H., & Nakajima, T. (2010). A crowdsourcing based mobile image translation and knowledge sharing service. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia* (pp. 1–9). New York, NY, USA: ACM. doi:10.1145/1899475.1899481

Lofi, C., Selke, J., & Balke, W.-T. (2012). Information extraction meets crowdsourcing: a promising couple. *Datenbank-Spektrum*, *12*(2), 109–120.

Malone, T. W., Laubacher, R., & Dellarocas, C. (2010). The collective intelligence genome. *Engineering Management Review, IEEE*, *38*(3), 38–52. doi:10.1109/EMR.2010.5559142

Marcus, A., Wu, E., Karger, D., Madden, S., & Miller, R. (2011). Human-powered sorts and joins. In *Proceedings of the VLDB Endowment* (Vol. 5, pp. 13–24). VLDB Endowment.

Mashhadi, A. J., & Capra, L. (2011). Quality control for real-time ubiquitous crowdsourcing. In *Proceedings of the 2nd International Workshop on Ubiquitous Crowdsouring* (pp. 5–8). New York, NY, USA: ACM. doi:10.1145/2030100.2030103

Ross, J., Irani, L., Silberman, M. S., Zaldivar, A., & Tomlinson, B. (2010). Who are the crowdworkers? Shifting demographics in Mechanical Turk. In *Proceedings of the 28th of the International Conference on Human factors in Computing Systems* (pp. 2863–2872). New York, NY, USA: ACM. doi:10.1145/1753846.1753873

Rouse, A. C. (2010). A preliminary taxonomy of crowdsourcing. In *Proceedings of the 21st Australasian Conference on Information Systems*. Brisbane.

Schall, D. (2012). Expertise ranking using activity and contextual link measures. *Data & Knowledge Engineering*, *71*(1), 92–113. doi:10.1016/j.datak.2011.08.001

Schenk, E., & Guittard, C. (2011). Towards a characterization of crowdsourcing practices. *Journal of Innovation Economics*, *7*(1), 93–107. doi:10.3917/jie.007.0093

Shao, B., Shi, L., Xu, B., & Liu, L. (2012). Factors affecting participation of solvers in crowdsourcing: an empirical study from China. *Electronic Markets*, *22*(2), 73–82.

Treiber, M., Schall, D., Dustdar, S., & Scherling, C. (2011). Tweetflows: flexible workflows with Twitter. In *Proceedings of the 3rd International Workshop on Principles of Engineering Service-Oriented Systems* (pp. 1–7). New York, NY, USA: ACM. doi:10.1145/1985394.1985395

Venetis, P., Garcia-Molina, H., Huang, K., & Polyzotis, N. (2012). Max algorithms in crowdsourcing environments. In *Proceedings of the 21st International Conference on World Wide Web* (pp. 989–998). New York, NY, USA: ACM. doi:10.1145/2187836.2187969

Vukovic, M. (2009). Crowdsourcing for enterprises. In *Proceedings of the 2009 Congress on Services - I* (pp. 686–692). Los Angeles: IEEE. doi:10.1109/SERVICES-I.2009.56

Vukovic, M., & Bartolini, C. (2010). Towards a research agenda for enterprise crowdsourcing. In T. Margaria & B. Steffen (Eds.), *Proceedings of the 4th International Conference on Leveraging Applications of Formal Methods, Verification, and Validation (ISoLA'10)* (Vol. 6415, pp. 425–434). Heraclion: Springer.

Wieringa, R., Maiden, N., Mead, N., & Rolland, C. (2006). Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Engineering*, *11*(1), 102–107.

Yuen, M.-C., King, I., & Leung, K.-S. (2011). A survey of crowdsourcing systems. In *Proceedings of the International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing* (pp. 766–773). doi:10.1109/PASSAT/SocialCom.2011.203

Zhai, Z., Sempolinski, P., Thain, D., Madey, G., Wei, D., & Kareem, A. (2011). Expert-citizen engineering: "Crowdsourcing" skilled citizens. In *Proceedings of the 9th International Conference on Dependable, Autonomic and Secure Computing (DASC)* (pp. 879–886). doi:10.1109/DASC.2011.148

Zhang, H., Law, E., Miller, R., Gajos, K., Parkes, D., & Horvitz, E. (2012). Human computation tasks with global constraints. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems* (pp. 217–226). New York, NY, USA: ACM. doi:10.1145/2207676.2207708

Zhang, Y., & van der Schaar, M. (2012). Reputation-based incentive protocols in crowdsourcing applications. In *Proceedings of the 31st International Conference on Computer Communications* (pp. 2140–2148). doi:10.1109/INFCOM.2012.6195597

Zhao, Y., & Zhu, Q. (2012). Evaluation on crowdsourcing research: current status and future direction. *Information Systems Frontiers*, 1–18.

A SYNOPSIS OF ENTERPRISE CROWDSOURCING LITERATURE

Lars Hetmank

Abstract. In the past few years, researchers have provided a desirable sense of clarity regarding the general term crowdsourcing and what it constitutes. However, with its emergence, several derivatives of the term have appeared in scientific literature. This research article focuses on enterprise crowdsourcing as one of the recent derivatives, which, due to its ambiguity, requires further discussion and clarification. Thus, the article aims to reveal the various nuances of how the term enterprise crowdsourcing is interpreted by diverse scholars. As the term has now gained reasonable momentum in available crowdsourcing literature, it is time to reflect. In this work, a systematic literature review is applied to survey different explanations of the term and to derive its constitutional characteristics. Additionally, the article provides an overview of crowdsourcing applications deployed in an enterprise context for both primary and support activities of the value-added chain. Finally, this paper concludes with suggestions of how to prevent misinterpretation and what key questions should be addressed in future research.

Note. This research article was published as Hetmank, L. (2014). A Synopsis of Enterprise Crowdsourcing Literature. In 22nd European Conference on Information Systems. Tel Aviv.

TABLE OF CONTENTS

LIST OF TABLES

1 INTRODUCTION

Since the first appearance of the term *crowdsourcing* in 2006, researchers have provided several contributions to clarify its meaning (Estellés-Arolas & González-Ladrón-de-Guevara, 2012; Howe, 2006). Along with its emergence, several derived concepts of the term appeared in scientific crowdsourcing literature. These derivatives aim to accommodate the peculiarities of different types of crowdsourcing contributions (e.g., crowdfunding, crowdvoting, crowdcreation), application domains (e.g., citizen science, crowdtesting), technical environments (e.g., mobile crowdsourcing), or organizational settings (e.g., enterprise crowdsourcing). Citizen science, for example, describes the use of crowdsourcing principles in the domain of scientific research, mainly by tapping into a crowd of amateurs and non-professional scientists (Hand, 2010). Mobile crowdsourcing, in contrast, focuses on the technical capabilities and features of mobile devices to harness the power of the crowd for certain use cases (Govindaraj, K.V.M., Nandi, Narlikar, & Poosala, 2011; Gupta, Thies, Cutrell, & Balakrishnan, 2012). When placing the general concept of crowdsourcing in the context of an enterprise that seeks to gain profit, literature often refers to the term *enterprise crowdsourcing*. Unfortunately, this term seems to be defined too vaguely, or it may be understood in more than one way. Thus, this work intends to shed light on the various nuances of the term *enterprise crowdsourcing* and to develop a framework that helps researchers clarify their own understanding and perception of the term.

The main objective of this research paper is to explore and analyze the current body of literature in order to get a consolidated view of the different perspectives and applications and to identify the key characteristics of enterprise crowdsourcing. As there is no clear and broad consensus on what the term *enterprise crowdsourcing* constitutes, the article strives to answer the following research question: What are the constitutional properties that make enterprise crowdsourcing unique compared to other types of crowdsourcing? The first step in answering the question is to adopt a theoretical perspective and analyze how the research community interprets the term *enterprise crowdsourcing* in their field of research (Section 4.1). The second step is to adopt a practice-oriented perspective to determine what types of application domains are typically associated with the term *enterprise crowdsourcing* (Section 4.2).

The next section lays the terminological foundations for this survey by briefly reviewing the concept of crowdsourcing and related terms. Section 3 introduces the methodology of the systematic literature review in answering the aforementioned research question. The findings of the review are presented in section 4, and they are then discussed in section 5. The work concludes with suggestions of how to prevent misinterpretation of the term *enterprise crowdsourcing*, and it states key questions that should be addressed in future research.

2 TERMINOLOGICAL FOUNDATION

The term *crowdsourcing* was first coined by Howe (2006), a contributing editor at Wired magazine. He defined the term as "the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call." Since that time, an increasing public and academic interest has been shown in the crowdsourcing business model.[1] Howe's definition highlights three key prerequisites to harness the benefits of crowdsourcing. First, the crowdsourcing task must be solvable by a large group. Second, the requester must have access to a large group of people that work either collaboratively or independently toward a solution. Third, the requester needs to attract these people to engage in a crowdsourcing task via an open call. This is mainly achieved by the use of social software applications and Web 2.0 technologies (Saxton, Oh, & Kishore, 2013).

From an etymological point of view, the neologism crowdsourcing is composed of the term *crowd*, which refers to a large group of workers who have gathered together to participate in an event – for example, a crowd gathered to watch something or to protest about something – and the term *sourcing*, which denotes a number of purchasing strategies aimed at finding, selecting, and engaging providers of goods and services (Longman, 2009). Therefore, crowdsourcing is heavily related to *outsourcing* practices in general and to *business process outsourcing* (BPO) in particular. Whereas outsourcing focuses on subcontracting parts of activities of the supply chain to independent suppliers (Voigt, Lackes, & Spiepermann, 2013), BPO puts emphasis on business processes as the core objects to be moved from inside the organization to an external provider (Duening & Click, 2005). Similarly, crowdsourcing shares the notion of outsourcing tasks to external agents. In the case of outsourcing, however, the agents are not predetermined, and they are mostly unknown.

[1] The search interest of the term *crowdsourcing* was determined with Google Trends (http://www.google.com/trends/, requested on 1st of December 2013).

Aside from Howe's initial definition, various other explanations have been suggested in related literature to characterize crowdsourcing. Recently, Estellés-Arolas and González-Ladrón-de-Guevara (2012) consolidated these diverse views to promulgate an integrated and consistent description of crowdsourcing. They define *crowdsourcing* in the following way:

> Crowdsourcing is a type of participative online activity in which an individual, an institution, a non-profit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of the task, of variable complexity and modularity, and in which the crowd should participate bringing their work, money, knowledge and/or experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while the requester will obtain and utilize to their advantage what the user has brought to the venture, whose form will depend on the type of activity undertaken. (Estellés-Arolas & González-Ladrón-de-Guevara, 2012, p. 197)

Although their proposed definition seems to be slightly cumbersome, it helps prevent misinterpretation, and it distinguishes the term from other concepts with similar meaning. In particular, their definition distinguishes itself from the concepts of collective intelligence, wisdom of crowds, commons-based peer production, human (-based) computation, open innovation, and open source.

As Leimeister (2010) states, *collective intelligence* is not a new concept, and it has been used by scientists to explain phenomena where humans or animals coordinate themselves to achieve a common goal. Hence, the term emphasizes the inherent decision-making abilities of large groups. Malone, Laubacher, and Dellarocas (2010) broadly define collective intelligence as "groups of individuals doing things collectively that seem intelligent" (p. 2). The conditions that characterize wise crowds and lead to collective intelligence have been widely discussed in the book *The Wisdom of Crowds*, and they comprise diversity of opinion, independence, decentralization, and aggregation (Surowiecki, 2005). Due to the fact that collective intelligence is more widely understood as a general term to describe situations in which large group of individuals make better, more informed decisions and choices than individuals or a group of experts, it can be considered as a superset of crowdsourcing (Quinn & Bederson, 2011). However, it must be noted that crowdsourcing, as opposed to collective intelligence, inherently takes more of a technology-oriented view rather than a socially-oriented view, in which group behavior plays a major role.

Another concept that closely relates to crowdsourcing may be described with the term *commons-based peer production*. Commons-based peer production depends on decentralized information gathering and exchange to lower the uncertainty of participants (Benkler, 2002). It differs from market-based production in that it is based on price mechanisms and firm-based production that relies on managerial hierarchies. To make commons-based peer production work for large-scale projects, three prerequisites are necessary (Benkler, 2002). First, the projects must be divisible into smaller modules that can be processed independently. Second, each of these modules must be relatively fine-grained so that the participants can self-select one according to their interest and motivation. Third, all contributions of the participants require easy and low-cost, often automated integration into a whole end result. Although both concepts, commons-based peer production and crowdsourcing, expect a large group of individuals, crowdsourcing – as opposed to commons-based peer production – not only focuses on tasks performed collaboratively, but also on tasks accomplished independently by individuals (Howe, 2006).

At first glance, the term *human computation* seems synonymous with crowdsourcing. Ahn (2005) defines it as "a paradigm for utilizing human processing power to solve problems that computers cannot yet solve" (p. 3). Thus, human computation and crowdsourcing both highlight the important role of humans in performing a task. However, compared to crowdsourcing, human computation puts the focus on replacing computers with humans and not on replacing traditional workers with an undefined large group of people (Quinn & Bederson, 2011).

Several authors widely investigated the relationship between crowdsourcing and concepts that emphasize aspects of openness, such as *open source* and *open innovation* (Rouse, 2010; Schenk & Guittard, 2011). Chesbrough, Vanhaverbeke, and West (2006) describe the concept of *open innovation* as "the use of purposive inflows and outflows of knowledge to accelerate internal innovation, and expand the markets for external use of innovation, respectively" (p. 1). Thus, the main idea of open innovation is that a company should not only rely on internally generated knowledge to support innovation processes but also on external knowledge sources. As opposed to open innovation, crowdsourcing might be used for open innovation initiatives, but it is not limited to such.

The term *open source* refers mainly to software in which the source code is available to the general public. Diverse, mostly geographically distributed developers – some of them paid and some of them volunteers – create the source code in a collaborative manner. Open source aims to form a counterpart to proprietarily developed software that is owned by a certain company. Although the idea of open source is mainly used for software development, there is ongoing research on how to apply the principles of open source to other application areas (Brabham, 2008). The notion of open source is also part of the crowdsourcing paradigm. However, the main difference between crowdsourcing and open source is how the company makes use of the intellectual property. In crowdsourcing, the intellectual property of a contribution is usually transferred to the company that issued the task, whereas open source licenses grant the right to copy, change, and redistribute.

Table 1 summarizes the intersecting and distinguishing properties of each related concept of crowdsourcing.

Table 1: Intersecting and distinguishing properties of related concepts of crowdsourcing

Related concept	Intersecting property	Distinguishing property
Outsourcing / Business process outsourcing (BPO)	Sourcing organizational tasks to external agents	Predetermined and known agents instead of an undefined large group of people
Collective intelligence / Wisdom of crowds	Shift from the individual to the collective	Takes a socially-oriented view rather than a technically-oriented view that is based on social software and Web 2.0 technologies and does not necessarily require an external crowd
Commons-based peer production	Refers to a new problem-solving and production model that harnesses the power of large numbers of individuals	Puts emphasis on tasks performed collaboratively
Human (-based) computation	Applies human processing power to solve problems	Replaces computers instead of traditional workers with an undefined large crowd
Open innovation	Uses external resources to improve the organizational innovativeness and efficiency	Focuses primarily on innovation processes
Open source	Denotes a decentralized production model based on a mostly geographically-distributed workforce	Refers mainly to software development in which the intellectual property is usually not transferred to the company

Out of the discussion about the term *crowdsourcing* and its related concepts, some basic principles and implications for establishing a better and more accurate understanding of the term *enterprise crowdsourcing* can be adopted. First, to harness the potentials of crowdsourcing, social and individual aspects should be taken into account, for example, how to motivate the crowd to participate or, if required, how to support collaboration among contributors. Second, in addition to social aspects, crowdsourcing focuses strongly on technical aspects. Therefore, crowdsourcing must consider the utilization of modern ICT systems that are based on social software and Web 2.0 technologies. Third, on an organizational level, the bottom-up approach of crowdsourcing must be aligned with the prevailing top-down goals of the company.

If crowdsourcing is a new production and problem-solving model based on features of social software and Web 2.0 technologies that harness the power of a large group of undefined people working either collaboratively or independently towards a common goal, what makes crowdsourcing unique in an enterprise context compared to other general-purpose or non-profit crowdsourcing applications, such as Wikipedia,[2] FoldIt,[3] or Ushahidi,[4] and thereby justifying the emergence of the term *enterprise crowdsourcing*?

3 METHODOLOGY

This article conducts a systematic literature review to answer the aforementioned research question and to gain a deeper theoretical understanding of the enterprise crowdsourcing domain. A systematic literature review provides researchers with a repeatable and well-structured procedure to identify, assess, and interpret relevant literature for a certain research objective (Webster & Watson, 2002).

Based on the principles of a systematic literature review, a search strategy should be derived after formulating the research question. Defining the search strategy consisted of determining the population, selecting the search resources, identifying the search terms, and defining several inclusion and exclusion criteria. Conference proceedings and journal papers were sought out according to the population. Only full papers that were accessible through the database subscription of the library were included. With respect to search resources, the following databases were queried: ACM Digital Library, Ebscohost (Academic Search Complete and Business Source Complete), Emerald, IEEE Xplore Digital Library, Sage Journals, ScienceDirect, SpringerLink, and Wiley. Based on the research questions, the following search strings were derived after several test queries were conducted: *Enterprise*

[2] Wikipedia can be found at http://www.wikipedia.org/.
[3] Foldit can be found at https://fold.it/.
[4] Ushahidi can be found at http://www.ushahidi.com/.

crowdsourcing, *business crowdsourcing*, and *corporate crowdsourcing*. The search result contained 69 articles (Table 2).

Table 2: Consulted databases and selected results

Consulted databases	Search results			Selected results		
	Enterprise Crowd-sourcing	Business Crowd-sourcing	Corporate Crowd-sourcing	Enterprise Crowd-sourcing	Business Crowd-sourcing	Corporate Crowd-sourcing
ACM Digital Library	12	3	1	7	0	1
Science Direct	2	1	0	2	0	0
Ebscohost	1	0	0	1	0	0
IEEE Xplore Digital Library	22	0	1	9	0	1
Emerald Insight	0	0	0	0	0	0
Wiley Online Library	1	0	0	0	0	0
Springer Link	18	5	1	8	2	1
SAGE Journals	1	0	0	0	0	0
Total	57	9	3	27	2	3
Total (all search terms)	69			32		
Total (without duplicates)				29		
Total (Google scholar cross-check)				33		

After preparing the search strategy of the systematic literature review, the selection of relevant studies in the search result was accompanied by three steps. The first step was studying the abstracts, introductions, and conclusions of each article to get an initial impression of the relevance and to sort out those articles that provided less or no contribution toward answering the research question. Second, Google Scholar was accessed to carry out a cross-check on the preliminarily selected results. Those results were supplemented with four additional articles. Finally, 33 publications (28 conference and 5 journal articles) were identified as relevant and were carefully read to record definitions of the term *enterprise crowdsourcing*, characteristics of enterprise crowdsourcing, and typical application domains of enterprise crowdsourcing (Table 2). Although these 33 research publications were a good starting point to answer the research question, additional resources were required that were not part of the systematic literature review, but which nevertheless helped to understand the concept of enterprise crowdsourcing.

4 RESULTS

This section presents the results of the systematic literature review. Subsection 4.1 focuses on the usage of the term *enterprise crowdsourcing* in scientific literature, and subsection 4.2 presents applications of enterprise crowdsourcing along the value-added chain.

4.1 CHARACTERISTICS OF ENTERPRISE CROWDSOURCING

The systematic literature review revealed that some confusion exists about what the term *enterprise crowdsourcing* actually means. Especially, some interpretations undermine or contradict the initial perception of crowdsourcing, which would be that crowdsourcing has the inherent property of outsourcing tasks to an external crowd (Howe, 2006). When analyzing the literature, two types of attempts to explain the very nature of enterprise crowdsourcing can be identified (Table 3). The first one limits the target audience to whom the crowdsourcing task is available to employees only (narrow definition). The second one focuses on enterprises as the source of potential crowdsourcing tasks and does not restrict the target group (broad definition).

Table 3: A selection of articles that describe the concept of enterprise crowdsourcing

Article	Interpretation of enterprise crowdsourcing	Type
Gassenheimer, Siguaw, and Hunter, 2013, p. 205	… a business entity's "use of an enthusiastic crowd or loosely bound public" to voluntarily provide solutions via online technology to the organization's problems.	broad
Hirth, Hoßfeld, and Tran-Gia, 2013, p. 2920	… the work is not done by a huge anonymous crowd, but by a crowd of company employees or employees of sub-contractors. Still the work is submitted to a pool of workers instead to a designated one, but using a verified crowd even confidential tasks can be crowdsourced.	narrow
Jayakanthan and Sundararajan, 2012, p. 178	… posits the use of crowdsourcing in the enterprise to "access scalable workforce on-line".	-
Jayakanthan and Sundararajan, 2011, p. 25	… tackle problems within enterprises – large business organizations … involve attracting the attention of individuals outside the organization and members of the general public, to solve problems and present solutions for the organization, … aim to utilize the capabilities of members within the organization.	broad
Lykourentzou, Vergados, Papadaki, and Naudet, 2013, p. 94	Corporate crowdsourcing occurs when crowdsourcing is applied, instead of web workers, to the human network of a company.	narrow
Skopik, Schall, and Dustdar, 2012, p. 299	… takes the usual concept of crowdsourcing on the Web and applies it to an enterprise collaboration context.	narrow

Stewart, Huerta, and Sader, 2009, p. 50	… distinction between two kinds of crowdsourcing: inside the firewall (within the enterprise, available to only its employees) [enterprise crowdsourcing], and outside the firewall (open to the general public) [public domain crowdsourcing].	narrow
Villarroel and Reis, 2010, p. 2	Intra-Corporate Crowdsourcing (ICC) refers to the distributed organizational model used by the firm to extend problem-solving to a large and diverse pool of self selected contributors beyond the formal internal boundaries of a multi-business firm: across business divisions, bridging geographic locations, leveling hierarchical structures.	narrow
Vukovic, Laredo, and Rajagopal, 2010, p. 461	… applicability of crowdsourcing methodology within the enterprise, thereby engaging internal networks of knowledge experts.	narrow
Vukovic, Laredo, Ruan, Hernandez, and Rajagopal, 2013, p. 984	… a process where a group of network-connected experts solve problems.	narrow

Stewart, Huerta, and Sader (2009), for example, take the view of the narrow definition and make a distinction between enterprise and public domain crowdsourcing. They argue that in enterprise crowdsourcing, the issued crowdsourcing task is only available to employees inside the firewall of a company, whereas in public domain crowdsourcing, the crowdsourcing task is also open to the general public outside the firewall of a company. Similarly, Hirth, Hoßfeld, and Tran-Gia (2013) note that compared to the original concept of crowdsourcing in which the work is completed by a large anonymous crowd, in enterprise crowdsourcing the crowd is formed by employees of the company or by sub-contractors. The authors precisely remark that the crowd of employees are somewhat verified and can be harnessed for business-critical and confidential tasks. However, the second part of their interpretation raises the question if employees of the partners, suppliers, and strategic alliances of the company can also be counted as part of the enterprise's crowd. Further details of which characteristics determine enterprise crowdsourcing are discussed in the work by Skopik, Schall, and Dustdar (2012). They emphasize that in collaborative enterprise crowdsourcing environments experts can, to some extent, be preselected and flexibly involved in ongoing tasks. Additionally, they argue that especially complex business tasks require mechanisms to support active coordination and collaboration between crowd members.

Gassenheimer, Siguaw, and Hunter (2013) draw attention to the idea that enterprise crowdsourcing aims to solve organizational problems or to serve business purposes instead of focusing on satisfying social or scientific demands. Likewise, Jayakanthan and Sundararajan (2011) describe enterprise crowdsourcing as "the use of crowdsourcing approaches to tackle problems within enterprises – large business organizations … [and] this may involve attracting the attention of individuals outside the organization and members of the general public" (p. 25). In this broad definition of enterprise crowdsourcing, the authors focus more on the enterprise as the source of the crowdsourcing task and do not restrict the size of the potential crowd to employees only. They discern that whether to harness the collective intelligence and workforce inside the company, outside the company, or both, may depend on the type and goal of the task. For example, if confidentiality is an issue, the company may restrict the target audience to employees, or in other words, to an internal crowd only. In contrast, the company may focus exclusively on the general public to avoid entrenched ways of thinking and to exploit the creative potential of a huge external crowd.

4.2 APPLICATION DOMAINS OF ENTERPRISE CROWDSOURCING

The reasons of a company for tapping into the power of the crowd are manifold and address aspects of cost reduction, time saving, and quality improvement (Vukovic et al., 2010). One reason is that crowdsourcing offers a way for organizations to get access to a large, globally distributed pool of workers with diverse skills, experiences, and knowledge, as well as an availability of 24 hours a day, seven days a week (O'Neill, Roy, Grasso, & Martin, 2013). Another motive – especially when engaging people outside the company – is to reduce personnel and equipment costs (Erickson & Trauth, 2013). Moreover, a company may use a large group of workers to minimize product development and service delivery time (Jayakanthan & Sundararajan, 2011).

When skimming through the literature of enterprise crowdsourcing, a variety of applications are launched along the entire value-added chain and range from accomplishing knowledge-intensive tasks, over creating user-generated content, to filtering and ranking data or content items (Sobczak & Groß, 2010). Enterprise crowdsourcing contains examples of applications for primary as well as support activities of the value-added chain. These applications consider both internal and external crowds.

According to primary activities, typical examples of enterprise crowdsourcing comprise launching innovation initiatives, developing and testing software, or solving geometric problems.

- *Launching innovation initiatives.* Firms have experimented with both internal and external innovation competitions (Jouret, 2009). The advantage in launching internal innovation initiatives through the mechanisms of crowdsourcing is twofold. First, companies can overcome their formal organizational boundaries and harvest the unexploited creative ability of all employees while protecting the undesired flow of intellectual property to competitors. Second, they can promote the formation of new or the maintenance of existing communities.

- *Developing and testing software.* The software developer's utilization rate in most of the large IT enterprises is far from perfect. Often developers wait for incoming projects, are in training, or their intellectual ability and experience do not match to the task requirement appropriately (Jayakanthan & Sundararajan, 2011). Applying a bottom-up approach to the software-development process in which employees select a task according to their interests, skills, and availability can improve their utilization rate.
- *Solving geometric problems.* Corney et al. (2010) show an example that uses crowdsourcing for visually comparing machined parts in computer-aided design and manufacturing (CAD/CAM) environments. Matching machined parts according to their similarity and creating a classification system supports the cost estimation for other machined parts and reduces the cost by reusing existing design or manufacturing information.

Several crowdsourcing applications are used to perform support activities within the company, such as applicant selection, business strategy development, process analysis, document and translation management, or knowledge acquisition.

- *Assessing resumes of job candidates.* In the domain of human resources management, enterprise crowdsourcing is applied to evaluate resumes of job applicants (Harris, 2011). As the evaluation of resumes is a repetitive, subjective, and highly labor-intensive task that cannot be processed easily by a computer algorithm, it is a perfect candidate for crowdsourcing. Like other business tasks that can be crowd-sourced, the review of resumes requires an appropriate task design and proper incentive mechanism.
- *Developing business strategies.* Opening up the strategy-development process of a company through a crowdsourcing approach allows the process to not only improve the quality of the strategy by considering the diverse viewpoints and specializations of the employees, but also to encourage enthusiasm and to establish alignment with the overall strategic direction (Gast & Zanini, 2012). Although a crowd-based approach increases overall transparency and continuously updates and evaluates the strategy of the company, special care must be taken as it sometimes leads to the undesirable effects of groupthink.

- *Conducting process analysis.* Somewhat similar to the previous example, an internal consulting department may engage the employees of a company in business process reengineering tasks. Therefore, the collective intelligence of the enterprise is used to conduct a process analysis with the goal of identifying and improving the existing model of processes and organizations. Keeping business-critical knowledge inside the company is the main advantage of this approach (Khasraghi & Tarokh, 2012).
- *Supporting document and translation management.* Further examples are found in the area of document management and processing. Enterprise crowdsourcing is applied, on the one hand, to validate and correct erroneous text modules of scanned documents that are processed by optical character recognition (OCR) programs (Karnin, Walach, and Drory, 2010) and, on the other hand, to create and translate technical documents (Stewart et al., 2009; Vukovic, Salapura, & Rajagopal, 2013).
- *IT inventory management and end-user support.* Finally, enterprise crowdsourcing solutions are used for knowledge acquisition in the domain of information technology (IT) inventory management and end-user support (Vukovic et al., 2010; Vukovic, Lopez, & Laredo, 2010; Vukovic & Naik, 2011). These solutions manage virtual teams of knowledge workers on demand to address knowledge-intensive tasks. The results often lead to a consolidated view of a particular knowledge domain.

5 DISCUSSION

The review of literature on enterprise crowdsourcing revealed that researchers use the term *enterprise crowdsourcing* in two different ways. The first way is in a broad sense in which the target group is not restricted. The second is in a narrow sense in which the target group is limited to the employees of a company (Section 4.1). As a consequence of this ambiguity and to prevent misunderstanding, researchers and practitioners should refer more consistently to intra-corporate (employees only), inter-corporate (contracted freelancers, partners, subcontractors, strategic alliances of the company), or corporate crowdsourcing (members of the public domain) to denote a certain target audience, and they should only use the term *enterprise crowdsourcing* (in a broad sense) to summarize activities in which all three target groups are addressed. Table 4 suggests a set distinguishing aspects to separate the two extremes of intra-corporate crowdsourcing and corporate crowdsourcing.

Table 4: Comparing intra-corporate with corporate crowdsourcing

Comparison criteria	Intra-corporate crowdsourcing	Corporate crowdsourcing
Target group	• Employees	• Public-domain
Job roles and formal relationships	• Known	• Mostly unknown
Task type	• Often complex, knowledge-intensive tasks	• Predominately simple tasks
Suitability	• Mainly for large and international companies	• Also possible for small and medium-sized companies
Opportunities	• Assigning a verified crowd to business-critical and confidential tasks • Utilizing free working capacity • Using existing business relationships, networks and communities, and organizational units for recommending crowdsourcing tasks	• Reduction of personnel and equipment costs • Decreasing product-development and service-delivery time • Benefit of having a larger pool of workers compared to relying only on an internal workforce
Risks	• Reaching the critical numbers of contributors to accomplish a crowdsourcing task due to the limited size and heterogeneity of the internal crowd • Jeopardizing traditional formal work settings	• Unwanted crowdsourcing activities of a crowd that is not easy to control • Legal aspects, such as loss of intellectual property, or issues of data privacy and security concerns • Difficult to integrate the crowdsourcing activity into the prevailing hierarchical organization or business processes, especially aligning the top-down approach of the organization with the bottom-up approach of crowdsourcing

Intra-corporate crowdsourcing seems to bear some characteristics that are worth highlighting. First, intra-corporate crowdsourcing focuses more on complex, knowledge-intensive tasks rather than on simple tasks (Lykourentzou et al., 2013). Second, due to the complexity of these tasks, active coordination between the crowd members is often required (Skopik et al., 2012). Third, a crowd that consists of employees only might be more reliable and trustworthy than an external crowd. Thus, they are more appropriate to solve confidential tasks (Hirth et al., 2013). Finally, the company usually knows the job roles – that is, the skills and the availability of its employees – and can therefore easily recommend a crowdsourcing task to a worker based on these criteria.

Although intra-corporate crowdsourcing obviously has some special characteristics and seems to be important for large, international companies, we clearly have to demarcate intra-corporate crowdsourcing from other concepts that also focus on distributed work within companies via ICT, such as computer-supported collaborative work (CSCW) or virtual teams. CSCW investigates how technology can support humans who collaboratively work together (Koch & Gross, 2006). A virtual team comprises individuals that are distributed geographically and across organizational units and who, enabled by ICT, work asynchronously together toward a common purpose (Schweitzer & Duxbury, 2010). The main differences between intra-corporate crowdsourcing and these concepts, however, are that the crowd member must not necessarily work collaboratively in teams and that the tasks will not be pre-assigned to a group of workers in advance. However, the CSCW research community is aware that they should focus not only on groupware that supports the collaborative work of small and medium-sized groups, but also on so-called crowdware that also considers large crowds (Schneider, Moraes, Moreira de Souza, & Esteves, 2012).

Interestingly, the shift from the generic concept of crowdsourcing to the more specific form of enterprise crowdsourcing closely resembles the discussion of the transformation from the free and open source software (FOSS) phenomenon to the commercial form of OSS 2.0 (Fitzgerald, 2006). Similar to OSS 2.0, enterprise crowdsourcing – both in the narrow sense and in the broad sense – requires experts with the necessary knowledge to address business needs efficiently. It also requires rigorous support of project management and quality control to integrate the sourced task into the complex business processes of the company. A further analogy can be drawn between the idea of transferring the open source principles to an inner or corporate source of software developers within an organization and the idea of applying the crowdsourcing approach to employees inside a large organization (intra-corporate crowdsourcing). While the requester-participant relationship in crowdsourcing might not be as close as the user-developer relationship in open source, both concepts share a sense of joint adventure among the people involved that is not common in the traditional production models of the company (Fitzgerald, 2006).

Regardless of the assembly of the target group (Section 4.1) and the supported business activities (Section 4.2), the term *enterprise crowdsourcing* is mainly used in the context of for-profit organizations. These organizations seek to improve their profits by accessing the crowd to perform strategic, administrative, or operational tasks that are normally accomplished by a designated employee or group of employees. Additionally, by overcoming formal organizational, geographical, and temporal boundaries, organizations strive to reduce both costs and execution time and to improve the quality of these tasks. To sum up, the main constitutional characteristic of enterprise crowdsourcing is that it serves the business needs of an organization and, as a consequence, must be deliberately measured.

6 CONCLUSION

To avoid any misunderstanding of the term *enterprise crowdsourcing* and to improve the discussion among researchers and practitioners, authors who publish in the field of enterprise crowdsourcing are encouraged to make clear and explicit statements about:

- Who is part of the target group (e.g., intra-corporate crowdsourcing, inter-corporate crowdsourcing, or corporate crowdsourcing)?
- Which strategic, administrative, or operational tasks does the crowd perform? Which business needs does the crowd address?
- Where is the enterprise crowdsourcing platform hosted (e.g., using in-house, on premises solution vs. crowdsourcing intermediaries, such as Amazon Mechanical Turk)?

In addition to clarifying the specialties of enterprise crowdsourcing, the following key questions to guide future research activities in the field of enterprise crowdsourcing should be addressed:

Key Question 1: How should an organization form and attract a critical mass of appropriate contributors within and beyond the boundaries of the organization?

One crucial prerequisite to enable crowdsourcing is the availability of a large number of potential participants. Sometimes, this is also referred to as the phenomenon of positive network effects because the value of a service increases when more participants are attracted and engaged in a crowdsourcing activity. However, this is only partly true. Often it is not just the size of the crowd that enables the success of a crowdsourcing initiative; rather, that the crowd is the right crowd is the key factor that enables the success of an initiative. Thus, agreeing to the narrow understanding of enterprise crowdsourcing (intra-corporate crowdsourcing), two questions basically arise: First, how many and what kind of employees does a company need to attract? And second, how heterogeneous do the employees have to be according to knowledge, skills, and experiences to solve a certain problem efficiently? As this may depend more or less on the type of a task, there is certainly a minimum threshold. It can be assumed that this critical level is not reached by most of the small and medium-sized enterprises (SMEs).

Key Question 2: How may formal organizational, geographical, and temporal boundaries be overcome?

A second precondition to tap into the power of crowdsourcing both inside and outside the company is the overcoming of organizational, geographical, and temporal boundaries. Villarroel and Reis (2010) describe enterprise crowdsourcing precisely as a "distributed organizational model used by the firm to extend problem-solving to a large and diverse pool of self-selected contributors beyond the formal internal boundaries of a multi-business firm: across business divisions, bridging geographic locations, leveling hierarchical structures" (p. 2). Therefore, on a technical level, ongoing research is required to investigate the effects of social software and Web 2.0 technologies on overcoming these boundaries. However, it also raises the question of how to deal with cultural differences among the crowd.

Key Question 3: How can a company efficiently allocate their financial, human, and technical resources to support a crowdsourcing activity?

La Veccia and Cisternino (2010) call attention to an efficient allocation of financial, human and technical resources. They notice that enterprise crowdsourcing must assure that these resources are not wasted and that the performance of each task is carefully evaluated. Moreover, they point out that current crowdsourcing business models fall short of adequately addressing most of the complex business processes of an enterprise. Therefore, they suggest that a new business model for enterprise crowdsourcing is required. This model must consider a wide range of aspects of outsourcing complex internal business processes to the crowd. These aspects are, for example, the adequate allocation of resources and the control of the process regarding delivery time, quality, and cost (O'Neill et al., 2013).

Key Question 4: How can the crowdsourcing initiative be integrated with existing knowledge activities and business processes?

Several authors have discussed the importance of integrating the crowdsourcing process with the existing business processes of the company (Vukovic et al., 2010). Most of the current crowdsourcing platforms, however, are not able to manage the complex workflow of subtasks efficiently (Khazankin, Satzger, & Dustdar, 2012). Further, as the crowd engages in these tasks voluntarily and is not directly assigned to them, only the modification of parameters, such as the type of incentive, number of contributors, or the available time, may influence the participation rate. To enhance automation and interoperability in an enterprise crowdsourcing environment, crowdsourcing systems should also be integrated into the prevailing business and social software applications of the company. The deployment of semantic vocabularies and web standards may offer a good starting point.

Key Question 5: How should the level of risk associated with legal aspects be handled?

When engaging in crowdsourcing activities, enterprises should be aware of legal aspects, such as data security regulations, copyright ownership, patent law, or employment law (Wolfson & Lease, 2011). First, exposing user or customer data to the crowd poses the risk of violating data security regulations. Further difficulties arise in joint inventorship that may compromise patents that an enterprise already holds. Particularly challenging is the distinction between co-inventors who significantly contribute to the conception of the invention and participants who simply work under the guidance of an inventor. Similar problems occur with the issue of copyrights and losing control of the work, especially when the crowd jointly contributes copyrightable parts to the whole product. In order to take advantage of the copyright ownership, enterprises should make sure that workers are considered employees who create works made for hire. This requires complex negotiation processes and mechanisms to allow a company to control and manage the transfer of intellectual property (Aitamurto, Leiponen, & Tee, 2011). Finally, enterprises should pay attention not to infringe on employment law. According to the legislation, companies must consider whether the crowd workers are considered employees or not. This usually depends on the task type offered and the contractual relationship between the requester and the participant of the crowdsourcing task.

Although several outstanding issues remain in the emerging field of enterprise crowdsourcing, clarification could be provided regarding what the term *enterprise crowdsourcing* actually means. Thus, to support ongoing research, enterprise crowdsourcing was analyzed from two different angles – a conceptual perspective that examined diverse interpretations of the term and a practical perspective that explored application domains of enterprise crowdsourcing.

REFERENCES

Ahn, L. Von. (2005). *Human Computation*. Carnegie Mellon University.

Aitamurto, T., Leiponen, A., & Tee, R. (2011). *The promise of idea crowdsourcing – benefits, contexts, limitations*. *ideasproject.com* (Vol. 1, pp. 1–30).

Benkler, Y. (2002). Coase's penguin, or, Linux and the nature of the firm. *Yale Law Journal*, *112*(3), 369–446.

Brabham, D. C. (2008). Crowdsourcing as a model for problem solving. *Convergence: The International Journal of Research into New Media Technologies*, *14*(1), 75–90. doi:10.1177/1354856507084420

Chesbrough, H., Vanhaverbeke, W., & West, J. (2006). *Open Innovation: researching a new paradigm*. OUP Oxford.

Corney, J. R., Torres-Sánchez, C., Jagadeesan, A. P., Yan, X. T., Regli, W. C., & Medellin, H. (2010). Putting the crowd to work in a knowledge-based factory. *Advanced Engineering Informatics*, *24*(3), 243–250. doi:10.1016/j.aei.2010.05.011

Duening, T. N., & Click, R. L. (2005). *Essentials of business process outsourcing*. *Essentials series*. Hoboken, NJ: Wiley.

Erickson, L. B., & Trauth, E. M. (2013). Getting work done: evaluating the potential of crowdsourcing as a model for business process outsourcing service delivery. In *Proceedings of the 2013 Annual Conference on Computers and People Research* (pp. 135–140). New York, NY, USA: ACM. doi:10.1145/2487294.2487323

Estellés-Arolas, E., & González-Ladrón-de-Guevara, F. (2012). Towards an integrated crowdsourcing definition. *Journal of Information Science*, *38*(2), 189–200. doi:10.1177/0165551512437638

Fitzgerald, B. (2006). The transformation of open source software. *MIS Quarterly*, *30*(3), 587–598.

Gassenheimer, J., Siguaw, J., & Hunter, G. (2013). Exploring motivations and the capacity for business crowdsourcing. *AMS Review*, *3*(4), 205–216. doi:10.1007/s13162-013-0055-8

Gast, A., & Zanini, M. (2012). The social side of strategy. *McKinsey Quarterly*, (2), 82–93.

Govindaraj, D., K.V.M., N., Nandi, A., Narlikar, G., & Poosala, V. (2011). MoneyBee: towards enabling a ubiquitous, efficient, and easy-to-use mobile crowdsourcing service in the emerging market. *Bell Labs Technical Journal*, *15*(4), 79–92. doi:10.1002/bltj.20473

Gupta, A., Thies, W., Cutrell, E., & Balakrishnan, R. (2012). mClerk: enabling mobile crowdsourcing in developing regions. In *Proceedings of the 2012 ACM annual* *conference on Human Factors in Computing Systems* (pp. 1843–1852). New York, NY, USA: ACM. doi:10.1145/2208276.2208320

Hand, E. (2010). People, power, change: movements of social transformation. *Nature*, *466*, 685–687. doi:10.1038/news.2010.106

Harris, C. G. (2011). You're Hired ! An examination of crowdsourcing incentive models in human resource tasks. In *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining* (pp. 15–18).

Hirth, M., Hoßfeld, T., & Tran-Gia, P. (2013). Analyzing costs and accuracy of validation mechanisms for crowdsourcing platforms. *Mathematical and Computer Modelling*, *57*(11–12), 2918–2932. doi:10.1016/j.mcm.2012.01.006

Howe, J. (2006). Crowdsourcing: a definition. Retrieved July 16, 2013, from http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html

Jayakanthan, R., & Sundararajan, D. (2011). Enterprise crowdsourcing solutions for software development and ideation. In *Proceedings of the 2nd International Workshop on Ubiquitous Crowdsouring* (pp. 25–28). New York, NY, USA: ACM. doi:10.1145/2030100.2030108

Jayakanthan, R., & Sundararajan, D. (2012). Enterprise crowdsourcing solution for software development in an outsourcing organization. In A. Harth & N. Koch (Eds.), *Proceedings of the 11th international conference on Current Trends in Web Engineering* (Vol. 7059, pp. 177–180). Berlin, Heidelberg: Springer.

Jouret, G. (2009). Inside Cisco's search for the next big idea. *Harvard Business Review*, *87*(9), 43–45.

Karnin, E., Walach, E., & Drory, T. (2010). Crowdsourcing in the document processing practice. In F. Daniel & F. Facca (Eds.), *Proceedings of the 10th international conference on Current trends in web engineering* (Vol. 6385, pp. 408–411). Berlin, Heidelberg: Springer.

Khasraghi, H. J., & Tarokh, M. J. (2012). Efficient business process reengineering with crowdsourcing. *International Journal of Applied Information Systems (IJAIS)*, *2*(7).

Khazankin, R., Satzger, B., & Dustdar, S. (2012). Optimized execution of business processes on crowdsourcing platforms. In *Proceedings of the 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing* (pp. 443–451).

Koch, M., & Gross, T. (2006). Computer-supported cooperative work – concepts and trends. In *Proceedings of the 11th Conference of the Association Information and Management (AIM)*. Luxembourg.

La Vecchia, G., & Cisternino, A. (2010). Collaborative workforce, business process crowdsourcing as an alternative of BPO. In F. Daniel & F. Facca (Eds.), *Proceedings of the 10th International Conference on Current Trends in Web Engineering* (Vol. 6385, pp. 425–430). Springer.

Leimeister, J. (2010). Collective intelligence. *Business & Information Systems Engineering*, *2*(4), 245–248. doi:10.1007/s12599-010-0114-8

Longman, P. (2009). *Dictionary of Contemporary English*. Pearson/Longman.

Lykourentzou, I., Vergados, D., Papadaki, K., & Naudet, Y. (2013). Guided crowdsourcing for collective work coordination in corporate environments. In C. Bădică, N. Nguyen, & M. Brezovan (Eds.), *Proceedings of the 5th International Conference on Computational Collective Intelligence Technologies and Applications* (Vol. 8083, pp. 90–99). Berlin, Heidelberg: Springer. doi:10.1007/978-3-642-40495-5_10

Malone, T. W., Laubacher, R., & Dellarocas, C. (2010). *Harnessing crowds: mapping the genome of collective intelligence* (No. 4732-09). *MIT Sloan School Working Paper 4732-09* (Vol. 1, pp. 1–21).

O'Neill, J., Roy, S., Grasso, A., & Martin, D. (2013). Form digitization in BPO: from outsourcing to crowdsourcing? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 197–206). New York, NY, USA: ACM. doi:10.1145/2470654.2470683

Quinn, A. J., & Bederson, B. B. (2011). Human computation: survey and taxonomy of a growing field. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)* (pp. 1403–1412). Vancouver.

Rouse, A. C. (2010). A preliminary taxonomy of crowdsourcing. In *Proceedings of the 21st Australasian Conference on Information Systems*. Brisbane.

Saxton, G. D., Oh, O., & Kishore, R. (2013). Rules of crowdsourcing: models, issues, and systems of control. *Information Systems Management*, *30*(1), 2–20. doi:10.1080/10580530.2013.739883

Schenk, E., & Guittard, C. (2011). Towards a characterization of crowdsourcing practices. *Journal of Innovation Economics*, *7*(1), 93–107. doi:10.3917/jie.007.0093

Schneider, D., Moraes, K., Moreira de Souza, J., & Esteves, M. G. P. (2012). CSCWD: Five characters in search of crowds. In *Proceedings of the 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD)* (pp. 634–641). doi:10.1109/CSCWD.2012.6221885

Schweitzer, L., & Duxbury, L. (2010). Conceptualizing and measuring the virtuality of teams. *Information Systems Journal*, *20*(3), 267–295.

Skopik, F., Schall, D., & Dustdar, S. (2012). Discovering and managing social compositions in collaborative enterprise crowdsourcing systems. *International Journal of Cooperative Information Systems*, *21*(4), 297–341.

Sobczak, S., & Groß, M. (2010). *Crowdsourcing: Grundlagen und Bedeutung für das E-Business* (1st ed.). Boizenburg: Werner Hülsbusch.

Stewart, O., Huerta, J. M., & Sader, M. (2009). Designing crowdsourcing community for the enterprise. In *Proceedings of the ACM SIGKDD Workshop on Human Computation –*

HCOMP '09 (pp. 50–53). New York, New York, USA: ACM Press. doi:10.1145/1600150.1600168

Surowiecki, J. (2005). *The Wisdom of Crowds*. New York: Anchor.

Villarroel, J. A., & Reis, F. (2010). Intra-corporate crowdsourcing (ICC): leveraging upon rank and site marginality for innovation. In *Proceedings of CrowdConf*. San Francisco.

Voigt, K.-I., Lackes, R., & Spiepermann, M. (2013). Outsourcing. *Gabler Wirtschaftslexikon*. Retrieved July 29, 2013, from http://wirtschaftslexikon.gabler.de/Archiv/54709/outsourcing-v9.html

Vukovic, M., Laredo, J., & Rajagopal, S. (2010). Challenges and experiences in deploying enterprise crowdsourcing service. In B. Benatallah, F. Casati, G. Kappel, & G. Rossi (Eds.), *Proceedings of the 10th International Conference on Web Engineering* (Vol. 6189, pp. 460–467). Berlin, Heidelberg: Springer.

Vukovic, M., Laredo, J., Ruan, Y., Hernandez, M., & Rajagopal, S. (2013). Assessing service deployment readiness using enterprise crowdsourcing. In *Proceedings of the International Symposium on Integrated Network Management (IM 2013)* (pp. 984–989).

Vukovic, M., Lopez, M., & Laredo, J. (2010). PeopleCloud for the globally integrated enterprise. In A. Dan, F. Gittler, & F. Toumani (Eds.), *Proceedings of the 7th International Joint Conference on Service Oriented Computing (ICSOC&ServiceWave)* (Vol. 6275, pp. 109–114). Berlin, Heidelberg: Springer.

Vukovic, M., & Naik, V. K. (2011). Managing enterprise IT systems using online communities. In *Proceedings of the International Conference on Services Computing (SCC)* (pp. 552–559). doi:10.1109/SCC.2011.112

Vukovic, M., Salapura, V., & Rajagopal, S. (2013). Crowd-enabled technical writing services. In *Proceedings of the International Conference on Services Computing (SCC)* (pp. 635–642). doi:10.1109/SCC.2013.84

Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: writing a literature review. *MIS Quarterly*, *26*(2).

Wolfson, S. M., & Lease, M. (2011). Look before you leap: legal pitfalls of crowdsourcing. *Proceedings of the American Society for Information Science and Technology*, *48*(1), 1–10.

Towards a Semantic Standard for Enterprise Crowdsourcing – A Scenario-based Evaluation of a Conceptual Prototype

Lars Hetmank

Abstract. To cut expenses and save time, enterprise crowdsourcing is more and more used to disseminate corporate tasks, which are traditionally performed by a small group of people, to an undefined large workforce within and beyond the boundaries of a company. However, harnessing the positive effects of crowdsourcing faces several challenges, such as the efficient and proper assignment of a crowdsourcing task to an available and competent group of workers, or the securing of the integration and reuse of crowdsourcing data across heterogeneous business applications. To overcome these challenges, a semantic vocabulary for enterprise crowdsourcing is proposed and its applicability is shown by evaluating it against three diverse scenarios that may occur in real business environments. The vocabulary includes fifteen semantic elements to describe a crowdsourcing task and eight elements to define a crowdsourcing user.

Note. This research article was published as Hetmank, L. (2013). Towards a Semantic Standard for Enterprise Crowdsourcing – A Scenario-based Evaluation of a Conceptual Prototype. In 21st European Conference on Information Systems. Utrecht.

TABLE OF CONTENTS

LIST OF TABLES

1 INTRODUCTION

Crowdsourcing is a vigorous research area and a powerful mechanism for outsourcing tasks that are traditionally performed by designated employees to a large and undefined group of potential contributors (Das & Vukovic, 2011). Enterprise crowdsourcing in particular involves both harnessing the collective intelligence and workforce inside – across business divisions and hierarchical structures – and outside the company (Jayakanthan & Sundararajan, 2011). It can be used for a variety of applications, such as collecting and evaluating ideas, creating knowledge repositories, or collaboratively developing new products. The main advantage of crowdsourcing lies in the way how it significantly changes the business processes by harnessing the skills, knowledge, or other resources of a distributed crowd of workers to achieve an outcome at lower cost and in shorter time (Vukovic & Bartolini, 2010). The development of a crowdsourcing system as well as its integration into an existing information and communication technology (ICT) infrastructure, however, can be a risky and challenging undertaking. First, the relevant tasks have to be reallocated to an undefined large group of corporate internal and external workers. Identifying an appropriate worker or a well-organized working group to whom to propose either manually or automatically a certain crowdsourcing task is a complex process that requires a lot of additional context-sensitive information, such as the task requirements, users' qualifications, or underlying social network relationships. Second, some of the data, which are required for an efficient crowdsourcing, already exist in other business applications and should be reused. Third, several attributes of the task specification, such as the target audience, the type and nature of the reward, or the confidentiality, determine the success of a crowdsourcing initiative and should be carefully configured by the requester of the task. In order to meet these challenges, a semantic vocabulary for enterprise crowdsourcing is proposed in this article. The vocabulary includes a well-defined set of semantic elements that are commonly shared and equally understood among software developers and architects. It aims to support the automation of an enterprise crowdsourcing system as well as the interoperability with other ICT systems. As the vocabulary is based on knowledge about the best or most appropriate practices, it also helps to raise the overall quality of the enterprise crowdsourcing system that will be developed (Sommerville, 2011).

The remainder of this article is structured as follows: Crowdsourcing faces a lot of challenges that must be addressed in practice. Thus, a selection of these challenges that motivates for a standard are depicted in chapter 2. The subsequent chapter 3 describes how design-science research is applied to develop and evaluate a semantic vocabulary for enterprise crowdsourcing. After that, the two concepts of the vocabulary, the crowdsourcing task and the crowdsourcing user, are described in-depth. For each of the two core concepts, several semantic elements and corresponding values are specified. Finally, in chapter 5, three different scenarios are built to evaluate and to demonstrate the applicability of the suggested elements. The article concludes with a summary of the main insights that are derived from the scenario-based evaluation and gives future prospects in the research and standardization of enterprise crowdsourcing.

2 CHALLENGES

Current research literature in the domain of crowdsourcing poses several challenges that have to be addressed when developing crowdsourcing systems in practice. To further motivate and to emphasize the necessity for a semantic standard, five of the main challenges of enterprise crowdsourcing are explained in detail:

1. *Allocation of Tasks*. Proposing the right task to the right person at the right time is a key challenge for the success of a crowdsourcing initiative (Nielsen, 2011). In this regard, Liu et al. (2010) point out two aspects to improve the appropriateness of the task allocation: the capacity and availability. Whereas the capacity denotes to the issue if a worker has the ability to accomplish the task, the availability indicates if a worker has the time to do the work and if the task is proposed at a convenient time. Both aspects have to be considered for an intelligent task routing mechanism that suggests a crowdsourcing task to the most likely audience. This mechanism should be based on an elaborate specification of task requirements and user expertise that increase the efficiency and the quality of the provided solutions (Cosley et al. 2007). A standard for enterprise crowdsourcing may provide elements to support the semantically rich representation of data that are required for an appropriate task assignment.

2. *Dynamic Team Formation*. Group formation or self-organization of people with either similar or diverse, cross-functional skills, knowledge, or experiences is often a prerequisite to solve large and complex tasks. Unfortunately, most of the existing crowdsourcing systems fall short of facilitating the flexible, dynamic, and proactive assembly of globally distributed teams (Vukovic, 2009). A first step towards an improvement may include detailed descriptions of the workers' qualifications or

information that is derived from their social networks. A semantic standard may support the structured recording of experiences, skills and knowledge.

3. *Data Integration and Exchange*. Data integration across diverse social software, business, and crowdsourcing applications as well as data exchange between them remains a key issue for future research. Crowdsourcing solutions often require the most recent data that exist in external business applications, such as enterprise dictionaries, knowledge repositories, or expert systems (Vukovic, Laredo, & Rajagopal, 2010). Therefore, when developing crowdsourcing systems, careful attention should be paid to the seamless integration of such applications. The introduction of a generic semantic standard for enterprise crowdsourcing may support this integration.

4. *Structured Task Specification*. The quality of the contributions of the crowd is highly dependent on the quality and detail of the task design. To improve quality it is necessary to provide a structured task specification and integrate the task with other business processes (Vukovic, Lopez, & Laredo, 2010). A well-defined semantic standard may guide the issuer of a crowdsourcing task towards a better task design.

5. *Transaction Transparency*. Crowdsourcing is often a complex process that addresses diverse participants who range from amateurs to experts, requires a variety of resources, involves several incentive methods, and uses various schemes to evaluate a user as well as their contributions. Most crowdsourcing workflows require a lot of experimentation, performance evaluation, and adjustment to work efficiently (Kittur, Khamkar, André, & Kraut, 2012). Thus, to increase the success and the quality of a crowdsourcing effort, a designer of these workflows needs an appropriate degree of transparency. A semantic standard for enterprise crowdsourcing helps to improve the transparency of a crowdsourcing process. It allows tracking the status of the contributions of the crowd and provides a foundation for a clear visualization of all elements within a crowdsourcing process.

3 METHODOLOGY

In this article, the design-science research (DSR) approach is applied to develop a semantic vocabulary (design artifact) that can be used to describe two of the main concepts in an enterprise crowdsourcing process: the task and the user (Hevner, March, Park, & Ram, 2004). Data that are stored in databases of either the crowdsourcing system itself or other external applications are currently not sufficiently represented and exchanged between different crowdsourcing systems and business applications. Furthermore, an efficient mapping of submitted tasks onto available users is based on semantically rich descriptions of tasks and users (problem relevance). Therefore, to improve the allocation and self-selection process of crowdsourcing tasks and to increase the interoperability between enterprise crowdsourcing and other ICT systems, a semantic vocabulary is proposed and evaluated against heterogeneous scenarios that may occur in real business environments (research contribution and design evaluation). The rigor in this article is guaranteed from the diligent and effective use of knowledge that was gained by a previously undertaken literature review as well as from an appropriate selection of the research method, which is in this case, an evaluation through scenario building (research rigor). The reason for choosing a scenario-based evaluation as a first proof of concept lies in the fact that the development of a vocabulary is highly complex and cost intensive. However, this paper presents just the first cycle in a development process of a semantic vocabulary for enterprise crowdsourcing. Further cycles will follow to improve the applicability of the vocabulary successively. The examination of real business case studies and practical examples will give insights for future improvements. Additionally, the prototype creation of a metadata schema using a schema definition language, such as the Extensible Markup Language (XML) schema or the Resource Description Framework (RDF) schema as well as the evaluation of the prototype through creating instances of real business examples are the next necessary measures to meet the challenges (search process). This article provides results for the technical-oriented as well as the management-oriented audiences. On the one hand, software developers get a detailed description of elements and attributes that can be consulted to construct own instances of the two main concepts: the crowdsourcing task and the user. On the other hand, managers acquire the basis for decision-making towards the standardization of enterprise crowdsourcing solutions (research communication). All DSR guidelines that are addressed in this article are summarized in Table 1.

Table 1: Application of DSR guidelines according to Hevner et al., 2004

DSR guideline	Embodiment
Design as an artifact	Building a semantic vocabulary for enterprise crowdsourcing
Problem relevance	Addressing the above mentioned challenges, such as task allocation, data integration and exchange, or transaction transparency

DSR guideline	Embodiment
Research contribution	Standardization to improve the allocation and self-selection of crowdsourcing tasks as well as the interoperability between the enterprise crowdsourcing system and other business applications
Design evaluation	Scenario building is used as a method
Research rigor	Based on results of previous studies in crowdsourcing
Search process	First step in the development process of a unified semantic vocabulary for enterprise crowdsourcing; further cycles will follow to improve the applicability of the vocabulary successively
Research communication	Research results for technical-oriented and management-oriented audiences are communicated through conferences, journals and prototype implementations

4 SEMANTIC VOCABULARY

This section introduces the two core concepts of the semantic vocabulary: the crowdsourcing task (Section 4.1) and the user (Section 4.2). From an extensive study of literature in the field of crowdsourcing, fifteen elements for specifying crowdsourcing tasks and eight elements for defining crowdsourcing users are derived.

4.1 TASK CONCEPT

A meaningful *task description* is efficient for implicit crowd filtering as potential workers select tasks that are most appropriate to them (Eickhoff & de Vries, 2011). It contains initial states, detailed instructions, goals, possible constraints as well as certain acceptance criteria (Robertson, 2001). Each crowdsourcing task can be addressed exclusively to the employees of an enterprise, to the public domain, or to both the employees and the public domain. Thus, the *target audience* of enterprise crowdsourcing can be set as an internal, external or hybrid crowd (Vukovic & Bartolini, 2010). Crowdsourcing tasks often differ in *complexity* and range from mundane to complex tasks (Brabham, 2008). Other classification schemes group crowdsourcing tasks into simple, moderate and sophisticated tasks (Rouse, 2010), or in simple, complex and creative tasks (Schenk & Guittard, 2011). A division into three complexity degrees is adopted for the semantic vocabulary, namely simple, moderate and complex. Besides the level of complexity, each task is also classified regarding its *type of action* that is performed, such as share, create, evaluate, or organize (Doan, Ramakrishnan, & Halevy, 2011). Moreover, a task may be assigned directly or indirectly to the crowd. In some cases, two or more tasks are bundled to one collection before assigning it to a potential worker, and in other cases, the task is split in several subtasks so that multiple workers can process each of them independently at the same time (Vukovic & Bartolini, 2010). This aspect is indicated by the element *modularization*.

The next two elements of the task concept refer to the nature and type of the reward. The *nature of the reward* describes how the contribution of a worker is rewarded. A reward may either be fixed, such as a certain amount of money after completing a task, or performance-based, such as a prize that depends on the ranking in a competition. If no reward is stated, the task is marked as voluntary. In contrast to the nature of the task, the *type of the reward* specifies what is rewarded. On the one hand, a reward may be of immaterial value, such as providing virtual points that improve the worker's reputation, money in the form of a bonus that increases the salary, or access to a resource, which may or may not be related to the actual crowdsourcing initiative itself. On the other hand, physical goods can be chosen to compensate workers for their spent efforts and resources (Corney, Torres-Sanchez, Jagadeesan, & Regli, 2009).

Four elements of the task concept relate to the time aspect. For some tasks, such as the collaborative creation of a knowledge repository, the focus lies on the accuracy of the contribution. In this case, the *latency* between issuing a task and getting an answer to the task does not matter. These tasks are defined as waitable. In other cases, such as an instant translation during a meeting, receiving an immediate reply is critical for the quality experience of the requester (Liu et al., 2010). This element addresses particularly the increasing role of real-time crowdsourcing (Bernstein, Brandt, Miller, & Karger, 2011). In addition to the latency, the *submission time* when the task is accessible for the crowd, the *duration* of how long the task takes to complete, and the *closure time* when the task expires may be set (Hirth, Hossfeld, & Tran-Gia, 2011).

Another important issue when designing a crowdsourcing task is the choice, whether the workers can see each other's contributions. This decision regarding the *visibility* is critically for the outcome of the task, as it may either foster collaboration to incrementally approach a better solution or promote greater diversity of contributions (Aparicio, Costa, & Braga, 2012). Enterprises that want to exploit crowdsourcing also have to challenge the issue of *confidentiality* as it is one of the biggest risks when involving the public community (Corney et al., 2010). For current purposes, low and high confidential tasks are distinguished. However, if new requirements have to be met in future, the graduation will be adopted.

Crowdsourcing systems may use these two values to decide if the task and the associated documents can be shared with third parties.

Finally, the last two semantic elements point to the human requirements and the technical resources that are needed to accomplish a task. A detailed description of the human requirements and the technical resources is an inevitable prerequisite for an intelligent and automatic allocation between the task and the crowd. *Human requirements* are comprised of, for example, the job tenure, professional positions, academic titles, certificates, or other qualifications, whereas *technical resources* refer to software applications, documents or datasets (Vukovic, 2009).

All introduced elements of the task concept are summarized in the following Table 2, whereas the first column refers to the element name, the second column gives a description to the element, and the third column makes a suggestion for possible data types (string, date, time, dateTime, anyURI) or element values. The data types are derived from the XML Schema specification (Biron & Malhotra, 2004).

Table 2: Semantic elements to specify a crowdsourcing task

Element	Description	Value
Task description	A meaningful *task description* contains the instructions, the initial states, the constraints, the acceptance criteria and the goals of a task.	<string>
Target audience	The element *target audience* describes the selection of people who form the crowd. They are recruited inside the company, outside the company, or both.	intern, extern, hybrid
Complexity	The element *complexity* specifies the amount of skills, experiences and knowledge that is required to solve the task.	simple, moderate, complex
Type of action	Every task is mapped to a *type of action* that the crowd performs.	create, evaluate, organize, share
Modularization	The element *modularization* states if the task is assigned directly or indirectly to the crowd. A task is assigned indirectly by bundling several tasks to one task or by splitting one task in several subtasks beforehand.	directly, bundled, split
Nature of the reward	The element *nature of the reward* describes how a contribution is rewarded.	voluntary, fixed, performance-based
Type of the reward	The element *type of reward* specifies what is rewarded.	none, virtual points, money, goods, access to resource

Element	Description	Value
Latency	The element *latency* specifies if the answer is waitable or if an immediate reply can be expected.	immediate, waitable
Submission time	The element *submission time* states the time when the task is accessible for the crowd.	<dateTime>
Closure time	The element *closure time* sets the time when the task expires.	<dateTime>
Duration	The element *duration* specifies the approximate time required to solve the task.	<time>
Visibility	The element *visibility* configures if the problem solvers can see the contribution of other workers.	hidden, visible
Confidentiality	The element *confidentiality* classifies if the task and the associated documents can or cannot be shared with third parties.	low, high
Human requirement	The element *human requirement* contains qualifications and characteristics that are needed to fulfill the task.	<string> or <anyURI>
Technical resource	The element *technical resource* specifies sources, e.g., database feeds or existing spreadsheets that are required to accomplish the task.	<string> or <anyURI>

4.2 USER CONCEPT

The users of any crowdsourcing system are mainly divided in those who submit crowdsourcing tasks (requester, client) and those who solve these tasks (recipient, participant, crowd, worker, provider). Both types of groups have particular characteristics that should be considered for efficient enterprise crowdsourcing. The *user identity*, such as the real name or a reference to an existing public profile on social networking sites, is the first element that is taken into account to describe a user. It improves the trustworthiness of the relationship between the worker and the requester (Klinger & Lease, 2011). Furthermore, the success of many crowdsourcing efforts, such as product innovations for certain markets or translation tasks, depends on the cultural background and the language skills of the recruited users. Thus, the information about the *nationality* of the user are added to the vocabulary (Antin & Shaw, 2012). Next, finding and selecting the right experts for a crowdsourcing task is a highly nuanced and context-sensitive problem that requires, besides the user's *qualifications*, also information about the *job title*, the *entry date* (job tenure), the associated *department* and the geographic *location* (Yarosh, Matthews, & Zhou, 2012). Finally, to preserve and improve the quality of future crowdsourcing contributions, Eickhoff and de Vries (2011) propagates for a more sophisticated worker grading system than just a prior acceptance rate. The types of accomplished tasks as well as the frequency distribution of certain input types, such as check boxes or free text fields, give insights into the quality of future engagements of the worker. Therefore, the user is also characterized by his or her references to prior *accomplishments*. Table 3 recapitulates the elements that are appropriate to describe a user of an enterprise crowdsourcing system.

Table 3: Semantic elements to describe a crowdsourcing user

Element	Description	Value
User identity	The *user identity* is either a real name or a reference to an existing social networking service.	<string> or <anyURI>
Nationality	The element *nationality* describes the legal relationship between the user and a state.	<string>
Qualification	The element *qualification* defines the skills, expertise or competencies of a user. It contains references to credentials, certificates, academic degrees, or even to an entire electronic portfolio of qualifications.	<string> or <anyURI>
Job title	The element *job title* characterizes the domain expertise as well as the leading position of a user.	<string>
Entry date	The element *entry date* defines the date of joining the company. Out of this, the job tenure can be derived.	<dateTime>
Department	Each user may be associated to a *department* of the company. It determines the organizational position of a user.	<string>
Location	The element *location* describes the place where the user is currently situated in. It determines the geographical position of a user.	<string>
Accomplishment	The element *accomplishment* refers to prior completed tasks.	<anyURI>

5 EVALUATION

To demonstrate the utility of the designed artifact, three different scenarios are constructed around the semantic vocabulary. The construction of these scenarios according to the designed vocabulary is a first descriptive evaluation and proof-of-concept. Each scenario contains all elements introduced in the vocabulary. In order to show the applicability of the vocabulary, most of the elements of the task concept are used heterogeneously across the three scenarios (Table 4).

Table 4: Use of semantic elements of the task concept across three example scenarios

Element	Scenario 1	Scenario 2	Scenario 3
Task description	Evaluate product design	Translate technical specification	Build company-wide virtual library
Target audience	Hybrid	Intern	Intern
Complexity	Simple	Complex	Simple
Type of action	Evaluate	Create	Share
Modularization	10 subtasks (bundled)	Each section equals one subtask (split)	<unspecified>
Latency	Immediate	Waitable	Waitable
Nature of the reward	Fixed and performance-based	Fixed	Voluntary
Type of the reward	15 reputation points plus bonus or discount of 5 (point-based)	80 Euro (payment)	http://example-company.com/virtual-library (access to resource)
Submission time	After release	2012-09-03 9:00 am	2012-09-30 10:00 am
Closure time	After 20 reviews for each product design	2012-09-17 4:00 pm	<unspecified>
Duration	1 minute	Half an hour	<unspecified>
Confidentiality	Low	High	Low
Visibility	Hidden	Visible (company-wide)	Visible (department-wide)
Human requirement	Job tenure of more than two years OR master in engineering, product design, marketing OR sales	Native German speaker OR GDS certificate in German language	<none>
Technical resource	http://www.flickr.com/photos/new-product-xyz	https://docs.google.com/document/d/123456789/edit	http://example-company.com/virtual-library/book-form

As a subset of an example crowd, four users are introduced as shown in Table 5. Not all values are used in the description of the three scenarios below. However, the example users give the reader an idea of how the elements of the user concept are applied.

Table 5: Example users based on the semantic vocabulary

Element	User 1	User 2	User 3	User 4
User identity	Alan Coulter	Adèle Girard	Markus Schmidt	https://www.xing.com/profile/Francesco-Carlone
Location	Cork	Lyon	Berlin	Turin
Nationality	Irish	French	German	Italian
Job title	Chief product designer	Junior product engineer	Senior product engineer	Junior software developer
Entry date	1993-04-01	2010-02-09	2003-09-01	2009-05-18
Department	Product development	Product engineering	Product engineering	Software development
Qualification	Master of Product Design and Development	Bachelor of Engineering	Master of Engineering, Certificate in Quality Management	PhD in Software Engineering, Java, C++, HTML, CSS
Accomplishment	http://example-company.com/cs/task/3241	<none>	<none>	<none>

5.1 EVALUATE PRODUCT DESIGN

In this scenario, a rather simple enterprise crowdsourcing task (complexity) of evaluating several product design proposals (task description) is presented.

Alan Coulter (user identity), the chief product designer (job title) of the product development (department), requires an immediate (latency) assessment of hundreds of product design proposals that were collected inside the product design department and outside the company through an open innovation competition last month. For the evaluation task (type of action), he also wants to address both the employees inside the company and the workers of the public community (target audience). Therefore, he first uploads all pictures of the drawn prototypes to a photo sharing community (technical resource). Furthermore, he decides to bundle ten subtasks of evaluating the product design to one single task that is going to be assigned to an individual user (modularization). The crowdsourcing task takes approximately one minute to accomplish (duration), is submitted to the crowd directly after the task is released in the crowdsourcing system (submission time), and is closed when each product design has at least 20 reviews (closure time). A worker receives 15 reputation points for each bundle of subtasks that he or she finishes. Additionally, the worker gets a bonus or discount of five points if the task meets or does not meet the end result of the evaluation task (nature and type of task). As the design task is in a very early stage of the product development cycle and customer integration is highly desirable, Alan does not to worry about issues of confidentiality. However, to receive independent answers, the design rating of the crowd cannot be seen by each other (visibility). Another attempt to get high quality results is the reasoned selection of human requirements. Therefore, Alan forms a crowd of workers that have either worked at least two years within the company or have a master's degree in engineering, product design, marketing or sales. After submitting the task, Markus Schmidt, who is situated in the German office, gets an inquiry to rate ten different product design proposals as his qualifications meets the defined human requirements. Additionally, numerous external voluntary workers and freelancers with the required qualifications are requested to engage in the crowdsourcing task.

5.2 TRANSLATE TECHNICAL SPECIFICATION

The enterprise crowdsourcing process that is illustrated in this scenario is the translation of a technical specification (task description).

Adèle Girard (user identity), who recently engineered a successful product for the French market, is instructed by her supervisor to send the technical specification for further assessment to Markus Schmidt, who is responsible for the German market. Adèle's as well as Markus' level of proficiency in either of the both languages is unfortunately not sufficient enough to communicate precisely with each other. She also does not know anyone in the narrow circle of colleagues who might help her. Fortunately, she has heard of an enterprise crowdsourcing solution that was integrated in the intranet of the company last week and allows to outsource complex translation tasks to other colleagues around the world (complexity, type of task, technical resource). She soon decides to use this new application for her own purposes. For that, she first splits the translation task in several sections (modularization) and sets the target audience to internal only (target audience) due to the high confidentiality that has to be guaranteed (confidentiality). She also wants that the distributed team of translators can correct each other's sections and therefore makes the contribution visible for every translator involved in the crowdsourcing task (visibility). Furthermore, to increase the probability of interaction between the potential translators, she decides to delay the submission time to the beginning of September, when the peak time of holiday in France and Germany will be over (submission time). She estimates a processing time of half an hour for each section (duration) and keeps the translation task open for the next two weeks (closure time). She further does not expect an immediate reply (latency).

The system suggests a fixed reward of 80 euros that is added as a bonus to the current salary (nature and type of reward). To address only colleagues with an appropriate level of German, the potential worker has to be either a native German speaker with French language skills or needs to have a GDS certificate in German language (human requirement).

5.3 BUILD COMPANY-WIDE VIRTUAL LIBRARY

In the last scenario, the idea of building a company-wide virtual library is depicted (task description).

Francesco Carlone (user identity) is employed as a junior software developer (job title) in a medium-sized company that is characterized with flat hierarchies. Because of the difficult market situation, he has unfortunately little work to do and would like to educate himself to issues of economics and information systems via self-study. He believes that literature on these topics might be available in other departments, that other colleagues might also want to know about their existence, and that they will probably support him (target audience). Therefore, Francesco makes an announcement to his colleagues that he wants to record all technical books and magazines that are physically available within each of the departments and put them in a knowledge repository. He soon starts to develop a crowdsourcing system for the simple task of collecting bibliographic references (complexity and type of action). Fortunately, he knows that most of his colleagues will provide him voluntarily with the necessary information, as they will get access to the repository in return (nature and type of reward). For the moment, he sets the visibility of the data records to department-wide, so that others can correct and do not add again an already existing bibliographic item (visibility). As the new knowledge repository prospers, he already thinks about integrating additional features in the system, such as collecting interests and experiences of his colleagues and experts from outside the company as well as integrating existing social networking sites.

6 SUMMARY

The main purpose of this work is to foster the standardization in the domain of enterprise crowdsourcing by providing a first conceptual prototype. As far as the author knows, this is the first attempt towards a semantic standard that improves the allocation of crowdsourcing tasks to employees and increases the interoperability between the enterprise crowdsourcing system and other business applications. To highlight the significance of the topic and to justify the efforts of developing a semantic standard, the article starts with an overview of current challenges that have to be addressed when deploying crowdsourcing systems in business environments. After briefly describing the design-science research approach in the context of this work, the principal outcome – a semantic vocabulary for enterprise crowdsourcing – is presented. It contains the key semantic elements of two of the main concepts in any crowdsourcing activities: the crowdsourcing task and the user. To show how these elements are used in real business environments and to prove the applicability of the vocabulary, three distinct business scenarios are created. This can be referred as a first evaluation of the designed artifact. Even though the scenario-based evaluation demonstrates the general applicability of the semantic vocabulary, it still reveals some starting points for future improvements and research. First, certain elements require further refinement in their level of detail. For example, the element that describes the modularization of the crowdsourcing task consists of two sub-properties: the type (bundled, split or unspecified) and the actual value (number of subtasks). Second, the value of an element can be the result of the crowdsourcing itself, for example, the closure time can be specified not only by the time but also by the number of provided contributions. Thus, a semantic vocabulary has to facilitate the definition of conditional expressions. Third, some of the elements are currently oversimplified, although they are complex in nature. For instance, the type of reward can be either a fixed value or even a function that allows calculating a dynamic value based on a sophisticated bonus scheme.

Although the current version of the semantic vocabulary for enterprise crowdsourcing leaves room for improvement, it offers already some support for the technical-oriented as well as the management-oriented audiences. On the one hand, software developers and architects obtain detailed descriptions of elements and attributes that support the construction of their own instances of the core entities. On the other hand, managers acquire the basis for decision-making towards the standardization of enterprise crowdsourcing as the consistent representation of the proposed elements not only supports the integration with other business applications but also improves the efficient and appropriate assignment of the crowdsourcing task to the user. The current version of the vocabulary contains only the two essential concepts: the task and the user. In future development cycles, additional concepts, such as a detailed description of the users' contributions or a specification of the varied incentive mechanisms, are integrated into the vocabulary. Additionally, as this proposal is primarily based on theoretical findings that are gained from an extensive literature study, other sources, such as business case studies, expert interviews, surveys, and real practical examples, have to be considered to refine and extend the vocabulary where necessary.

REFERENCES

Antin, J., & Shaw, A. (2012). Social desirability bias and self-reports of motivation: a study of Amazon Mechanical Turk in the US and India. In *Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems* (pp. 2925–2934). New York, NY, USA: ACM. doi:10.1145/2207676.2208699

Aparicio, M., Costa, C. J., & Braga, A. S. (2012). Proposing a system to support crowdsourcing. In *Proceedings of the Workshop on Open Source and Design of Communication* (pp. 13–17). New York, NY, USA: ACM. doi:10.1145/2316936.2316940

Bernstein, M. S., Brandt, J., Miller, R. C., & Karger, D. R. (2011). Crowds in two seconds: enabling realtime crowd-powered interfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (pp. 33–42). New York, NY, USA: ACM. doi:10.1145/2047196.2047201

Biron, P. V, & Malhotra, A. (2004). XML Schema Part 2: Datatypes Second Edition.

Brabham, D. C. (2008). Crowdsourcing as a model for problem solving. *Convergence: The International Journal of Research into New Media Technologies*, *14*(1), 75–90. doi:10.1177/1354856507084420

Corney, J. R., Torres-Sanchez, C., Jagadeesan, A. P., & Regli, W. C. (2009). Outsourcing labour to the cloud. *International Journal of Innovation and Sustainable Development*, *4*(4), 294–313.

Corney, J. R., Torres-Sánchez, C., Jagadeesan, A. P., Yan, X. T., Regli, W. C., & Medellin, H. (2010). Putting the crowd to work in a knowledge-based factory. *Advanced Engineering Informatics*, *24*(3), 243–250. doi:10.1016/j.aei.2010.05.011

Cosley, D., Frankowski, D., Terveen, L., & Riedl, J. (2007). SuggestBot: using intelligent task routing to help people find work in Wikipedia. In *Proceedings of the 12th International Conference on Intelligent User Interfaces* (pp. 32–41). New York, NY, USA: ACM. doi:10.1145/1216295.1216309

Das, R., & Vukovic, M. (2011). Emerging theories and models of human computation systems: a brief survey. In *Proceedings of the 2nd International Workshop on Ubiquitous Crowdsouring* (pp. 1–4). New York, NY, USA: ACM. doi:10.1145/2030100.2030102

Doan, A., Ramakrishnan, R., & Halevy, A. Y. (2011). Crowdsourcing systems on the World-Wide Web. *Communications of the ACM*, *54*(4), 86–96. doi:10.1145/1924421.1924442

Eickhoff, C., & de Vries, A. P. (2011). How crowdsourcable is your task? In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM)* (pp. 11–14). Hong Kong.

Hevner, A., March, S., Park, J., & Ram, S. (2004). Design science in information systems research. *Mis Quarterly*, *28*(1), 75–105.

Hirth, M., Hossfeld, T., & Tran-Gia, P. (2011). Anatomy of a crowdsourcing platform – using the example of Microworkers.com. In *Proceedings of the 5th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)* (pp. 322–329). doi:10.1109/IMIS.2011.89

Jayakanthan, R., & Sundararajan, D. (2011). Enterprise crowdsourcing solutions for software development and ideation. In *Proceedings of the 2nd International Workshop on Ubiquitous Crowdsouring* (pp. 25–28). New York, NY, USA: ACM. doi:10.1145/2030100.2030108

Kittur, A., Khamkar, S., André, P., & Kraut, R. (2012). CrowdWeaver: visually managing complex crowd work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work* (pp. 1033–1036). New York, NY, USA: ACM. doi:10.1145/2145204.2145357

Klinger, J., & Lease, M. (2011). Enabling trust in crowd labor relations through identity sharing. In *Proceedings of the American Society for Information Science and Technology* (Vol. 48, pp. 1–4). doi:10.1002/meet.2011.14504801257

Liu, Y., Lehdonvirta, V., Kleppe, M., Alexandrova, T., Kimura, H., & Nakajima, T. (2010). A crowdsourcing based mobile image translation and knowledge sharing service. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia* (pp. 1–9). New York, NY, USA: ACM. doi:10.1145/1899475.1899481

Nielsen, M. (2011). *Reinventing discovery: the new era of networked science*. Princeton University Press.

Robertson, S. I. (2001). *Problem solving* (1. publ.). Hove: Psychology Press.

Rouse, A. C. (2010). A preliminary taxonomy of crowdsourcing. In *Proceedings of the 21st Australasian Conference on Information Systems*. Brisbane.

Schenk, E., & Guittard, C. (2011). Towards a characterization of crowdsourcing practices. *Journal of Innovation Economics*, *7*(1), 93–107. doi:10.3917/jie.007.0093

Sommerville, I. (2011). *Software Engineering* (9th ed.). Boston: Pearson.

Vukovic, M. (2009). Crowdsourcing for enterprises. In *Proceedings of the 2009 Congress on Services - I* (pp. 686–692). Los Angeles: IEEE. doi:10.1109/SERVICES-I.2009.56

Vukovic, M., & Bartolini, C. (2010). Towards a research agenda for enterprise crowdsourcing. In T. Margaria & B. Steffen (Eds.), *Proceedings of the 4th International Conference on Leveraging Applications of Formal Methods, Verification, and Validation (ISoLA'10)* (Vol. 6415, pp. 425–434). Heraclion: Springer.

Vukovic, M., Laredo, J., & Rajagopal, S. (2010). Challenges and experiences in deploying enterprise crowdsourcing service. In B. Benatallah, F. Casati, G. Kappel, & G. Rossi (Eds.), *Proceedings of the 10th International Conference on Web Engineering* (Vol. 6189, pp. 460–467). Berlin, Heidelberg: Springer.

Vukovic, M., Lopez, M., & Laredo, J. (2010). PeopleCloud for the globally integrated enterprise. In A. Dan, F. Gittler, & F. Toumani (Eds.), *Proceedings of the 7th International Joint Conference on Service Oriented Computing (ICSOC&ServiceWave)* (Vol. 6275, pp. 109–114). Berlin, Heidelberg: Springer.

Yarosh, S., Matthews, T., & Zhou, M. (2012). Asking the right person: supporting expertise selection in the enterprise. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems* (pp. 2247–2256). New York, NY, USA: ACM. doi:10.1145/2207676.2208382

Developing an Ontology for Enterprise Crowdsourcing

Lars Hetmank

Abstract. The improvement of the efficacy of enterprise crowdsourcing activities is heavily dependent on finding, sharing, and integrating the right information for certain use cases. These efforts may include activities, such as recommending a crowdsourcing task to a competent worker or evaluating an ongoing or completed crowdsourcing project. However, to pave the way for intelligent enterprise crowdsourcing platforms, the semantic richness of the data must be improved. Therefore, an ontology including a wide set of classes and properties is proposed in this paper. The ontology development is based on the ontology engineering methodology. A first general assessment of the ontology is given at the end of the paper, which describes how it addresses major crowdsourcing requirements.

Note. This research article was published as Hetmank, L. (2014). Developing an Ontology for Enterprise Crowdsourcing. In Multikonferenz Wirtschaftsinformatik 2014. Paderborn.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Crowdsourcing is a fertile research area and a powerful mechanism for outsourcing tasks that are traditionally performed by designated employees to a large and undefined group of potential contributors (Das & Vukovic, 2011). Enterprises use this mechanism for a variety of applications, such as collecting and evaluating ideas (Villarroel & Reis, 2010), solving geometric problems (Corney et al., 2010), creating knowledge repositories (Vukovic, Lopez, & Laredo, 2010), or collaboratively developing new products (Jayakanthan & Sundararajan, 2012). The main advantage of enterprise crowdsourcing lies in the way it significantly changes the business processes by harnessing the skills, knowledge, or other resources of a distributed crowd of workers to achieve an outcome at lower cost and in shorter time (Vukovic & Bartolini, 2010).

The development of a crowdsourcing system as well as its integration into an existing information and communication technology (ICT) infrastructure, however, can be a risky and challenging undertaking. First, the relevant tasks have to be reallocated to an undefined large group of users. Identifying the right task that matches the interest and ability of a worker or a well-organized working group is a complex process that requires a lot of additional context-sensitive information, such as the task requirements, users' qualifications, or underlying social network relationships (Schall, 2012). Second, some of the data that are required for efficient crowdsourcing, such as data of human resources or social networks, already exist in other business applications and should be reused. Third, several attributes of the task specification, such as the target audience, the type and nature of the reward, or the chosen evaluation procedure, determine the success of a crowdsourcing initiative and should be deliberately configured (Doan, Ramakrishnan, & Halevy, 2011).

In order to meet these challenges, the overall objective of this work is to develop a practically useful and lightweight ontology to capture, store, and share crowdsourcing data. The central motivation behind this technical artifact is that requesters obtain a controlled vocabulary to communicate specific details about their crowdsourcing activity. Based on this, crowdsourcing or other business applications may embed all information that is required to solve a certain crowdsourcing task into a webpage along with other existing displaying information. This will have several profound advantages. First, contributors can easily and very precisely search for tasks that correspond to their interests, skills, knowledge, or availability. Second, enterprise crowdsourcing systems but also crowdsourcing intermediaries outside the firewall of an organization may extract the crowdsourcing data that comes from other business applications, such as project management, enterprise resource planning, or social software, and use it for further processing without retyping and copying the data. Third, these systems or intermediaries are able to proactively recommend a crowdsourcing task to potential contributors, e.g., based on their social network profiles. Finally, enterprise or external semantic search engines may exploit the structured data and provide users with enhanced search, browsing, or navigation capabilities.

The remainder of this article is organized as follows: In section 2, we introduce the methodology and explain how the first version of the ontology is developed. In section 3, several requirements that an ontology for enterprise crowdsourcing must address are discussed in detail. The section concludes with a set of central questions that give further guidance for the ontology development. After that, all classes and properties of the ontology are explained in-depth in section 4. Finally, in section 5, the requirements illustrated in section 3 are revised and each of them is briefly assessed to demonstrate the applicability of the proposed ontology. The article concludes with a review of aspects that require further discussion and gives an outlook for future research.

2 METHODOLOGY

The development process of the ontology follows the ontology engineering approach proposed by Uschold and King (1995). Although this methodology has its limitations, it provides for the necessary simplicity and less overhead to support the development of a first and lightweight version of the ontology. The following steps must be performed: (i) identify the purpose and scope of the ontology, (ii) build the ontology, (iii) evaluate the ontology, and (iv) document the ontology.

Purpose and scope: In this paper, we aim to design an ontology that facilitates the structured recording of enterprise crowdsourcing data in an organized and meaningful way. Typical instruments that are beneficial to define the scope and purpose of the ontology are motivating scenarios and informal competency questions (Grüninger & Fox, 1995). In a preliminary work, three distinct business scenarios were created to demonstrate the general applicability of a potential ontology for enterprise crowdsourcing (Hetmank, 2013b): (i) evaluate product design proposals, (ii) translate a technical specification, and (iii) build a company-wide virtual library. To further support the conceptualization process of the ontology, we draw up a set of requirements and central questions (Section 3).

Build the ontology: Based on the three motivating scenarios, we derived an initial set of semantic elements, which is now extended and supported by findings of a previously undertaken review of existing crowdsourcing literature (Hetmank, 2013a). The conceptual model of the crowdsourcing ontology is visualized as a Unified Modeling Language (UML) class diagram and exemplarily implemented in the Web Ontology Language (OWL) by using the open-source ontology development editor Protégé (Section 4).

Evaluate the ontology: As a frame of reference to evaluate the ontology, we use the requirements and central questions stated in section 3. We will also give some examples how the ontology is applied in practice (Section 5).

Document the ontology: One of the main barriers to the widespread dissemination of an ontology is an inadequate documentation of the ontology. Thus, the results of the ontology development process will be provided as a well-documented specification explaining each of the classes and properties in detail.

3 REQUIREMENTS

The motivating scenarios showed that employing the right crowdsourcing data for a particular use case and working context may significantly enhance a crowdsourcing activity. Supported by literature and based on the scenarios, five key requirements are derived to guide the design of the ontology:

- *R1: Task specification.* A crowdsourcing ontology should facilitate an appropriate architectural support to define tasks (Pedersen et al., 2013; Vukovic et al., 2010).
- *R2: Task allocation.* A crowdsourcing ontology should support both the user in selecting a task (pull method) and the system in recommending a task (push method) taking aspects into account, such as the suitability of a task for a worker, the worker's availability, or the confidentiality of a task. Current crowdsourcing systems require an intelligent task routing mechanism based on an elaborate specification of task requirements and detailed user profiles (Cosley, Frankowski, Terveen, & Riedl, 2007; Nielsen, 2011).

- *R3: Team building.* A crowdsourcing ontology should offer concepts to identify existing as well as to form new working groups. The flexible, dynamic, and proactive assembly of globally distributed teams with members that have either similar or diverse cross-functional skills, knowledge, or experiences is often a prerequisite to solve large and complex tasks (Vukovic, 2009).

- *R4: Transaction transparency and quality control.* A crowdsourcing ontology should include properties that the requester of a task can consult and statistically evaluate in order to optimize the crowdsourcing activity (Kittur, Khamkar, André, & Kraut, 2012).

- *R5: Interoperability.* A crowdsourcing ontology should improve the data integration and exchange between the crowdsourcing system and other ICT systems, e.g., social software, enterprise dictionaries, knowledge repositories, or expert systems (Vukovic, Laredo, & Rajagopal, 2010).

To give some further guidance, central questions are posed for each requirement statement (Table 1).

Table 1: Requirements and central questions

<table>
<tr><th colspan="2">Requirement</th><th>Central question</th></tr>
<tr><td colspan="2">Task specification (R1)</td><td>Which semantic elements support requesters in their efforts of specifying a crowdsourcing task that will draw an audience?</td></tr>
<tr><td rowspan="2">Task allocation (R2)</td><td>Task distribution (requester-oriented)</td><td>Which semantic elements aid requesters in proposing a crowdsourcing task to an appropriate and available user with the required qualifications?</td></tr>
<tr><td>Task selection (participant-oriented)</td><td>Which semantic elements improve the self-selection of a crowdsourcing task that fits best to the participants' knowledge, skill, and experience?</td></tr>
<tr><td rowspan="2">Team building (R3)</td><td>Team identification (requester-oriented)</td><td>Which semantic elements are used to identify existing teams, working groups, or online communities of a particular knowledge domain in social networks that are suitable for a certain crowdsourcing task?</td></tr>
<tr><td>Team formation (participant-oriented)</td><td>Which semantic elements help to foster the self-formation process of the participants?</td></tr>
<tr><td colspan="2">Transaction transparency and quality control (R4)</td><td>Which semantic elements support the evaluation process of a crowdsourcing user or contribution and are essential to improve the transparency of a crowdsourcing activity according to specific roles?</td></tr>
<tr><td colspan="2">Interoperability (R5)</td><td>Which elements are required to support the interoperability between the crowdsourcing system and other business and social software applications?</td></tr>
</table>

4 ELEMENTS OF THE ONTOLOGY

In this section, we propose an ontology[1] for capturing, storing, and linking crowdsourcing data (Figure 1). The designed ontology includes 24 classes, 22 object properties, and 30 unique datatype properties to describe the main aspects of a crowdsourcing model (CSM). In the next subsections, we introduce the key conceptual entities of the crowdsourcing ontology, namely user, project, task, requirement, reward mechanism, evaluation mechanism, and contribution. To increase the human readability of the class instances, we add to some of the core classes a title and a description attribute. The title consists of a short phrase and the description offers longer sentences to describe the instance in a meaningful way. Along with the key concepts, we describe additional classes, subclasses and properties. Class names begin with capital letters and property names with lower case letters after the namespace prefix (csm).

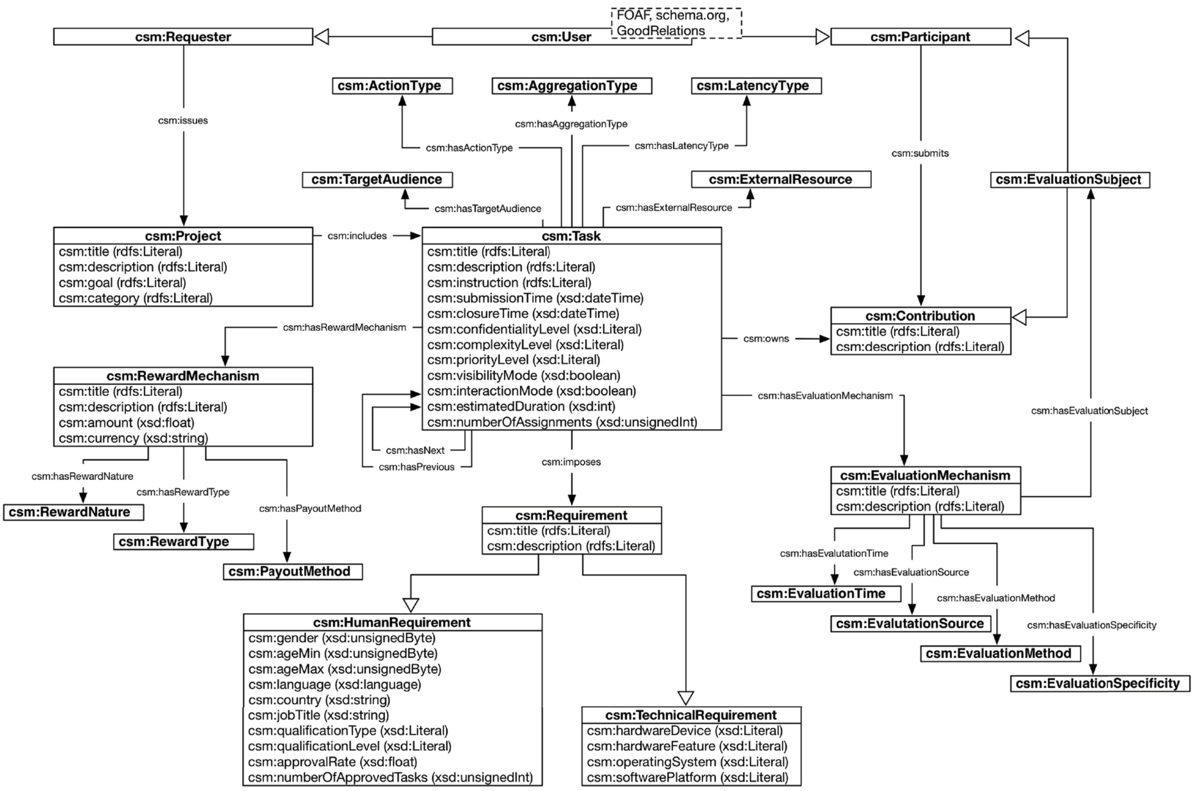

Figure 1: Classes, object and datatype properties of the crowdsourcing ontology

[1] A detailed specification and documentation of the CSM ontology can be found at: http://purl.org/csm/1.0.

4.1 USER

Klinger and Lease showed that the user identity, such as the real name or a reference to an existing public profile on a social networking site, improves the trustworthiness of the relationship between the worker and the requester (Klinger & Lease, 2011). Thus, the requester and the participant class are introduced into the ontology. Both concepts are subclasses of the user class (csm:User), which is in turn equivalent to the union of the person and organization concept of both the FOAF and the schema.org vocabulary. To specify the legal form, size, and role in the value chain, the user can also be related to the business entity concept of the GoodRelations ontology. The requester class (csm:Requester) identifies the initiator of a crowdsourcing project (csm:Project) and is either an individual, a company, a public organization, or a crowdsourcing intermediary. As opposed to the requester, the participant (csm:Participant) searches for an issued task or accepts a manually or automatically assigned crowdsourcing task (csm:Task). After accepting a task, the participant might also submit one or more contributions (csm:Contribution).

4.2 PROJECT

Each project (csm:Project) consists of one or more crowdsourcing tasks (csm:Task) and requires a well-defined and meaningful goal (csm:goal) to lead the crowd in the right direction (Chandler & Kapelner, 2013). Additionally, a category describing the general crowdsourcing activity, such as idea generation, problem-solving, or content creation, may be defined. All datatype properties and possible values of the project class are summarized in Table 2.

Table 2: Datatype properties of the project class

Element	Possible values
csm:goal	free text that contains the goal of the task
csm:category	idea generation, problem-solving, content creation, etc.

4.3 TASK

Eickhoff and De Vries demonstrated that crowd selection based on worker origin as well as implicit crowd filtering based on task design has a significant impact on the prevalence of malicious workers, and hence on the quality of the overall outcome (Eickhoff & de Vries, 2011). For that reason, the ontology contains classes and properties to describe the task and the required participants in detail. Since the appropriate specification of human and technical requirements is a core success factor of any crowdsourcing process, an independent class (csm:Requirement) is designed, which is described in the subsequent section.

A task (csm:Task) is the smallest indivisible unit of work that is clearly described by a single instruction (csm:instruction), such as rating a new product idea, labeling a picture, translating text, or finding an advertising slogan. Sometimes, an instruction is not sufficient enough to describe all aspects. External resources (csm:ExternalResource) refer to all additional inputs that are required to accomplish the crowdsourcing task, for example, web applications, documents, or datasets. A web application may be an online survey tool that is not part of the crowdsourcing system itself but has to be used to solve the problem. The requester sets the submission time (csm:submissionTime) when the task is accessible for the crowd and the closure time (csm:closureTime) when the task expires.

The identification of the target audience (csm:TargetAudience) allows the requester to set an initial general restriction of the size of the crowd. The crowdsourcing task can be addressed either to the employees of the enterprise, to the public domain, i.e., people who are not employees of the company, or to both the employees and the public community. Thus, the target audience can be set as an internal, external, or hybrid crowd (Vukovic & Bartolini, 2010). Setting the target audience of a crowdsourcing activity is important for the company if the confidentiality of the contributions becomes an issue. Additionally, the level of confidentiality (csm:confidentialityLevel) informs the participants whether technical documents or additional data can be passed on to third parties.

Doan et al. introduce several types of target problems. The participants in a crowdsourcing activity can evaluate, share, network, build artifacts, or execute tasks (Doan et al., 2011). Likewise, Corney et al. classify crowdsourcing work into three main types of tasks: creation tasks, evaluation tasks, and organizational tasks (Corney, Torres-Sanchez, Jagadeesan, & Regli, 2009). The specification of an action type (csm:ActionType) that has to be performed can support the participant in searching for a suitable task. Moreover, crowdsourcing tasks often differ in complexity and range from simple, moderate, to sophisticated tasks (Rouse, 2010). Other classification schemes group crowdsourcing tasks into simple, complex, and creative tasks (Schenk & Guittard, 2011). A division into three complexity degrees is adopted, namely simple, moderate, and complex (csm:complexityLevel).

Three semantic elements focus on the task significance, urgency, and effort. Depending on the importance of concurrent tasks, the requester may either increase or decrease the attention of the crowd. This can be done by indicating different priority levels (csm:priorityLevel). For some tasks, such as the collaborative creation of a knowledge repository, the focus lies on the accuracy of the contribution. In this case, the latency (csm:LatencyType) between issuing a task and getting an answer to the task does not matter. These tasks are defined as untimed. In other cases, such as an instant translation during a meeting, receiving an immediate reply is critical for the quality experience of the requester (Liu et al., 2010). This class particularly addresses the increasing prevalence of real-time crowdsourcing (Bernstein, Brandt, Miller, & Karger, 2011). Finally, the required effort can be determined by setting the estimated time required to complete the task (csm:estimatedDuration).

Influencing the degree of interaction between crowd members has a significant impact on the quality of the crowdsourcing outcome (Aparicio, Costa, & Braga, 2012). The ontology offers two mechanisms to control the interaction among crowd members. The first one influences the interaction indirectly by setting the visibility of contributions (csm:visibilityMode). The visibility defines whether the workers can see each other's contributions. This decision regarding visibility is critical for the outcome of the task, as it may either foster collaboration to incrementally approach a better solution or promote greater diversity of contributions if they are kept locked. The second mechanism enables or disables peer-to-peer collaboration by forbidding or permitting direct communication between the participants (csm:interactionMode).

Schenk and Guittard distinguish between two central aggregation mechanisms (csm:AggregationType) to combine the contributions of the crowd, namely integrative and selective crowdsourcing (Schenk & Guittard, 2011). In an integrative crowdsourcing process, the individual's contribution only becomes valuable when combined with other contributions. In contrast to integrative crowdsourcing, selective crowdsourcing perceives the individual's contribution as inherently valuable, as it addresses a specific problem directly. It is assumed that for both mechanisms the creation of the individual's contribution can proceed in parallel. However, if the outcome of a crowdsourcing process relies on improving a contribution iteratively by responding to contributions from other participants, such as creating a document and refining it step by step, a third aggregation mechanism called iterative crowdsourcing is introduced. The number of people who work in either of these three modes are determined by the datatype property csm:numberOfAssignments. An overview of all classes and properties related to the task class is given in Table 3.

Table 3: Classes and properties related to the task class

Element	Possible values
csm:instruction	free text that contains the instruction for the task
csm:submissionTime	time when the task is accessible for the crowd
csm:closureTime	time when the task expires
csm:TargetAudience	internal, external, hybrid
csm:confidentialityLevel	low (limited impact), moderate (serious impact), high (severe impact)
csm:ActionType	evaluate, share, network, build artifacts, or execute tasks
csm:complexityLevel	simple, moderate, complex
csm:priorityLevel	low, neutral, high
csm:LatencyType	immediate, untimed
csm:estimatedDuration	estimated time required to complete the task
csm:visibilityMode	hidden (false), visible (true)
csm:interactionMode	disabled (false), enabled (true)
csm:AggregationType	integrative, selective, iterative
csm:numberOfAssignments	number of participants who will be assigned to one task

4.4 REQUIREMENTS

The requirement class (csm:Requirement) consists of two subclasses (Table 4 and Table 5): the human requirement class (csm:HumanRequirement) and the technical requirement class (csm:TechnicalRequirement). As mentioned before, attracting, finding, and selecting the right users for a given crowdsourcing task is highly nuanced and context-sensitive, requiring the configuration of several demographic characteristics of the crowd, such as the gender, the age, the spoken language, or the country of origin (csm:gender, csm:ageMin, csm:ageMax, csm:language, csm:country). Other properties define the qualifications of the participant. These are the job title, the qualification type, the corresponding qualification level, the approval rate, and the number of approved tasks of a certain qualification type (csm:jobTitle, csm:qualificationType, csm:qualificationLevel, csm:approvalRate csm:number-OfApprovedTask). From a technical point of view, some forms of crowdsourcing, such as mobile crowdsourcing, require mechanisms to set hardware and software specifications beforehand (csm:hardwareDevice, csm:hardwareFeature, csm:operatingSystem, csm:soft-warePlatform).

Table 4: Datatype properties of the human requirement class

Element	Possible values
csm:gender	1-digit gender code according to ISO/IEC 5218
csm:ageMin	minimum age in years
csm:ageMax	maximum age in years
csm:language	2-digit language code according to ISO 639-1
csm:country	3-digit country code according to ISO 3166-1 alpha-3
csm:jobTitle	6-digit code of the standard occupational classification (SOC)
csm:qualificationType	any academic degree, certificate, or skill needed to solve the task
csm:qualificationLevel	the proficiency level of a qualification (poor, fair, good, very good, excellent)
csm:approvalRate	the ratio of properly solved tasks to the number of submitted tasks
csm:numberOfApprovedTasks	the number of properly solved tasks

Table 5: Datatype properties of the technical requirement class

Element	Possible values
csm:hardwareDevice	any hardware device (e.g., personal computer, mobile phone, smartphone, tablet computer)
csm:hardwareFeature	any build-in feature of the hardware device (e.g., processor, memory, GPS, camera, accelerometer, gyrometer)
csm:operatingSystem	any operating system for personal computers or mobile devices (e.g., Microsoft Windows, OS X, Linux, Android, Windows Phone, iOS)
csm:softwarePlatform	any operating system independent platform (e.g., Java, Firefox)

4.5 REWARD MECHANISM

One challenge in enterprise crowdsourcing is to offer an appropriate mix of incentives to motivate the crowd to participate in a crowdsourcing activity (Leimeister, Huber, Bretschneider, & Krcmar, 2009). As the requester has little or no influence on the intrinsic motivation of the participants, the ontology focuses mainly on incentives representing different types of direct compensation. Other types of motives and corresponding incentives are modeled only indirectly. For example, the incentive to foster self-marketing may be provided by the functionality to create user profiles that allow the participants to present their skills, experience, and knowledge to the crowd community.

To customize a reward, three classes can be related to the reward mechanism class (csm:RewardMechanism). First, the nature of the reward (csm:RewardNature) describes how a worker's contribution is rewarded. A reward may either be fixed, such as a certain amount of money disbursed after completing a task, proportional, such as a share of virtual points, or performance-based, such as a prize that depends on the ranking in a competition (Corney et al., 2009). If no reward is stated, the nature of the reward is marked as voluntary. Second, the type of the reward (csm:RewardType) allows the requester to specify what kind of reward is offered. On the one hand, a reward may be of immaterial value, such as providing virtual points that improve the worker's reputation, money in the form of a bonus that increases the salary, or access to a resource, which may or may not be related to the actual crowdsourcing initiative itself, for example, a knowledge repository. In addition, physical goods may be used to compensate workers for their efforts. Third, the payout method (csm:PayoutMethod) defines which participants in the crowdsourcing activity are rewarded, for example, all, the winner only, or the top ten participants. Finally, the amount (csm:amount), including an optional currency code (csm:currency), indicates the number of points or the monetary value of the reward. All classes and properties that are related to the reward mechanism class are recapitulated in Table 6.

Table 6: Classes and properties related to the reward mechanism class

Element	Possible values
csm:RewardNature	voluntary, fixed, proportional, performance-based
csm:RewardType	none, virtual points, money, discount, coupon, lottery, good, resource access
csm:PayoutMethod	all, winner, top 10
csm:amount	number of points or monetary value
csm:currency	3-digit currency code according to ISO 4217

4.6 EVALUATION MECHANISM

In the evaluation mechanism class (csm:EvaluationMechanism), two evaluation subjects (csm:EvaluationSubject) are distinguished: the user and the contribution. A user may be evaluated before starting the first task (ex-ante) or after finishing a task (ex-post). The former applies to entry questions, pre-qualification tasks, or gold standard data to determine the expertise or skill level of a worker (Corney et al., 2010). The latter considers acceptance and rejection decisions about historic contributions (Mashhadi & Capra, 2011). The users may also get feedback on the quality of the contribution during the work in progress. Thus, the evaluation mechanism class differentiates between three points of evaluation time, although they do not all logically apply to both evaluation subjects; for instance, a contribution cannot be evaluated before its creation.

The next class of the evaluation mechanism represents the choice of the evaluation source (csm:EvaluationSource). Zhao and Zhu note that the quality of contributions may either be checked manually by experts or peer workers, often by using a voting and rating mechanism, or automatically by the crowdsourcing system itself, using a specific data processing technique, such as a data mining or machine learning algorithm (Zhao & Zhu, 2012). Additionally, evaluation services offered by third party organizations may be employed. Dow et al. mention two further sources of evaluation (Dow, Kulkarni, Klemmer, & Hartmann, 2012). The contributors may assess their own work or the requesters may make the evaluation themselves.

Hirth et al. describe two major evaluation methods (csm:EvaluationMethod): the majority decision and the control group approach (Hirth, Hossfeld, & Tran-Gia, 2011). The majority decision approach assigns the task to multiple users who submit their individual results to the crowdsourcing system, and finally selects the most frequent result. In contrast, the control group approach assigns the task to one worker who completes the task. Afterwards, the crowdsourcing system sends the control group multiple validation tasks with the request to rate the submitted solutions. The solution will be accepted if the majority of the control group members decide it is correct.

The last class of the evaluation mechanism refers to the specificity of the evaluation (csm:EvaluationSpecificity), which may be a simple acceptance or rejection, a rating, for example, on a five star rating scale, an assessment form that codifies domain knowledge into pre-authored statements, or a custom free text response (Dow et al., 2012). Table 7 summarizes all classes related to the evaluation mechanism class.

Table 7: Classes related to the evaluation mechanism class

Element	Possible values
csm:EvaluationSubject	user, contribution
csm:EvaluationTime	before, simultaneously, after
csm:EvaluationSource	requester, self, expert, peer workers, third party, algorithm
csm:EvaluationMethod	majority decision, control group
csm:EvaluationSpecificity	accept or reject, rating, assessment form, free form

4.7 CONTRIBUTION

Contributions (csm:Contribution) comprise all data records, content items, documents, or code fragments that are part of the solution of the crowdsourcing task (csm:Task).

5 EVALUATION

The proposed ontology is an attempt to increase the semantics in enterprise crowdsourcing environments. Although a full evaluation based on the annotation of existing crowdsourcing data assets is in progress, the following brief assessment shows first evidence that the key representational requirements are well covered. The general applicability of the ontology is proven with respect to the requirements and central questions stated in section 3:

- *R1: Task specification.* The ontology defines a wide range of classes and properties to create a consistent and complete task specification, such as the goal, the instruction, or the reward and evaluation mechanism. A crowdsourcing system may guide the requester of a task through each of these elements by providing valuable hints on how to compose meaningful and consistent descriptions. A requester can also apply the ontology to search for participants who have already solved a crowdsourcing task with similar task characteristics.

- *R2: Task allocation.* The ontology includes several classes and properties to describe human and technical requirements of a crowdsourcing task in detail. On the one hand this allows the participants to easily find and select a task that fits best with their experience and knowledge. On the other hand a crowdsourcing system may recommend automatically a crowdsourcing task according to the self-defined qualifications or demographic characteristics of the user. Besides the human and technical requirements, a potential user might search for available crowdsourcing tasks according to preferred rewards, time constraints, or certain task types (evaluate, create, etc.).

- *R3: Team building.* The proposed ontology allows for the definition of multiple requirements to form homogenous or heterogeneous competence clusters. A crowdsourcing system may suggest additional team members that have skills, experiences, or knowledge that are still missing in order to accomplish the task. The integration of the FOAF and schema.org vocabulary gives access to the enterprise social network and hence to existing virtual communities of the user.

- *R4: Transaction transparency and quality control.* The ontology allows the tracking of the status of a crowdsourcing project, for example, we can apply the ontology to query how many and what type of participants contributed to a task. Additionally, we might learn about appropriate reward (type, nature, and amount) and evaluation mechanisms (time, method, source, specificity) for certain action types of a crowdsourcing task by statistically evaluating previously completed projects.
- *R5: Interoperability.* The definition of a common shared set of classes and properties that are equally understood among software architects and developers supports the integration of a crowdsourcing system into an existing ICT infrastructure and allows for standardized data exchange between diverse systems.

6 CONCLUSION

The principal outcome of this work is a lightweight ontology to support enterprise crowdsourcing activities. Although the ontology that is expressed as an OWL ontology offers a reasonable set of classes and properties to describe diverse crowdsourcing activities, several challenges remain to be addressed in future research and discussed among a wider audience. First, some of the datatype properties, such as the confidentiality level or the qualification type, are currently modeled as literal data types but might be candidates for individuals. As the objective is to develop a lightweight and practically useful ontology, balancing between simplicity and semantics of the crowdsourcing ontology remains a key challenge. Second, the proposed ontology currently adopts only concepts and properties from the FOAF, schema.org, and GoodRelations specifications. The semantic web community strongly recommends the reuse of existing standards and vocabularies. Therefore, future research should focus on the integration of additional specifications, such as Dublin Core, activitystrea.ms, SIOC, or PROV-O. This may, for example, prove useful in describing the contribution and external resource classes in more detail. Third, even though the ontology suggests a set of classes and properties to describe the reward and evaluation mechanism of a crowdsourcing activity, additional investigation and adjustment of the ontology is required in order to represent even more complex configurations. Therefore, we encourage researchers from the enterprise modeling, crowdsourcing, linked data, and semantic web community to critically review, comment, and where necessary to change or to extend the ontology. Further rigorous evaluation steps, such as conducting interviews with experts or annotating large crowdsourcing data assets, should be pursued in order to achieve successive adjustment and improvement as well as the widespread dissemination and acceptance of the ontology in research and practice.

REFERENCES

Aparicio, M., Costa, C. J., & Braga, A. S. (2012). Proposing a system to support crowdsourcing. In *Proceedings of the Workshop on Open Source and Design of Communication* (pp. 13–17). New York, NY, USA: ACM. doi:10.1145/2316936.2316940

Bernstein, M. S., Brandt, J., Miller, R. C., & Karger, D. R. (2011). Crowds in two seconds: enabling realtime crowd-powered interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (pp. 33–42). New York, NY, USA: ACM. doi:10.1145/2047196.2047201

Chandler, D., & Kapelner, A. (2013). Breaking monotony with meaning: Motivation in crowdsourcing markets. *Journal of Economic Behavior & Organization*, *90*, 123–133. doi:10.1016/j.jebo.2013.03.003

Corney, J. R., Torres-Sanchez, C., Jagadeesan, A. P., & Regli, W. C. (2009). Outsourcing labour to the cloud. *International Journal of Innovation and Sustainable Development*, *4*(4), 294–313.

Corney, J. R., Torres-Sánchez, C., Jagadeesan, A. P., Yan, X. T., Regli, W. C., & Medellin, H. (2010). Putting the crowd to work in a knowledge-based factory. *Advanced Engineering Informatics*, *24*(3), 243–250. doi:10.1016/j.aei.2010.05.011

Cosley, D., Frankowski, D., Terveen, L., & Riedl, J. (2007). SuggestBot: using intelligent task routing to help people find work in Wikipedia. In *Proceedings of the 12th International Conference on Intelligent User Interfaces* (pp. 32–41). New York, NY, USA: ACM. doi:10.1145/1216295.1216309

Das, R., & Vukovic, M. (2011). Emerging theories and models of human computation systems: a brief survey. In *Proceedings of the 2nd international workshop on Ubiquitous crowdsouring* (pp. 1–4). New York, NY, USA: ACM. doi:10.1145/2030100.2030102

Doan, A., Ramakrishnan, R., & Halevy, A. Y. (2011). Crowdsourcing systems on the World-Wide Web. *Communications of the ACM*, *54*(4), 86–96. doi:10.1145/1924421.1924442

Dow, S., Kulkarni, A., Klemmer, S., & Hartmann, B. (2012). Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (pp. 1013–1022). New York, NY, USA: ACM. doi:10.1145/2145204.2145355

Eickhoff, C., & de Vries, A. P. (2011). How crowdsourcable is your task? In *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the Fourth ACM International Conference on Web Search and Data Mining (WSDM)* (pp. 11–14). Hong Kong. Retrieved from http://siplab.tudelft.nl/sites/default/files/csdm2011.pdf

Grüninger, M., & Fox, M. S. (1995). Methodology for the design and evaluation of ontologies. In *Proceedings of the International Joint Conference on AI (IJCAI-95), Workshop on Basic Ontological Issues in Knowledge Sharing* (pp. 1–10). Montreal.

Hetmank, L. (2013a). Components and functions of crowdsourcing systems – a systematic literature review. In *11th International Conference on Wirtschaftsinformatik* (pp. 55–69). Leipzig.

Hetmank, L. (2013b). Towards a semantic standard for enterprise crowdsourcing – a scenario-based evaluation of a conceptual prototype. In *21st European Conference on Information Systems*. Utrecht.

Hirth, M., Hossfeld, T., & Tran-Gia, P. (2011). Cost-optimal validation mechanisms and cheat-detection for crowdsourcing platforms. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on* (pp. 316–321). Seoul. doi:10.1109/IMIS.2011.91

Jayakanthan, R., & Sundararajan, D. (2012). Enterprise crowdsourcing solution for software development in an outsourcing organization. In A. Harth & N. Koch (Eds.), *Current Trends in Web Engineering* (Vol. 7059, pp. 177–180). Springer Berlin / Heidelberg.

Kittur, A., Khamkar, S., André, P., & Kraut, R. (2012). CrowdWeaver: visually managing complex crowd work. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (pp. 1033–1036). New York, NY, USA: ACM. doi:10.1145/2145204.2145357

Klinger, J., & Lease, M. (2011). Enabling trust in crowd labor relations through identity sharing. *Proceedings of the American Society for Information Science and Technology*, *48*(1), 1–4. doi:10.1002/meet.2011.14504801257

Leimeister, J. M., Huber, M., Bretschneider, U., & Krcmar, H. (2009). Leveraging crowdsourcing: activation-supporting components for IT-based ideas competition. *Journal of Management Information Systems*, *26*(1), 197–224. doi:10.2753/MIS0742-1222260108

Liu, Y., Lehdonvirta, V., Kleppe, M., Alexandrova, T., Kimura, H., & Nakajima, T. (2010). A crowdsourcing based mobile image translation and knowledge sharing service. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia* (pp. 1–9). New York, NY, USA: ACM. doi:10.1145/1899475.1899481

Mashhadi, A. J., & Capra, L. (2011). Quality control for real-time ubiquitous crowdsourcing. In *Proceedings of the 2nd international workshop on Ubiquitous crowdsouring* (pp. 5–8). New York, NY, USA: ACM. doi:10.1145/2030100.2030103

Nielsen, M. (2011). *Reinventing discovery: the new era of networked science*. Princeton University Press.

Pedersen, J., Kocsis, D., Tripathi, A., Tarrell, A., Weerakoon, A., Tahmasbi, N. , Xiong, J., Deng, W., Oh, O. & de Vreede, G.-J. (2013). Conceptual foundations of crowdsourcing: a review of IS research. In *HICCS-46 Hawaii International Conference on System Science*. Grand Wailea, Maui, Hawaii.

Rouse, A. C. (2010). A preliminary taxonomy of crowdsourcing. In *21st Australasian Conference on Information Systems*. Brisbane.

Schall, D. (2012). *Service-oriented crowdsourcing: architecture, protocols and algorithms*. New York: Springer.

Schenk, E., & Guittard, C. (2011). Towards a characterization of crowdsourcing practices. *Journal of Innovation Economics*, *7*(1), 93–107. doi:10.3917/jie.007.0093

Uschold, M., & King, M. (1995). Towards a methodology for building ontologies. In *Proceedings of the International Joint Conference on AI (IJCAI-95), Workshop on Basic Ontological Issues in Knowledge Sharing*. Budapest.

Villarroel, J. A., & Reis, F. (2010). Intra-corporate crowdsourcing (ICC): leveraging upon rank and site marginality for innovation. In *CrowdConf 2010*. San Francisco.

Vukovic, M. (2009). Crowdsourcing for enterprises. In *Proceedings of the 2009 Congress on Services - I* (pp. 686–692). Los Angeles: IEEE. doi:10.1109/SERVICES-I.2009.56

Vukovic, M., & Bartolini, C. (2010). Towards a research agenda for enterprise crowdsourcing. In T. Margaria & B. Steffen (Eds.), *ISoLA'10 Proceedings of the 4th international conference on Leveraging applications of formal methods, verification, and validation* (Vol. 6415, pp. 425–434). Heraclion: Springer.

Vukovic, M., Laredo, J., & Rajagopal, S. (2010). Challenges and experiences in deploying enterprise crowdsourcing service. In B. Benatallah, F. Casati, G. Kappel, & G. Rossi (Eds.), *Web Engineering* (Vol. 6189, pp. 460–467). Berlin, Heidelberg: Springer.

Vukovic, M., Lopez, M., & Laredo, J. (2010). PeopleCloud for the globally integrated enterprise. In A. Dan, F. Gittler, & F. Toumani (Eds.), *Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops* (Vol. 6275, pp. 109–114). Springer.

Zhao, Y., & Zhu, Q. (2012). Evaluation on crowdsourcing research: current status and future direction. *Information Systems Frontiers*, 1–18.

An Ontology for Enhancing Automation and Interoperability in Enterprise Crowdsourcing Environments (Technical Report)

Lars Hetmank

Abstract. Enterprise crowdsourcing transforms the way in which traditional business tasks can be processed by harnessing the collective intelligence and workforce of a large and often diversified group of people. At the present time, data and information residing within enterprise crowdsourcing systems and other business applications are insufficiently interlinked and are rarely made publicly available in an open and semantically structured manner – neither to the corporate intranet nor to the World Wide Web (WWW). However, the semantic annotation of enterprise crowdsourcing activities is a promising research and application domain. The Semantic Web and its related technologies, methods and principles for publishing structured data offer an extension of the traditional layout-oriented Web to provide more intelligent and complex services.

This technical report describes the efforts toward a universal and lightweight yet powerful Semantic Web vocabulary for the domain of enterprise crowdsourcing. As a methodology for developing the vocabulary, the approach of ontology engineering is applied. To illustrate the purpose and to limit the scope of the ontology, several informal competency questions as well as functional and non-functional requirements are presented. The subsequent conceptualization of the ontology applies different sources of knowledge and considers various perspectives. A set of semantic entities is derived from a review of existing crowdsourcing applications and a review of recent crowdsourcing literature. During the domain capture, all partial results of the review are integrated into a consistent data dictionary and structured as a UML data schema. The designed ontology includes 24 classes, 22 object properties and 30 datatype properties to describe the key aspects of a crowdsourcing model (CSM). To

demonstrate the technical feasibility, the ontology is implemented using the Web Ontology Language (OWL). Finally, the ontology is evaluated by means of transforming informal to formal competency questions, comparing it to existing semantic vocabularies, and calculating ontology metrics. Evidence is shown that the CSM ontology covers the key representational needs of the enterprise crowdsourcing domain. At the end of the technical report, current limitations are illustrated and directions for future research are proposed.

Note. This research article was published as Hetmank, L. (2014). Technical Report: An Ontology for Enhancing Automation and Interoperability in Enterprise Crowdsourcing Environments.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF CODE LISTINGS

LIST OF ABBREVIATIONS

API	Application Programming Interface
AR	Attribute Richness
CAM	Contextualized Attention Metadata
CSM	Crowdsourcing Model
DERI	Digital Enterprise Research Institute
DC	Dublin Core
DOAP	Description of a Project
DTD	Document Type Definition
FOAF	Friend of a Friend
GR	Good Relations
HTML	HyperText Markup Language
ICT	Information and Communication Technology
IR	Inheritance Richness
IRI	Internationalized Resource Identifier
JSON	JavaScript Object Notation
JSON-LD	JSON-based Serialization for Linked Data
OWL	Web Ontology Language
PROV	Provenance Ontology
PURL	Persistent URL
RDF	Resource Description Framework
RDFa	RDF in Attributes
RDFS	RDF Schema
RDF/XML	XML Syntax for RDF
RG	Research Goal
RR	Relationship Richness
SIOC	Semantically-Interlinked Online Communities
SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language
Turtle	Terse RDF Triple Language
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

W3C	WWW Consortium
WS-BPEL	Web Services Business Process Execution Language
WWW	World Wide Web
XFN	XHTML Friends Network
XHTML	Extensible HTML
XML	Extensible Markup Language
XPDL	XML Process Definition Language
XSD	XML Schema Definition

1 INTRODUCTION

Leveraging the knowledge and workforce of a large, undefined group of people to solve organizational tasks via Web-based technologies is the central idea of enterprise crowdsourcing[1] (Brabham, 2013; Estellés-Arolas & González-Ladrón-de-Guevara, 2012; Howe, 2008). In a traditional corporate context, these organizational tasks are performed by dedicated experts, project teams, or departments, and comprise a wide range of different types of tasks with various complexity levels, such as collecting and categorizing data, generating ideas, writing content, translating documents, or solving complex problems (Hetmank, 2014a). Well-known examples of technical platforms supporting enterprise crowdsourcing include Amazon Mechanical Turk[2] and InnoCentive.[3] Besides these externally hosted crowdsourcing intermediaries, several corporate-specific, often task-customized solutions exist, for instance, IBM's PeopleCloud, a crowdsourcing application to manage scalable virtual teams of knowledge workers (Lopez, Vukovic, & Laredo, 2010), or CrowdREquire, a crowdsourcing platform for requirements engineering (Adepetu, Ahmed, & Abd, 2012).

Deploying enterprise crowdsourcing provides numerous advantages for a company (Table 1). While primarily benefiting from the reduction of personnel and equipment costs, companies can also externalize their risk of failure in executing a task, for example, they may transfer the uncertainties in finding a solution or in running an experiment to a large number of workers (Ye & Kankanhalli, 2013). Additional benefits often result from accessing the valuable, distributed, heterogeneous knowledge and skills of the crowd (Saxton, Oh, & Kishore, 2013). For example, companies can find workers to accomplish tasks which they are unable to solve themselves (Malone, Laubacher, & Dellarocas, 2010). Further, the heterogeneity of the crowd may lead to an improved quality or at least to an increased innovativeness of the solutions. Finally, a company may gain advantages according to time aspects. With enterprise crowdsourcing, companies are able to support a flexible workforce (Khazankin, Satzger, & Dustdar, 2012). It allows to efficiently utilize free, especially short-term working capacities, but also to mitigate shortages of experts in certain subject areas or company locations (Kittur

---

[1] The term enterprise crowdsourcing is currently used inconsistently in academic literature. The narrow understanding of the term limits the target audience to the employees of large, mostly multinational firms, such as IBM or SAP. In contrast, the broad understanding does not restrict the target audience. In this technical report, the author takes the view of the broad definition, however, stresses the importance of outsourcing organizational tasks to the crowd. For a more detailed discussion about the term enterprise crowdsourcing please refer to (Hetmank, 2014a).
[2] Amazon Mechanical Turk can be found at https://www.mturk.com/mturk/.
[3] InnoCentive can be found at https://www.innocentive.com/.

et al., 2013). Additionally, due to the large number and the various special skills of the crowd, organizational tasks are often solved in shorter time.

Although the benefits of enterprise crowdsourcing seem to be promising, several challenges that companies may encounter still remain (Table 1). First, many crowdsourcing tasks do not get the required attention from the crowd, either because of the sheer abundance of the tasks at crowdsourcing platforms or simply due to inappropriate task specifications, incentive schemes, or evaluation mechanisms (Ye & Kankanhalli, 2013). For potential participants it is often difficult to identify tasks that match their interests, skills, experiences and expert knowledge (Schall, 2012, p. 2). Another challenge, especially in enterprise crowdsourcing that focuses more on complex, knowledge-intensive tasks rather than on simple tasks, is how to connect the large crowd of people to create an outcome that is more cost-effective, qualitatively better, and less time consuming than the efforts of any individual or team of traditional workers (Lykourentzou, Vergados, Papadaki, & Naudet, 2013; Skopik, Schall, & Dustdar, 2012). In general, when replacing the traditional and pre-assigned workers of an organizational task with members of the crowd, organizations must ensure, on the one hand, that enough and appropriate workers engage in the crowdsourcing task, and, on the other hand, that the submitted contributions achieve a sufficient quality.

Table 1: Benefits and challenges of enterprise crowdsourcing

Benefits	Challenges
• reducing personnel and equipment costs • externalizing the risk of failure in executing a task • accessing the valuable, distributed, heterogeneous knowledge and skills of the crowd • improving the quality and increasing the innovativeness of the solutions • supporting a flexible workforce • utilizing free, especially short-term working capacities, but also mitigating shortages of experts in certain subject areas or company locations • solving tasks in shorter time due to the large number and the various special skills of the crowd	• getting the required attention from the crowd to solve a crowdsourcing task • engaging the crowd to create an outcome that is more cost-effective, qualitatively better, and less time consuming than the efforts of any individual or team of traditional workers

As a possible solution to overcome the above-mentioned challenges, this technical report investigates the potentials of applying Semantic Web technologies in the domain of enterprise crowdsourcing. Present Semantic Web vocabularies are designed to describe, for example, aspects of electronic commerce (GoodRelations), social network services (FOAF), or online communities (SIOC).[4] Unfortunately, these vocabularies fall short of representing certain aspects of enterprise crowdsourcing, such as reward schemes, evaluation mechanisms, or human and technical requirements. Thus, to fill this gap, this report proposes the crowdsourcing model (CSM) ontology that is tailored to the specific needs of enterprise crowdsourcing.

The remainder of the technical report is structured as follows: The next chapter 2 explains the overall research objective. Chapter 3 describes ontology engineering as the underlying methodology that is applied to develop an ontology for enterprise crowdsourcing. After briefly introducing the methodology, each activity of the methodology is explained in a separate chapter (from Chapter 4 to 6). In chapter 4, the purpose and scope including several informal competency questions, and functional and non-functional requirements are presented. After that and according to the research objective, the ontology development process is illustrated at a conceptual, logical, and physical level of abstraction (Chapter 5). Finally, in chapter 6, the ontology is evaluated to demonstrate the general applicability of the proposed semantic entities of the ontology. The report concludes with a critical reflection of the results and describes several aspects of the ontology that require further discussion (Chapter 7).

[4] The benefits and limitations of applying these vocabularies in the crowdsourcing domain are discussed in section 5.3.1.

2 RESEARCH OBJECTIVE

In this chapter, the scope of the technical report is determined by defining an overall research objective. The objective of this work is to develop a lightweight and extensible ontology for capturing, storing, utilizing, and sharing crowdsourcing data that is grounded on Semantic Web technologies and Linked Data principles. In the context of the Semantic Web, this kind of easy-to-use ontologies are also referred to as *semantic vocabularies* or *Semantic Web vocabularies* (Grimm, Abecker, Völker, & Studer, 2011). Generally speaking, the ontology aims to improve the automation and interoperability in enterprise crowdsourcing environments. To gradually achieve this objective, the following three research goals (RG) are pursued:

RG 1: The first goal is to establish a conceptual foundation that comprehensively describes essential aspects of enterprise crowdsourcing from various perspectives. The foundation should be based both on a review of existing crowdsourcing applications and on a review of previously published crowdsourcing literature.

RG 2: The second goal is to derive a universal and technology-independent data dictionary *and schema* for the domain of enterprise crowdsourcing from the results of the review. The data dictionary and schema should provide the basis for designing an ontology that facilitates the structured recording of the key crowdsourcing concepts, relationships, and attributes in an organized and meaningful way.

RG 3: The last goal is to evaluate the designed ontology regarding feasibility and utility. Therefore, the ontology should be prototypically implemented by using a schema definition language. Additionally, use case scenarios should be created to exemplarily show the applicability of the ontology.

As depicted in the description of the research goals, the development process of the ontology should pass through all levels of abstractions: starting from the contextual and conceptual layer (conceptual foundation), over the logical layer (data dictionary and schema), to the physical layer (implementation and instantiation). In the short term, the choice of a certain schema definition language, such as the Document Type Definition (DTD), the Extensible Markup Language (XML) schema definition (XSD), the Resource Description Framework Schema (RDFS), or the Web Ontology Language (OWL) usually depends on the company's existing IT infrastructure. The same is true for languages to describe the instances of an ontology, such as the Terse RDF Triple Language (Turtle), the JavaScript Object Notation for Linked Data (JSON-LD), the RDF in Attributes (RDFa), the RDF syntax for RDF (RDF/XML), the microformats language, or the microdata language. In the long run, however, the sustainability of these languages is unpredictable. Thus, these three layers provide academics and practitioners with the necessary flexibility for adaptation and extension on the required level of abstraction.

3 ONTOLOGY ENGINEERING

This chapter describes the overall methodology that is applied for building the ontology for the domain of enterprise crowdsourcing. In the domain of computer and information science, Gruber (2009) defines an *ontology* as "a set of representational primitives with which to model a domain of knowledge or discourse" (pp. 1963–1965). The representational primitives of an ontology are divided into *classes* (concepts), *relationships* (object properties), and *attributes* (datatype properties). In this work, the term *semantic entity* is used as an auxiliary term to generally refer to any of the three types of representational primitives.

Similar to a model, an ontology does not aim to represent the entire world of interest, instead it covers only selected aspects of the reality which are relevant to address the specific purpose of an ontology. Thus, finding and selecting these essential aspects is of key importance for the ontology development process, and hence for this technical report. The methodology that is applied in this research project is *ontology engineering.*[5] Ontology engineering provides a systematic and objective procedure for developing ontologies (Sure, Staab, & Studer, 2009). This procedure includes a set of activities to support the conceptualization, design, implementation, and deployment of ontologies (Devedzić, 2002).

In the last decades, several different methodologies have been suggested to support ontology engineering. A comprehensive comparison between these methodologies is drawn in Jones, Bench-Capon, and Visser (1998), Gómez-Pérez, Fernández-López, and Corcho (2005), or Casellas (2011). Each of these methodologies has its benefits and limitations. As a consequence, methods and procedures of various ontology engineering methodologies are employed in this report.

One method that is adopted for initiating the ontology engineering process forms part of the methodology by Grüninger and Fox (1995). They suggest that an essential step toward an ontology is to describe *motivating scenarios* in the form of story problems or examples that are not sufficiently addressed by existing ontologies. The description of scenarios is a typical method to understand the scope and the motivation behind the proposed ontology regarding its applications. To start with the ontology development and to elicit first semantic entities, three distinct motivating scenarios representing typical enterprise crowdsourcing activities have already been presented in a previous article of the author (Hetmank, 2013b).

[5] The methodology of *ontology engineering* is sometimes also referred as *ontological engineering.*

Another method that has proved beneficial for the ontology engineering is the identification of a set of *informal competency questions*. Grüninger and Fox (1995) but also other ontology developers, such as Hepp (2008), formulate competency questions in natural languages that should be answered by the ontology once it is expressed in a formal language. In this paper, an initial set of competency questions is elaborated based on the motivating scenarios (Section 4.1). Additionally, several *functional and non-functional requirements* are introduced to further guide the ontology development (Section 4.2 and Figure 1).

The overall procedure for building the crowdsourcing ontology, however, is derived from a suggestion by Uschold and King (1995). According to them four activities must be performed: (A) identify the purpose and scope of the ontology, (B) develop the ontology, (C) evaluate the ontology, and (D) document the ontology (Figure 1). The activity of ontology development is further grouped into the following steps: conceptualization,[6] domain capture, integration, and implementation. The embodiment of each of the four steps is described in detail in chapter 5.

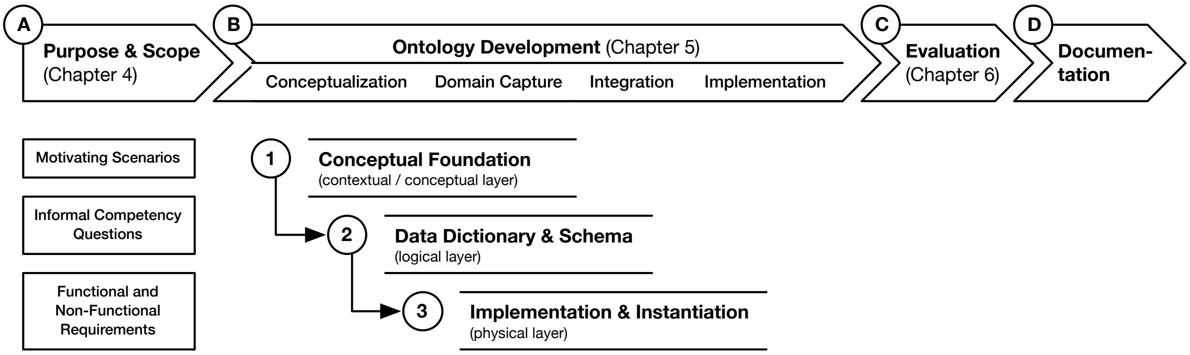

Figure 1: Ontology engineering process of the enterprise crowdsourcing ontology

Although the methodology by Uschold and King (1995) has its advantages, there is one main weakness. It is the missing conceptualization, which starts with the development of a less formal domain model prior to the implementation of the ontology. Thus, another methodology that is considered for the ontology engineering is a methodology called *Methontology* that was developed by the ontology group at the Universidad Politécnica de Madrid (Gómez-Pérez et al., 2005, pp. 125–142). The main contribution of this methodology is that, on the one hand, it offers guidance for the conceptualization of the ontology construction and, on the other hand, that it follows the idea of an ontology building life cycle based on evolving prototypes. To manage the complex undertaking of ontology development for enterprise crowdsourcing, both suggestions are taken into account. Besides the idea of an ontology life

[6] Note that Uschold and King (1995) did not integrate the *conceptualization* step in the methodology.

cycle and the realization of a detailed conceptualization, *Methontology* covers additional aspects that are worth adopting. One suggestion is the early consideration and integration of existing semantic vocabularies and standards. Thus, the technical report focuses on this issue before implementing the ontology.[7]

In this report, a three-layer design approach is proposed that moves gradually from the knowledge to the implementation level (Figure 1). The development of the ontology initiates with the construction of a conceptual foundation. Based on this, a data dictionary as well as a semi-formal data schema is derived. The data dictionary contains a set of entities with a precise semantic definition. Within the data schema the entities are connected by specific relationships. It provides the basement for the implementation of the ontology using a schema definition language (Haslhofer & Klas, 2010; Rahm & Bernstein, 2001). A schema can be declared in various schema definition languages, such as the Extensible Markup (XML) schema, the RDF Vocabulary Description Language (RDFS), or the Web Ontology Language (OWL). In this technical report, the schema is exemplarily implemented in OWL by using the open-source ontology development editor *Protégé*.[8]

This chapter presented the ontology engineering methodology used for building the ontology for the enterprise crowdsourcing domain. In the next chapter, the purpose and scope of the ontology is defined, which is the first activity of the ontology engineering process.

[7] Note that Uschold and King (1995) proposed this the other way around.
[8] The ontology development editor *Protégé* can be found at http://protege.stanford.edu/.

4 PURPOSE AND SCOPE

The first activity of ontology engineering is the definition of the primary purpose, granularity, range of the intended users, and scope of the ontology. As stated above, the main *purpose* of the ontology is to provide a controlled vocabulary for capturing, storing, utilizing, and sharing crowdsourcing data. Moreover, the ontology aims to enable intelligent software agents acting on our behalf to reason about these data and it is mainly intended to be used in the Semantic Web and Linked Data context.

Due to the fact that large and expressive ontologies require an increased demand of resources for reviewing and understanding the specification, and hence often lead to a lower adoption, the ontology should have an appropriate level of *granularity*. In this regard, Hepp (2007) notes that "in practice, useful ontologies must be small enough to have reasonable familiarization and commitment costs and big enough to provide substantial added value for using them" (p. 94). Thus, the ontology that is developed for the domain of enterprise crowdsourcing should be lightweight and practically beneficial.

According to the range of *intended users* of the ontology, two groups of users can be distinguished: those who directly and those who indirectly benefit from the ontology. The first group consists of developers, implementers, and maintainers of the ontology who directly reuse or, where necessary, adopt the ontology for their own purposes, especially in the domain of enterprise crowdsourcing or other related areas. The users of an enterprise crowdsourcing application are usually subsumed under the second group. They indirectly benefit from the augmented and enhanced capabilities that the ontology facilitates. For example, the requester of a crowdsourcing task gains advantage from recommending features, whereas the crowd itself profits from improved searching and browsing capabilities.

As stated earlier, two methods are useful to define the *scope* of the ontology: describing motivating scenarios and identifying informal competency questions. Whereas three motivating scenarios have been presented in a previous paper, the view is now completed in this report by specifying the expressive and reasoning requirements of the ontology in form of several informal competency questions (Section 4.1). Additionally, several functional and non-functional requirements that guide the development process are introduced (Section 4.2).

4.1 INFORMAL COMPETENCY QUESTIONS

The main purpose of *informal competency questions* is to direct the development and to provide a test suite for the evaluation of the ontology (Obrst, Ceusters, Mani, Ray, & Smith, 2007). Thus, before starting to develop the ontology, the purpose and scope of the ontology is sharpened by raising several informal competency questions (Uschold & Grüninger, 1996, p. 29). These questions are derived from the three motivating scenarios that were presented in an earlier paper (Hetmank, 2013b). The three scenarios encompassed:

Scenario 1: the evaluation of product design proposals,

Scenario 2: the translation of a technical specification, and

Scenario 3: the building of a company-wide virtual library.

The derived questions comprise not only top-down questions that consider the nature of the domain (e.g., CQ-R01, CQ-P01) but also bottom-up questions that encompass queries to the instances of the ontology (e.g., CQ-R02, CQ-P02). Although the questions below are non-exhaustive and vary in their level of abstraction, they exemplarily depict possible queries to the ontology, and hence the expressive power of the ontology. The informal competency questions are differentiated between questions from the requester's perspective and questions from the participant's perspective.

From the requester's perspective, the ontology may consider the following informal competency questions:

CQ-R01: Which task characteristics should be considered when defining a crowdsourcing task?

CQ-R02: Given a set of task requirements, which participants are suitable to solve a crowdsourcing task?

CQ-R03: Who has already done a crowdsourcing task with similar or identical task characteristics compared to the one to be issued?

CQ-R04: How many crowdsourcing tasks has a certain participant already solved?

CQ-R05: Which crowdsourcing tasks are issued internally to the employees or externally to the general public?

CQ-R06: Based on the analysis of historical crowdsourcing data, which type and amount of reward is appropriate for a crowdsourcing task with similar or identical task characteristics?

From the perspective of a potential candidate who wants to engage in a crowdsourcing task, the ontology may consider the following informal competency questions:

CQ-P01: Which task characteristics should be considered when choosing a crowdsourcing task?

CQ-P02: When will the crowdsourcing task be submitted or closed?

CQ-P03: Given a set of human requirements (e.g., qualifications, interests, demographic preferences), which crowdsourcing tasks fit best for a participant?

CQ-P04: Given the preference for a certain type and amount of a reward (e.g., money, reputation points) as well as some industry sector constraints (e.g., a participant dislikes to work for a military company), which crowdsourcing tasks are available?

CQ-P05: Given the preference for a type of task (evaluate, create, etc.) as well as some time constraints (e.g., the participant wants to work on some low cognitive tasks from 1:00 p.m. to 3:00 p.m.), which crowdsourcing tasks are available?

CQ-P06: Which crowdsourcing tasks are available for a certain technical device (e.g., mobile phone, smartphone, tablet computer, personal computer) or technical feature (e.g., GPS, camera, display resolution and size, accelerometer, gyrometer)?

While these competency questions give a first impression of the expressiveness of the ontology, they are more of a descriptive rather than a prescriptive nature. Thus, additionally a set of requirements is provided that the ontology should meet.

4.2 REQUIREMENTS

Similar to software system requirements, *requirements* for ontologies can be distinguished between functional and non-functional requirements. Whereas functional requirements state what services or functions the ontology should provide, non-functional requirements constrain these characteristics by overall quality criteria (Sommerville, 2011, p. 84). This work focuses first on the functional requirements (Section 4.2.1) and then draws the attention to the non-functional requirements of the enterprise crowdsourcing ontology (Section 4.2.2).

4.2.1 Functional Requirements

The functional requirements for the design of the enterprise crowdsourcing ontology are underpinned by the crowdsourcing literature. Table 2 summarizes the top-level functional requirements. Each of the top-level requirements includes a set of additional aspects that are discussed in the next four subsections.

Table 2: Functional requirements of the enterprise crowdsourcing ontology

Code	Top-level functional requirement	Description
F-01	Structured task specification	A crowdsourcing ontology should facilitate an appropriate architectural support to define tasks. (Source: Chandler & Kapelner, 2013; Kittur, Chi, & Suh, 2008; Liu, Bias, Lease, & Kuipers, 2012; Lopez et al., 2010)
F-02	Efficient task allocation	A crowdsourcing ontology should support both the user in selecting a task (pull method) and the system in recommending a task (push method) taking into account (i) the suitability of the task for a worker, (ii) the worker's availability, and (iii) the worker's motivational aspects. (Source: Corney et al., 2010; Cosley, Frankowski, Terveen, & Riedl, 2007; Satzger, Psaier, Schall, & Dustdar, 2013; Schall, 2012)
F-03	Dynamic team building	A crowdsourcing ontology should provide the foundation to identify existing working groups as well as to form new, globally distributed teams depending on the task requirements and based on the workers' existing social networks. (Source Kearns, 2012; Law & von Ahn, 2011; Vukovic, 2009)
F-04	Transaction transparency and quality control	A crowdsourcing ontology should include semantic entities that the requester of a task can consult and statistically evaluate in order to optimize the crowdsourcing activity. (Source: Dai, Mausam, & Weld, 2010; Kittur, Khamkar, André, & Kraut, 2012; Kulkarni, Can, & Hartmann, 2012; Liu et al., 2012)

4.2.1.1 Structured Task Specification (F-01)

The quality of the contributions of the crowd is highly dependent on the quality and detail of the task design. In order to receive useful contributions from the crowd and to reduce unnecessary spam, crowdsourcing tasks must be carefully designed (Liu et al., 2012). Kittur, Chi, and Suh (2008) suggest that requesters should not only issue verifiable crowdsourcing tasks but also explicitly indicate that the contributions of the crowd will be examined. They also note that special care must be taken in the design of subjective or qualitative tasks. Besides issuing verifiable tasks, Chandler and Kapelner (2013) found out that issuing meaningful tasks plays an important role to produce more and better results. Moreover, it has the positive side effect that the members of the crowd require less compensation for their efforts. In a nutshell, to improve the overall quality, Lopez, Vukovic, and Laredo argue for providing structured task specifications and note that crowdsourcing tasks must be integrated into the corresponding business processes (Lopez et al., 2010). Thus, the first top-level requirement is:

F-01: A crowdsourcing ontology should facilitate an appropriate architectural support to define tasks.

This requirement includes five aspects. The crowdsourcing ontology should provide semantic entities for:

- creating detailed task descriptions (F-01.a),
- declaring effective incentive schemes (F-01.b),
- formalizing evaluation mechanisms (F-01.c),
- recording human and technical requirements (F-01.d and F-01.e), and
- managing contributions and controlling their outcome (F-01.f).

4.2.1.2 Efficient Task Allocation (F-02)

Proposing crowdsourcing tasks to suitable and trustworthy workers at the right time, and in the right way, is key for increasing the efficiency and quality of the contributions, and hence the success of a crowdsourcing initiative (Nielsen, 2011). An efficient task allocation is particularly important for crowdsourcing tasks that demand a special talent (Corney et al., 2010). There exist manifold algorithms to distribute tasks based on communities, context or skills (Satzger et al., 2013). To benefit from these algorithms, crowdsourcing systems require an intelligent task routing mechanism that is based on an elaborate specification of task requirements and detailed user profiles (Cosley et al., 2007). Moreover, current crowdsourcing platforms offer limited search and navigation support in helping the crowd to identify relevant tasks corresponding to their interests, skills, and knowledge (Schall, 2012, p. 2, p. 14). Consequently, the second top-level requirement is:

F-02: A crowdsourcing ontology should support both the user in selecting a task (pull method) and the system in recommending a task (push method) taking into account (i) the suitability of the task for a worker, (ii) the worker's availability, and (iii) the worker's motivational aspects.

This requirement is further divided into three aspects. The crowdsourcing ontology should provide semantic entities for:

- publishing user descriptions (F-02.a),
- proclaiming suitability including interests, skills, experience, and expert knowledge (F-02.b), and
- indicating availability comprising both time and location properties, such as the vacation time, the working schedule, the free working capacities, and the place of work (F-02.c).

4.2.1.3 Dynamic Team Building (F-03)

The formation of goal-directed relationships between the participants of a crowdsourcing system with either similar or diverse cross-functional skills, knowledge, or experiences is often a prerequisite to solve large and complex tasks (Law & von Ahn, 2011, p. 61). Unfortunately, most of the existing crowdsourcing systems do not exhibit the interdependence of user actions that challenging collective tasks require and fall short of facilitating a flexible, dynamic, and proactive assembly of globally distributed teams (Kearns, 2012; Vukovic, 2009). As a consequence, the third top-level requirement is:

F-03: A crowdsourcing ontology should provide the foundation to identify existing working groups as well as to form new, globally distributed teams depending on the task requirements and based on the workers' existing social networks.

Two aspects are considered for a closer examination. In order to provide more sophisticated recommendation features, the crowdsourcing ontology should offer mechanisms for:

- analyzing social network relationships, such as friendships and work relationships (F-03.a), and
- evaluating activities within online communities, in particular social interaction and communication among the community members (F-03.b).

4.2.1.4 Transaction Transparency and Quality Control (F-04)

Crowdsourcing is often a complex process, which addresses diverse participants who range from amateurs to experts, requires a variety of resources from the crowd, such as their creativity, knowledge, or money, involves several incentive methods, and uses various schemes to evaluate the users and their contributions. Most crowdsourcing processes necessitate a good deal of experimentation, performance evaluation, and adjustment to work efficiently (Kittur et al., 2012; Liu et al., 2012). A fundamental challenge in the design of workflows is how to decompose a single complex crowdsourcing task into multiple smaller subtasks and how to combine them into one or more workflows (Kulkarni et al., 2012). These subtasks can be chained using either parallel, sequential, or iterative processing (Dai et al., 2010). Finally, all partial contributions must be efficiently aggregated to an overall contribution. Therefore, the last top-level-requirement that guides the development process is:

F-04: A crowdsourcing ontology should include semantic entities that the requester of a task can consult and statistically evaluate in order to optimize the crowdsourcing activity.

Deriving semantic entities that meet the first three top-level requirements will certainly improve the requester's transparency of the overall workflow. However, two aspects are additionally taken into account. The ontology demands semantic entities for:

- restricting the access to a crowdsourcing task, for example, according to a certain confidentiality level (F-04.a), and
- describing the characteristics of the crowdsourcing workflow (F-04.b).

4.2.2 Non-Functional Requirements

Ontology development is a design process that is influenced by design decisions. These design decisions should be guided by a set of ontology design criteria. Gruber (1995) proposes five criteria to support the design and evaluation process of ontologies (Table 3). These criteria provide also a basis for the design of the enterprise crowdsourcing ontology.

Table 3: Design criteria according to Gruber (1995)

Code	Design criteria	Description
D-01	Clarity	An ontology should effectively communicate the intended meaning of the defined concepts, relationships, and attributes and should be independent from social and computational contexts. A clear documentation and shining examples may prevent the misunderstanding of the semantic entities.
D-02	Coherence	An ontology should be logically consistent for both the formal and informal descriptions of the concepts, relationships, and attributes.
D-03	Extensibility	An ontology should offer a conceptual foundation that allows adding new concepts, relationships, and attributes without revising the existing definitions.
D-04	Minimal encoding bias	An ontology should be specified at the knowledge level without depending on a particular symbol-level encoding.
D-05	Minimal ontology commitment	An ontology should make as few statements as possible about the world being modeled.

Additionally, Gómez-Pérez (1996) suggests a set of principles to ensure that the ontology properly implements the functional requirements and competency questions (Table 4). Although this set of principles has some overlap with Gruber's list of design criteria, it also extends the list at some point. To assure that an ontology is well-verified, it can be checked for inaccuracies in its architecture (A), in its lexicon and syntax (L), and in its content (C).

Table 4: Verification criteria according to Gómez-Pérez (1996)

Code	Verification criteria	Description
A-01	Soundness	An ontology should follow the principles of design of the environment in which the ontology is embedded (see design criteria D-01 to D-05)
L-01	Correctness	The ontology and its classes and properties should be lexically and syntactically correct.
C-01	Consistency	An ontology should not lead to contradictory conclusions from valid input data.
C-02	Completeness	An ontology should be semantically complete and should cover all concepts, relationships, and attributes of the real world that are relevant for the purpose and scope of the ontology.
C-03	Conciseness	An ontology should only gather useful and concise information.
C-04	Expandability	see Extensibility (D-03)
C-05	Sensitiveness	An ontology should be robust to changes, for example, when including or modifying a class or property.

Besides the before mentioned criteria, an additional set of aspects is recommended to improve the acceptance and dissemination of the ontology (Table 5). These are mainly related to the issue of seamless data integration and exchange across diverse social software, business, and crowdsourcing applications, as well as data exchange between them. Crowdsourcing solutions often require the most recent data that exist in external business applications, such as enterprise dictionaries, knowledge repositories, or expert systems (Vukovic, Laredo, & Rajagopal, 2010).

Table 5: Additional criteria for the ontology

Code	Additional criteria	Description
B-01	Compatibility	The ontology should be compatible with existing W3C standards and recommendations, such as RDF and OWL.
B-02	Independence	The ontology should provide an abstraction level that is independent from different syntax, such as RDF/XML, RDFa, Turtle.
B-03	Human readability	The ontology should not only be machine-readable but also human-readable, lightweight and simple to use (see also D-01)
B-04	Availability	The ontology should be widely disseminated by making it freely available for the general public.
B-05	Integration	The ontology should improve the data integration and exchange between a crowdsourcing system and other information and communication technology (ICT) systems. On this account, the ontology should reuse existing Semantic Web vocabularies or standards.

It is worth to remark that a proper ontology will not comply equally with all aforementioned criteria and that some of the criteria are even mutually contradictory, such as D-05 and C-02 (cf. Vrandečić, 2009, p. 295). Therefore, these criteria must be considered with care when designing an ontology.

This chapter introduced informal competency questions and defined functional as well as non-functional requirements which the ontology should comply with. The competency questions and the functional requirements will later lay the foundation for evaluating the ontology (Section 6.1 and 6.2). Additionally, the functional requirements will be applied within the system review to examine the capabilities of current crowdsourcing systems to describe semantics of crowdsourcing data (Section 5.1.1). As the non-functional requirements are difficult to measure, they are only used to guide the process of the ontology development.

5 ONTOLOGY DEVELOPMENT

This chapter explains the building process of the ontology. It contains the following steps: conceptualization (Section 5.1), domain capture (Section 5.2), integration (Section 5.3), and implementation (Section 5.4). Although these steps are described successively, the building process follows not a rigorous linear, but an evolutionary approach.

5.1 CONCEPTUALIZATION

The purpose of *conceptualization* is to structure the domain knowledge of enterprise crowdsourcing. Conceptualization was only indirectly suggested by Uschold and King (1995), however, several researchers claimed the importance of providing a set of intermediate representations on different abstraction levels (Gómez-Pérez, Fernández, & de Vicente, 1996). The conceptualization forms the basis for the domain capture (Section 5.2).

To elicit potential candidates of semantic entities for enterprise crowdsourcing, a system review of 15 crowdsourcing applications is conducted (Section 5.1.1). Additionally, a review of current crowdsourcing literature is adopted to identify further key concepts, relations, and attributes (Section 5.1.2).

5.1.1 System Review

The overall goal of the system review is threefold. First, the review provides an impression of what types of semantic entities are currently utilized by existing crowdsourcing applications. Second, the review shows how each of these entities addresses the functional requirements stated in section 4.2.1, and thus, also identifies potential gaps in meeting them. Finally, the review yields a set of essential elements that are covered by all systems. An overview of the selected platforms including the application domains, the types of tasks that are processed on the platform, the modes of deployment (standalone, intranet, internet), and the availability of an application programming interface (API) are given in Table 6.

Table 6: Overview of common crowdsourcing platforms

Platform	Domain	Type of task	Deployment	API
Amazon mTurk	Microtask	verify data, de-duplicate data, collect data, train algorithm, categorize data, classify data, rate data, rank data, test search relevance, test product usability, research, moderate content, create content, transcribe audio or video, translate content	internet	YES
Atizio	Open innovation and co-creation	generate ideas, evaluate ideas, implement ideas	intranet, internet	NO
crowdSpring	Design	design logo, create website, design print product, find company name	internet	NO
CrowdWorx	Microtask	forecast data (sales, demands, revenues, market share, sentiments, costs, price, development time, delivery time, etc.), evaluate product ideas	stand-alone portal, intranet, internet	YES
designenlassen	Design	design product (logo, website, business card, corporate identity, banner ad, poster, advertisement, flyer), find company name, slogan, or domain name	internet	NO
elance	Job marketplace	create or translate content, make design or multimedia product, create engineering or manufacturing specification, provide customer support, offer financial, marketing, sales, or legal services	internet	YES
Gengo	Translation	translate content in multiple languages	internet	YES
Innocentive	Open Innovation and Co-creation	do brainstorm activity, generate ideas, solve complex theoretical and practical problem, create prototype	intranet, internet	NO
MobileWorks	Microtask	generate lead, categorize data, digitize document, collect feedback, label data, run research survey, test usability, carry out quality assessment	internet	YES
oDesk	Job marketplace	create website, develop software, design information system, write and translate content, do administrative support, design multimedia product, provide customer support, offer financial, marketing, sales, or legal services	internet	YES
Seedmatch	Crowdfunding	fund start-up company	internet	NO
Startnext	Crowdfunding	fund small-sized private project	internet	YES
UnserAller	Open innovation and co-creation	generate ideas, develop new product, conduct survey	internet	NO
uTest	Software testing	functional testing, security testing, load testing, localization testing, usability testing	internet	NO
ziptask	Job marketplace	do general office work, develop software, make design or multimedia product, do research, create content	internet	NO

During the system review, the user frontends, and if available, the APIs of the 15 crowdsourcing platforms are examined. All found semantic entities are clustered according to the functional requirements. Most of them support the task specification (F-01), the task allocation (F-02), as well as the transaction transparency and quality control (F-04) (for a summary see Table 7; a complete overview is given in Appendix A).

Table 7: Utilization of semantic entities in crowdsourcing systems

Functional requirements	**Semantic entities**
Structured task specification (F-01)	• Semantic entities that are used to describe crowdsourcing projects or tasks are, for example, the title, the description, the goal, the task type, the visibility, acceptance criteria, a set of instructions, some keywords or categories, and links to required external resources. • There are additional elements to specify time aspects of crowdsourcing tasks, such as the start date, the end date, the estimated duration, and the priority or urgency level. • The reward of crowdsourcing tasks is determined by the amount and type of payment.
Efficient task allocation (F-02)	• A requester may narrow down the target audience of a crowdsourcing task according to the qualification, the interest, the number of approved tasks, the spoken language, the location, the reputation, and some demographic data (minimum age, maximum age, gender). • A user may search for a crowdsourcing task that covers time aspects (submission time, closure time, and duration), the type and amount of reward, the accepted language, the location, the product or project category, the most recommended or supported project, the required qualification and interest.
Transaction transparency and quality control (F04)	• Selecting preferred or blocking malicious users as well as checking the expertise of users with pre-evaluation tasks or recent crowdsourcing activities maintain the overall quality of the contributions. • The number of assignments and the completion status of a crowdsourcing task are used to monitor and control the crowdsourcing process. • The type of workflow specifies how multiple tasks of a crowdsourcing project are processed.

Although some of the semantic entities, such as qualification, interest, or availability, might be utilized for dynamic and proactive team building (F-03), less support in that regard is provided by the crowdsourcing application itself. Data integration and exchange (B-05) are largely maintained by APIs. Seven out of the 15 analyzed crowdsourcing platforms provide an API to get access to the functions and data of the platform. However, none of the studied crowdsourcing applications applies existing Semantic Web vocabularies or standards, such as Dublin Core, FOAF, or GoodRelations, to use data from or make it available for other business applications easily.

The identified semantic entities during the system review form the basis for the data dictionary that is created in the domain capture step (Section 5.2). In the next section, a theoretical study will complement the results of the review.

5.1.2 Literature Review

In this section, a preliminary set of semantic entities is derived from scientific literature. In favor of a holistic view on enterprise crowdsourcing, the author analyzes the literature from both a system-oriented perspective (Section 5.1.2.1) and a process-oriented perspective (Section 5.1.2.2).

5.1.2.1 System-oriented Perspective

In the scientific literature, the term *crowdsourcing system* is currently used inconsistently. (Hetmank, 2013a). This makes it difficult to derive a universal set of semantic entities from the literature. As summarized in a previous study of the author, several definitions exist that vary in the level of detail and address different perspectives of a crowdsourcing system, namely the organizational, the technical, the process-oriented, and the human-centric perspective (Hetmank, 2013a). To achieve a shared understanding of crowdsourcing systems, a first conceptual model of typical components and functions was presented in the same study (Figure 2).

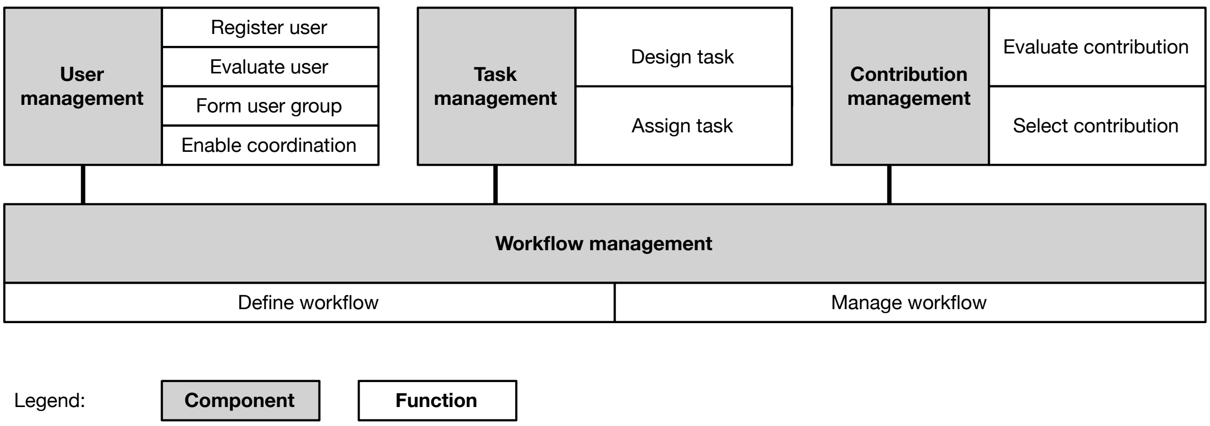

Figure 2: Conceptual model of a crowdsourcing system

The model contains four principal components: user management, task management, contribution management, and workflow management. Each of the components includes functions that should be considered when implementing a crowdsourcing system. This model is now applied as an auxiliary instrument to determine a preliminary set of semantic entities that are potential candidates for the enterprise crowdsourcing data dictionary.

Table 8: Classification schemes

Source	Dimension
Doan, Ramakrishnan, and Halevy (2011)	• Nature of collaboration • Type of target problem • Design of incentive mechanism • Task complexity • Impact of contribution • Approach to combine solutions • Method to evaluate users • Degree and distribution of manual effort • Role of human users • Type of architecture
Geiger, Rosemann, and Fielt (2011); Geiger, Fielt, Rosemann, and Schader (2012)	• Treatment of external elements • Value of the relationship with an external element
Erickson (2011)	• Distribution over time • Distribution over space
Yuen, King, and Leung (2011)	• Application
Rouse (2010)	• Nature of the crowdsourcing task / supplier capabilities • Distribution of benefits • Nature of the motivation to participate
Schenk and Guittard (2011)	• Nature of the crowdsourcing process • Type of task
Corney, Torres-Sanchez, Jagadeesan, and Regli (2009)	• Nature of the task • Nature of the crowd • Nature of the payment
Geiger, Seedorf, Schulze, Nickerson, and Schader (2011)	• Pre-selection • Accessibility • Aggregation • Remuneration
Erickson, Petrick, and Trauth (2012a, 2012b); Erickson (2012)	• Organizational uses of the crowd • Common task • Crowd knowledge • Value of the crowd • Preferred crowd location
Quinn and Bederson (2011)	• Motivation • Quality control • Aggregation • Human skill • Process order • Task-request cardinality
Malone, Laubacher, and Dellarocas (2010)	• Goal • Staffing • Incentives • Process
Zwass (2010)	• Performers • Motivation • Structural task complexity • Intellective demands • Effort intensity • Time frame • Aggregation

During the last few years, researchers gained a deeper understanding by classifying the wide range of emerging crowdsourcing applications according to different dimensions (Table 8). To benefit from their insights, twelve of the most prominent crowdsourcing taxonomies and classification schemes are analyzed in two steps.

In the first step, each of the dimensions that are found in the classification schemes is mapped onto the components and functions of the conceptual model (Figure 2). Thus, the various types of components and functions of a crowdsourcing system are used as a coding schema to identify the functional roles that correspond to a dimension (see Appendix B, column 3 “Dimension (characteristic)” and column 4 “Component (function)”).

In the second step, based on the functional roles, for each of the dimensions one or more semantic entities are derived (see Appendix B, column 5 “Semantic entity”). These entities provide a basis for a clear description of and a distinction between diverse crowdsourcing applications. A summary of the derived semantic entities is presented in Table 9. The entities are grouped into six categories: task characteristics, time aspects, task requirements, motivation, quality, and workflow. The categories and the assigned entities lay the foundations for the domain capture step of the ontology building process (Section 5.2).

Table 9: Semantic entities derived from existing classification schemes

Category	Semantic entities
Task characteristics	goal, type of action, complexity level, impact level, category, target audience
Time aspects	submission time, closure time, Estimated time of duration, latency
Task requirements	Human requirement, Technical requirement
Motivation	reward mechanism
Quality	evaluation mechanism
Workflow	type of aggregation, visibility, sequence of work, accessibility, number of assignments, interaction mode

5.1.2.2 Process-oriented Perspective

In this section, the author introduces an idealized and generic crowdsourcing process that aims to represent various types of crowdsourcing activities. With the aid of this generic process, common key concepts and relationships are identified. The process is based on a consolidated view on descriptions of other research papers (Table 10), however, compared to them, it has a higher level of granularity and considers also additional aspects, such as providing feedback on the requesters’ task specifications or the participants’ contributions.

Table 10: Synopsis of descriptions of crowdsourcing processes

Source	Process tasks
Gassmann (2010)	preparation, initiation, execution, evaluation, exploitation
Geiger, Seedorf, Schulze, Nickerson, and Schader (2011)	preselection of contributions, accessibility of peer contributions, aggregation of contributions, remuneration for contributions
Khasraghi and Tarokh (2012)	submit task, select task, submit result, return result, reward
Vukovic (2009)	submit request, query providers, negotiate request criteria, bid, participate, execute request, validate completion, pay for request, charge for request, submit ratings
Zhao and Zhu (2012)	submit task, negotiate, inquire request, push & pull, participate, bid, validate, reward

The proposed generic crowdsourcing process is composed out of four sub-processes: (i) the task specification, (ii) the task allocation, (iii) the contribution management, and (iv) the post-task management (Figure 3):

I. *Task specification:* When defining a task of a crowdsourcing project several design characteristics have to be taken into account, such as a set of instructions, some acceptance criteria, the definition of a target audience, the determination of human and technical requirements, as well as the specification of reward and evaluation mechanisms (see also the task characteristics in Table 9 of the previous section). The next step within this sub-process is to decide whether a task is split into several subtasks, or multiple tasks are bundled into one single task. Finally, the crowdsourcing task is submitted to the crowd.

II. *Task allocation:* The second sub-process focuses on assigning a crowdsourcing task to one or multiple potential candidates. This can be pursued from two perspectives. On the one hand, the participants can search for a task. On the other hand, either the requester or the recommender engine of the crowdsourcing system can propose a crowdsourcing task to a latent user. If the task specification is not clear or imprecisely defined, the participant may ask for feedback that optionally entails the redefinition of the crowdsourcing task. Depending on the configuration of the crowdsourcing process, a candidate may either apply for a crowdsourcing task or directly select one. If the users apply for a task, the requester (manually) or the crowdsourcing system (automatically) pre-evaluate them and select one or more that are appropriate for solving the task.

III. *Contribution management:* After the crowdsourcing task is assigned to a number of suitable participants, each of them starts submitting one or more contributions. The requester or the peers themselves can then provide feedback on the contributions, which may lead to a resubmission of the contribution. Finally, all contributions are evaluated and those are selected that are ready for aggregation or for the final solution. Computer algorithms often support the evaluation, selection and aggregation of the contributions. Sometimes, these process steps are also turned over to third party organizations.

IV. *Post-task management:* Finally, if the task is solved properly, the users might be eligible to get a reward according to the defined incentive scheme. Each of the participants may also get a final evaluation that can be used to enhance the task recommendation and workforce selection of future crowdsourcing activities.

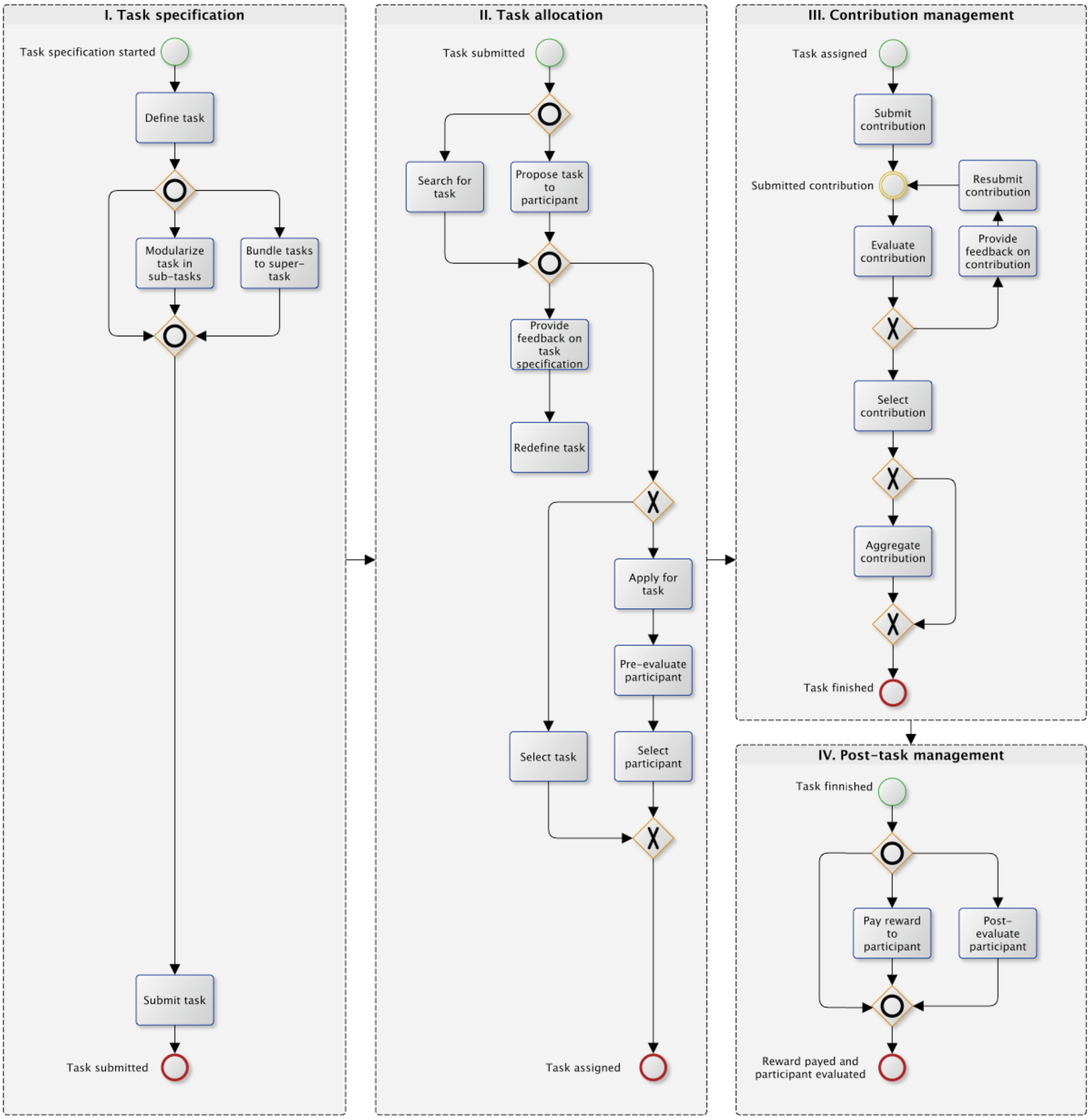

Figure 3: Idealized and generic crowdsourcing process

It should be noted that many of the process tasks that are presented in this process could either be completed manually by both the requester and the crowd, or automatically by the crowdsourcing system. For example, the process tasks "propose task to participant" and "evaluate contribution" can be initiated by all three agents: the requester, the participant, and the crowdsourcing system. As a consequence, no specific roles are associated with each of the process tasks.

Based on the description of the crowdsourcing process, seven key concepts (project, task, participant, requester, evaluation mechanism, reward mechanism, contribution) and three core relationships (include, submit, issue) are determined (Table 11). In addition to the semantic entities that resulted from taking a system-oriented perspective (Section 5.1.2.1), these concepts and relationships provide a foundation for the domain capture step of the ontology building process (Section 5.2).

Table 11: Key concepts derived from the generic crowdsourcing process

Key concept	Description
Project	A crowdsourcing project *includes* one or more crowdsourcing tasks.
Task	A crowdsourcing task has a set of task characteristics, task requirements, reward mechanisms, and evaluation mechanisms.
Participant	A participant *submits* one or more contributions to a crowdsourcing task.
Requester	A requester *issues* a crowdsourcing project.
Evaluation mechanism	An evaluation mechanism specifies who evaluates what with which method at what time.
Reward mechanism	A reward mechanism specifies the amount, type and nature of the reward.
Contribution	A contribution belongs to exactly one crowdsourcing task.

5.2 DOMAIN CAPTURE

During conceptualization (Section 5.1), the enterprise crowdsourcing domain was informally perceived and described adopting two approaches: a system review and a literature review. In the *domain capture*, the fragmented results of the conceptualization are now organized and transformed into a semi-formal specification using a set of intermediate representations (Gómez-Pérez et al., 2005 p. 130). These intermediate representations provide the basis for implementing the ontology with a specific coding or schema definition language (Section 5.4). As a first intermediate representation, a data dictionary of the key terms is created (Gómez-Pérez et al., 1996). Appendix C includes the data dictionary with all essential concepts, relations, and attributes for the enterprise crowdsourcing domain. Each term is identified by a clear and distinct name and has a consistent description that offers the meaning of the term. Based on the proposed data dictionary, a second enhanced intermediate representation in form of a semi-formal Unified Modeling Language (UML) diagram is designed to model the concept hierarchy and the binary relations between the concepts (Figure 4). Due to not dealing with a large-scale ontology and unlike suggested in the *Methontology* approach by Gómez-Pérez et al. (2005), this is done in one single step. The schema also includes a first suggestion of which data type to use for a certain data property.

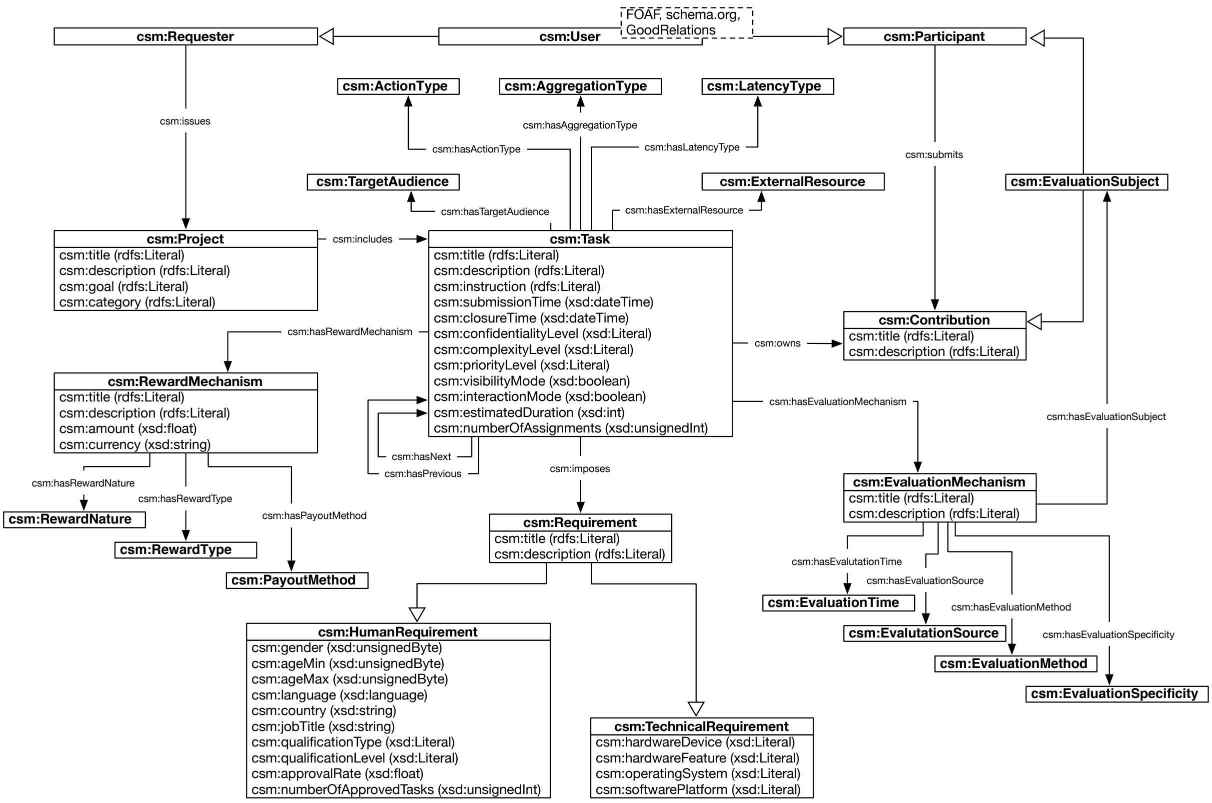

Figure 4: Semi-formal specification of the ontology

The ontology has been already introduced in a previous article of the author (Hetmank, 2014b) and a detailed documentation of the concepts and properties can be found in the CSM ontology specification.[9] A brief description of the ontology is given in the subsequent paragraph.

The CSM ontology considers two roles for an enterprise crowdsourcing activity: the requester (csm:Requester) and the participant (csm:Participant). Both are modeled as subclasses of a universal user class (csm:User) that acts as a link between the ontology and the

[9] The CSM ontology specification can be found at http://www.purl.org/csm/.

concepts of other vocabularies. For current purposes, the person and organization concept of the FOAF and schema.org vocabulary (foaf:Person, foaf:Organization, schema.org/Person, schema.org/Organization) as well as the business entity concept of the GoodRelations vocabulary (gr:BusinessEntity) are associated to benefit from the reuse of their properties (for more details see also section 5.3). Each crowdsourcing project (csm:Project) that a requester issues includes one or more crowdsourcing tasks (csm:Task). The sequence of multiple crowdsourcing tasks within a project can be determined by using the csm:hasNext and csm:hasPrevious property. The task concept comprises several data properties to specify the task characteristics, such as instruction (csm:instruction), submission and closure time (csm:submissionTime, csm:closureTime), or the confidentiality and priority level (csm:confidentialityLevel, csm:priorityLevel). Some aspects of the crowdsourcing task are designed as independent classes to enhance the semantics of the overall crowdsourcing ontology. These classes are, for instance, the reward and evaluation mechanism (csm:RewardMechanism, csm:EvaluationMechanism), or the human and technical requirement (csm:HumanRequirement, csm:TechnicalRequirement).

5.3 INTEGRATION

Before implementing an ontology for a certain purpose and domain, ontologist should look for existing vocabularies and standards that might be partially adapted or completely reused (Uschold & King, 1995). Thus, several semantic vocabularies and standards are assessed (Section 5.3.1) and the implication for the design is discussed (Section 5.3.2).

5.3.1 Semantic Vocabularies and Standards

Academics and practitioners worldwide have developed a vast number of semantic vocabularies and standards. To narrow down potential candidates for reuse, several selection criteria are set beforehand:

- The vocabulary should be documented in detail.[10]
- The vocabulary should be preferably popular, for example, highly referenced by Semantic Web documents.[11]

[10] An overview of well-documented ontologies can be found at at http://www.w3.org/wiki/Good_Ontologies.

[11] Hepp (2007) provides a snapshot of popular ontologies that are highly ranked by the Semantic Web search engine swoogle (http://swoogle.umbc.edu/).

- The vocabulary should address aspects of the knowledge domain of enterprise crowdsourcing, which means that it should give details on the main concepts, such as the crowdsourcing user (either the requester or the participant) or the crowdsourcing task, as well as the relations between these concepts.

Taking the aforementioned selection criteria into consideration, 15 semantic vocabularies for conducting further investigations are identified (Table 12).

Table 12: Potential vocabularies for reusing in the crowdsourcing domain

Vocabulary	Implementation	Application domain
Activity Streams (activitystrea.ms)	JSON, XML, RDF	vocabulary to describe activities in social software applications and services
Contextualized Attention Metadata (CAM)	XML	vocabulary to describe the objects that attract a user, the actions a user performs with these objects, and the use contexts (contextual information)
Description of a Project (DOAP)	RDF	vocabulary to describe software projects
Dublin Core (DC)	HTML, XML, RDF, plain text	vocabulary to describe documents
Friend of a friend (FOAF)	RDF	vocabulary to describe users (user profiles), their relationships (social network), and objects they create
GoodRelations (GR)	microdata, RDFa, OWL	vocabulary for product, price, store, and company data (e-commerce)
hCalendar/h-event	microformats	vocabulary to describe events
hCard/h-card, vCard	microformats (HTML syntax), RDF	vocabulary to describe people, companies, and organizations (user profiles)
Open Social	JSON, XML	vocabulary to describe social network information and services
Provenance ontology (PROV)	OWL, XML, plain text (human readable version)	vocabulary to describe entities, activities, and people (provenance data)
schema.org	microdata (HTML syntax)	a collection of vocabularies to describe persons (user profiles), organizations, products, events, and actions performed on objects
Semantically-Interlinked Online Communities (SIOC)	RDF	vocabulary to describe online communities, such as blogs, discussion forums and mailing lists
Web Services Business Process Execution Language (WS-BPEL)	XML	vocabulary to describe business processes
XML Process Definition Language (XPDL)	XML	vocabulary to describe business processes
XHTML Friends Network (XFN)	HTML (*rel* attribute)	vocabulary to describe human relationships (social network)

These vocabularies aim to define:

1. people, organizations, and information objects (DOAP, DC, GoodRelations, hCard/vCard, schema.org),
2. events and contextual information (activitystrea.ms, hCalender, CAM, PROV),
3. social networks and online communities (FOAF, Open Social, XFN, SIOC), or
4. business processes and workflows (WS-BPEL, XPDL).

In the next subsections, the vocabularies are briefly reviewed to assess their applicability to the crowdsourcing domain.

5.3.1.1 People, Organizations, and Information Objects

The *Description of a Project* (DOAP) vocabulary can be useful if the crowdsourcing activity is a software development project. The vocabulary is mainly intended to describe existing rather than future software projects. However, some of the attributes of the vocabulary, such as the operating system, the platform, or the programming language, are helpful in defining the technical requirements for a crowdsourcing project.

The *Dublin Core* (DC) specification is one of the best-known metadata sets for describing web resources, in particular documents. The metadata elements of DC can be applied to specify various aspects of the crowdsourcing contributions and the required external resources, for example, the creator, the subject, the audience, rights and links to other documents.

The *GoodRelations* (GR) vocabulary is a popular and widespread ontology for the e-commerce domain. Especially the concepts *business entity* and *business entity type* are candidates for reuse. Both concepts allow a detailed description of the users involved in a crowdsourcing activity. The concept *payment method* and the concept *price specification* can extend the definition of the reward mechanism of a crowdsourcing project.

The *hCard/vCard* open format offers attributes to record address and contact information of a person, a company, or an organization. It offers similar attributes as the concept *person* of the schema.org and the FOAF specification and can be valuable for creating user profiles of a crowdsourcing user.

The *schema.org* specification offers an extensive collection of classes and attributes to describe persons, organizations, products, events, and actions. Over 70 action types are defined in the schema.org specification. Some of them, such as assess, create, find, organize, and update, can also be suitable to specify a certain crowdsourcing task. Especially the concepts *person* and the concept *organization* are valuable candidates for reuse.

5.3.1.2 Events and Contextual Information

The *Activity Streams* (activitystrea.ms) standard provides a set of semantic entities to describe activities in social software applications and services. The standard offers a wide range of object types (e.g., article, file, image, note, review, video) that can be adopted to describe different kinds of crowdsourcing contributions as well as external resources. The action types (e.g., add, delete, receive, tag, share) that are introduced in the standard are mainly related to web contents, documents, and persons and provide less value to describe crowdsourcing activities.

The *Contextualized Attention Metadata* (CAM) schema contains an essential set of concepts and relationships to describe contextual information. Although the model does not include the necessary semantics to define specific facets of enterprise crowdsourcing, it can be applied on a higher level of abstraction to describe crowdsourcing events.

The *hClalendar* vocabulary contains properties to define events. The elements of the vocabulary can be adopted to describe time aspects (e.g., start data, end data, duration) and location aspects of a crowdsourcing activity.

The *Provenance ontology* (PROV-O) consists of classes and properties to describe provenance information and has currently the status of a W3C recommendation. Similar to the CAM schema, the PROV-O is very generic and mainly suited to describe aspects on a high level of abstraction.

5.3.1.3 Social Networks and Online Communities

The *Friend of a friend* (FOAF) vocabulary is a lightweight, and highly accepted specification to describe social networks. The concepts *person* and *organization* can be utilized to describe the participants and the requesters of a crowdsourcing task. Moreover, the concept *group* and the object property *member* can model the aspect of team building in a crowdsourcing activity. The users' social network itself can be described by the object property *knows*. Three types of concepts, namely *document*, *image*, and *project* can be employed to define crowdsourcing contributions and external resources.

The *Semantically-Interlinked Online Communities* (SIOC) ontology provides semantic entities to describe discussion methods, such as blogs, forums and mailing lists. In the crowdsourcing domain, the elements can be used to represent communication interactions among the participants or between the requester and the participants.

The *XHTML Friends Network* (XFN) metadata profile contains 18 different alternatives to describe relationships between people. The differentiation between friendship, professional relationship, or family relations can be helpful to support the team building process in a crowdsourcing activity.

5.3.1.4 Business Processes and Workflows

The *Web Services Business Process Execution Language* (WS-BPEL) is an XML-based standard for the description of business processes. The activities of the business processes are implemented as web services, which are platform independent software components designed to support distributed business applications. The basic version of the WS-BPEL standard, however, does not consider human interaction. Thus, two extensions of the WS-BPEL standard, namely BPEL4People and WS-Human-Task, are defined to model the concept of human tasks that are accomplished by people (OASIS, 2010). Although human interactions are now supported, little is known how to apply these standards in the domain of crowdsourcing (Schall, 2012).

Another XML-based standard to describe business processes is the *XML Process Definition Language* (XPDL). Whereas BPEL focuses mainly on the orchestration of web services, XPDL is designed at its core to also handle workflows performed by people. The standard allows to define user tasks (TaskUser activity) where a human actor performs the task with the assistance of a software application (WfMC, 2012, p. 112).

The main weakness of both standards, the WS-BPEL and the XPLD is that they do not provide the necessary granularity to describe a crowdsourcing task in detail. However, to improve the task allocation, more information is required, such as the given reward, some time constraints, or the necessary qualification to solve the task.

5.3.2 Implications for the Design

The analysis of the existing semantic vocabularies and standards provides valuable insight and leads to two design impacts for the development of the ontology. First, the possibility to describe users and their social networks within enterprise crowdsourcing environments is already well supported by standards, such as FOAF and schema.org. Thus, to empower the CSM ontology with social profiles and networking features, these two standards will be re-used. Second, to link the CSM ontology to the semantics of the ecommerce world, the business entity concept of the GoodRelations ontology is adopted.

5.4 IMPLEMENTATION

The *implementation* begins with choosing a representation language for the ontology. The Web Ontology Language (OWL) is applied to formalize a prototype of the ontology. When considering the non-functional requirements, especially the aspects of compatibility, integration and expandability, two reasons lead to the choice of OWL. First, OWL is a well-established and accepted standard and thus not only increases the chance of dissemination and reusability of the ontology in the future, but also complies with the requirement to ensure interoperability between different information systems. This is mainly due to the fact that OWL lies within the responsibility of the World Wide Web Consortium (W3C), an international standards organization. Second, OWL is a very expressive representation language, and hence offers the necessary condition for future changes and extensions of the ontology.

For the implementation of the ontology, the open source ontology development editor *Protégé* was applied. The tool supports the overall modeling process of the ontology by providing several useful functions. For example, it offers functions to efficiently create and annotate semantic entities of an ontology, to graphically visualize an ontology, and to automatically generate a human-readable documentation out of the ontology.

To ensure the quality and, in particular, to address the non-functional requirements (Section 4.2.2), various design rules to formalize the ontology are considered:

- A persistent uniform resource locator (PURL) is used to redirect the location of the requested ontology. Thus, the PURL offers a constant reference to the specification and documentation of the ontology, while the actual address of the website can change in the future.[12]
- The uniform resource identifier (URI) prefix *csm* is chosen in such a way that it does not conflict with existing namespaces. To prevent multiple usage of the same namespace

[12] The PURL of the CSM ontology is http://www.purl.org/csm/.

for different URIs in future, the new prefix mapping *csm* is registered at the namespace lookup service developed at the Digital Enterprise Research Institute (DERI).[13]

- The axioms of the ontology are properly indented and grouped in classes, object properties, data properties, and individuals (see Appendix E).
- The ontology uses URIs that can be interpreted by human readers.
- The classes of the ontology are labeled with singular nouns (sometimes with a preceding adjective) and start with a capital letter, such as "Task".
- The object properties of the ontology are labeled either with verbs in the third person, such as "issues", or with a combination of the prefix "has" and the class name to which it is related, such as "hasRewardType". All object properties start with lower case.
- The data properties of the ontology are labeled with plural nouns and start with lower case.
- If a semantic entity is composed of multiple words each subsequent word begins with capital letters, such as "qualificationType".
- Only data types that are widely supported by various Semantic Web tools are applied, for example, the element "estimatedDuration" applies xsd:int instead of xsd:duration.

The designed CSM ontology finally contains 24 classes, 22 object properties, and 30 data properties to describe the key aspects of a typical enterprise crowdsourcing activity. Additionally, 51 individuals are suggested, which may be extended in future versions of the ontology. All semantic entities of the ontology are supplemented with additional annotation information, whereas the rdfs:comment property is applied to explain the meaning of the entity and the rdfs:seeAlso property offers literature references and suggestions for further reading.

[13] The namespace lookup service can be found at http://prefix.cc/about.

6 EVALUATION

Evaluating an ontology is an essential activity in the ontology engineering process that should not be neglected. As Gómez-Pérez et al. (2005) assert "it is unwise to publish an ontology or to implement a software application that relies on ontologies written by others (even yourself) without evaluating first its content [...]" (p. 178). Ontology evaluation can be divided into ontology verification and ontology validation. Whereas ontology verification focuses on building an ontology correctly, ontology validation proves that the set of all semantic entities of an ontology really corresponds to the domain of the real world that should be modeled (Gómez-Pérez, 1996).

There exist various approaches on how to verify and validate an ontology, which fall mainly in the following categories (Brank, Grobelnik, & Mladenić, 2005; Kayed, 2013):

1. How well does the ontology fit the domain of knowledge for which it is created?
2. How well does the ontology perform in the context of application?
3. How well does the ontology meet a set of criteria, standards, and requirements?
4. How well does the ontology work compared to other ontologies and vocabularies in the same domain?
5. Which characteristics does the ontology show according to certain ontology metrics?

Unlike other software products, ontologies face the challenge that their developers cannot simply compile, run and test them in the context of their predefined application domain (Vrandečić, 2009). Another difficulty results from the fact that within the context of the somehow uncontrolled Semantic Web, ontologies are often "used and extended in ways not expected by the [original] creators" (Vrandečić, 2009, p. 294). Due to the difficulties in evaluating an ontology, a multifaceted evaluation is pursued that applies multiple methods to address the five above mentioned questions.

To answer the first two questions that focus on the domain of knowledge and the context of application, three use case scenarios are shown on how the CSM ontology can be used to query enterprise crowdsourcing data on the Semantic Web (Section 6.1). According to question three, the designed ontology should satisfy the previously established requirements. Additionally, considering question four, the performance of the developed ontology should be compared to other ontologies and Semantic Web vocabularies. To address question tree and four, the CSM ontology and each of the semantic vocabularies presented in section 5.3.1 are contrasted with different aspects of the functional requirements introduced in section 4.2.1. To answer question five, this work finally draws the attention to three ontology metrics, which describe the general structure of the CSM ontology (Section 6.3).

6.1 TRANSFORMING INFORMAL TO FORMAL COMPETENCY QUESTIONS

An ontology can be evaluated by leveraging use case scenarios. As Obrst et al. (2007) remark, these "task-based evaluations offer a useful framework for measuring practical aspects of ontology deployment, such as the human ability to formulate queries using the query language provided by the ontology" (p. 148). In this technical report, three use case scenarios are presented:

- use case scenario 1: specifying crowdsourcing tasks (Section 6.1.1),
- use case scenario 2: finding and recommending crowdsourcing tasks (Section 6.1.2), and
- use case scenario 3: monitoring and managing crowdsourcing tasks (Section 6.1.3).

For each of the use case scenarios two example queries are presented. All example queries are written as informal competency questions, which are then followed by their corresponding formal query. The formal query is expressed in the SPARQL protocol and RDF query language (SPARQL) (Harris & Seaborne, 2013). SPARQL provides a language for querying RDF graphs via pattern matching. It is similar to the structured query language (SQL), but entails the use of RDF graphs, internationalized resource identifiers (IRIs), and XML schema data types (Della Valle & Ceri, 2011). All queries were tested with the SPARQL query engine Twinkle[14] and the SPARQL server Fuseki[15] using two sample data instances.

The first sample data instance represents a corporate translation project. It consists of two crowdsourcing tasks. The first one requests for translating a technical specification and the second one for translating an in-house memo (Figure 5). Both crowdsourcing tasks impose different language skills (French, German, and Polish) and include distinctive reward mechanisms (money and reputation points). Due to assuming a higher confidentiality, the translation of the memo is only issued to the employees of the company (internal crowd), whereas the translation of the technical specification considers also the general public domain (hybrid crowd). The submission and closure time of both tasks is equal, however, the estimated time for performing the tasks differs.

[14] The SPARQL query tool Twinkle can be found at http://www.ldodds.com/projects/twinkle/.
[15] The SPARQL server Fuseki can be found at http://jena.apache.org/documentation/serving_data/.

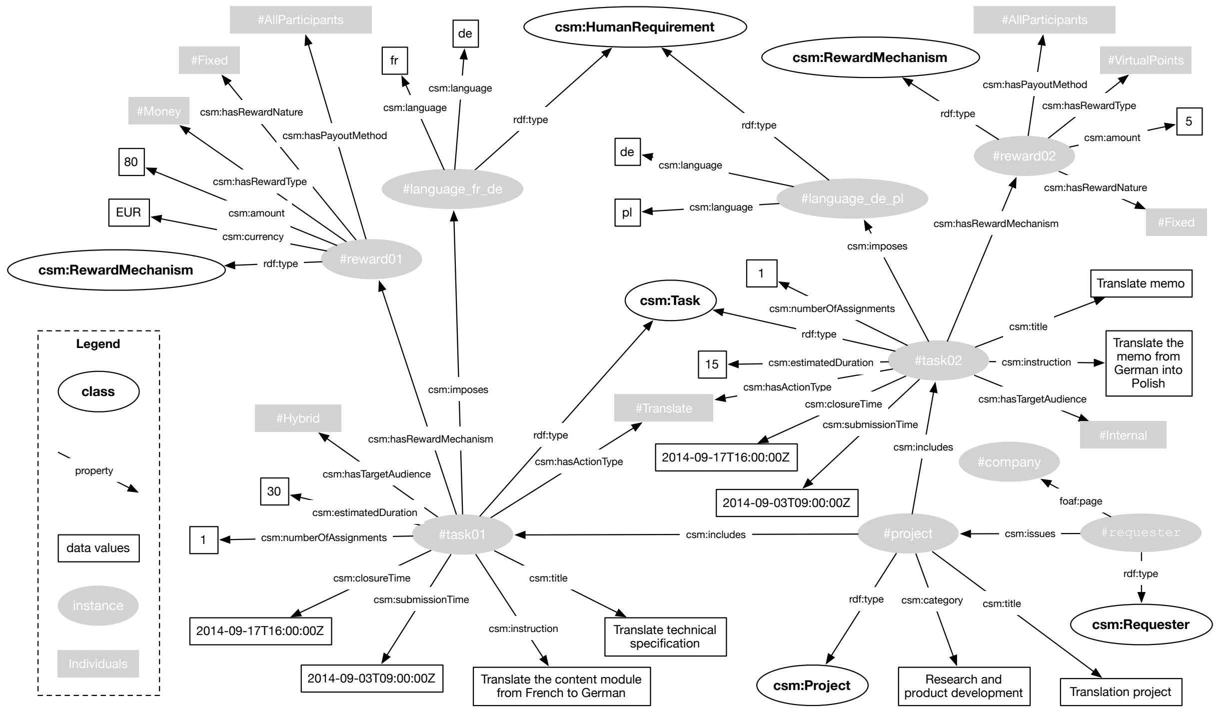

Figure 5: Translation project (sample data 1)

The second sample data instance illustrates a software development project that includes four tasks. Two of them are software development tasks requiring specific programming skills and two of them are software-testing tasks presuming that the potential participants use a certain operating system for conducting the tests (Figure 6). The software development tasks build upon one another and are processed successively, while the software-testing task can be performed side by side. Some of the tasks own contributions that are submitted by participants. Only participants of the software development tasks are evaluated. While the peers evaluate the participants of the first task after they have performed the task, the requester assesses the participants of the second task before they engage in the task.

The structured data of both sample data instances are codified using the Resource Description Framework in Attributes (RDFa) (Adida, Birbeck, McCarron, & Herman, 2013). RDFa extends the HTML syntax with a set of attributes to semantically annotate things on the Web (Sporny, 2013). RDFa introduces only a few simple HTML attributes to make statements about web resources in the form of subject-predicate-object expressions. Besides using the set of RDFa Lite attributes, namely vocab, typeof, property, resource, and prefix, also attributes of the RDFa Core are applied including content, datatype, about, and rel (Herman, Adida, Sporny, & Birbeck, 2013). The two sample data instances comply with the

CSM ontology. The HTML/RDFa source code can be found in the Appendix F (Sample Data Instance 1) and in the Appendix G (Sample Data Instance 2).

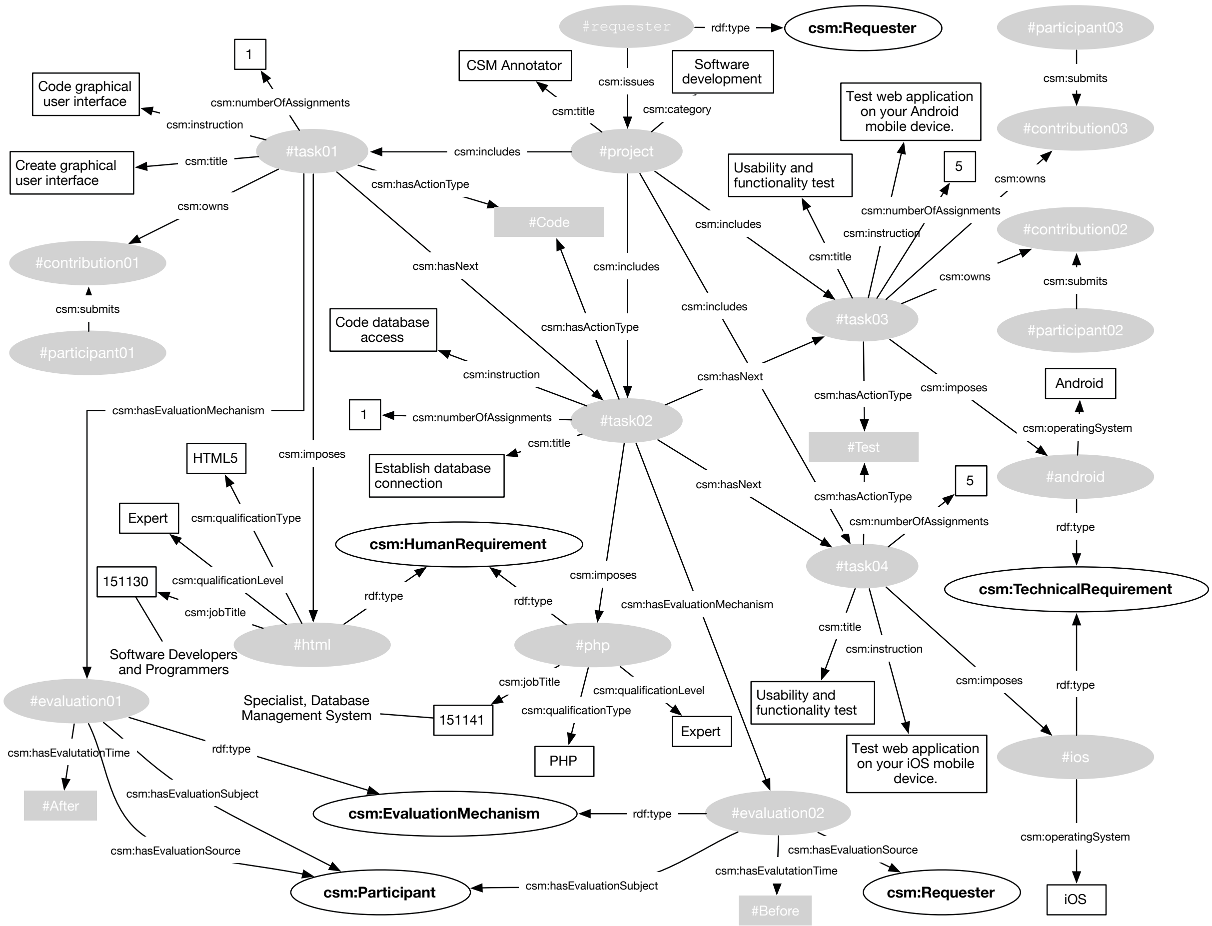

Figure 6: Software development project (sample data 2)

6.1.1 Specifying Crowdsourcing Tasks

The first use case scenario illustrates how requesters can be supported in specifying a crowdsourcing task. Requesters often face the challenge of having to define crowdsourcing tasks that will draw an audience and that will lead to the anticipated and desired quality of the contributions. As a starting point, a crowdsourcing system may apply the terminological knowledge of the ontology to guide the requester by suggesting common properties worth considering during the specification process of a crowdsourcing project. Additionally, based on the task requirements and the analysis of historical crowdsourcing data, a system may give valuable advice, for example, for choosing an appropriate incentive or evaluation mechanism. Example 1 illustrates a query for identifying reward mechanisms of previously issued translation tasks with an estimated duration of 30 minutes.

Example 1: Which type, nature and amount of reward are appropriate for a translation task that lasts approximately 30 minutes? (query applied to sample data 1)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX csm: <http://purl.org/csm/1.0#>
SELECT ?RewardType ?RewardNature ?RewardAmount ?RewardCurrency
WHERE {
   ?Task csm:title ?Title .
   ?Task csm:hasActionType csm:Translate .
   ?Task csm:estimatedDuration ?estimatedDuration .
   FILTER (?estimatedDuration = "30"^^xsd:int)
   ?Task csm:hasRewardMechanism ?RewardMechanism .
   ?RewardMechanism csm:hasRewardType ?RewardType .
   ?RewardMechanism csm:hasRewardNature ?RewardNature .
   ?RewardMechanism csm:amount ?RewardAmount .
   ?RewardMechanism csm:currency ?RewardCurrency
}
```

Listing 1: Example for querying reward mechanisms

To anticipate the potential success rate for solving a certain crowdsourcing task, requesters are occasionally interested in identifying suitable and available candidates in advance. Example 2 shows a query for finding HTML5 developers who have already been engaged and evaluated in a crowdsourcing activity.

Example 2: Who has already participated in a previous crowdsourcing project that required HTML5 skills and has been evaluated afterwards? (query applied to sample data 2)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX csm: <http://purl.org/csm/1.0#>
SELECT ?Participant
WHERE {
   ?Participant csm:submits ?Contribution .
   ?Task csm:owns ?Contribution .
   ?Project csm:includes ?Task .
   ?Task csm:imposes ?Requirement .
   ?Requirement rdf:type csm:HumanRequirement .
   ?Requirement csm:qualificationType "HTML5"@en .
   ?Task csm:hasEvaluationMechanism ?EvaluationMechanism .
   ?EvaluationMechanism csm:hasEvaluationSubject csm:Participant .
   ?EvaluationMechanism csm:hasEvaluationTime csm:After
}
```

Listing 2: Example for querying participants

6.1.2 Finding and Recommending Crowdsourcing Tasks

The second use case scenario describes how the task allocation process can be supported. Often the number of crowdsourcing tasks that are available on crowdsourcing platforms can be overwhelming. Thus, a requester may want to recommend a crowdsourcing task that fits the human capabilities or technical resources of the participants. Example 2 has already presented how potential candidates can be identified. Similar to the requesters, participants want to efficiently filter and find crowdsourcing tasks that correspond to their interests and skills. Example 3 demonstrates a query that sets the preference of the type of task to translation tasks issued to the company's employees. The task shall require French and German language skills.

Example 3: "Which translation tasks are issued internally to the employees only and require Polish and German language proficiency?" (query applied to sample data 1)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX csm: <http://purl.org/csm/1.0#>
SELECT ?Task ?Title
WHERE {
   ?Task csm:title ?Title .
   ?Task csm:hasTargetAudience csm:Internal .
   ?Task csm:hasActionType csm:Translate .
   ?Task csm:imposes ?Requirement .
   ?Requirement rdf:type csm:HumanRequirement .
   ?Requirement csm:language "pl"^^xsd:language .
   ?Requirement csm:language "de"^^xsd:language
}
```

Listing 3: Example for querying tasks that fit the participant's interests and skills

Time constraints and reward expectations may also play a crucial role for identifying appropriate tasks. Example 4 illustrates how the general public can search for tasks that last no longer than 45 minutes and offer a reward of at least 15 Euro.

Example 4: "Which public domain crowdsourcing tasks offer a reward of at least 15 Euro worth and do not take longer than 45 minutes?" (query applied to sample data 1)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX csm: <http://purl.org/csm/1.0#>
SELECT ?Task ?Title
WHERE {
   ?Task csm:title ?Title .
   {?Task csm:hasTargetAudience csm:External .} UNION
   {?Task csm:hasTargetAudience csm:Hybrid .}
   ?Task csm:hasRewardMechanism ?RewardMechanism .
   ?RewardMechanism csm:amount ?amount.
   FILTER (?amount >= "15"^^xsd:float) .
   ?RewardMechanism csm:currency "EUR"^^xsd:string .
   ?Task csm:estimatedDuration ?estimatedDuration .
   FILTER (?estimatedDuration <= "45"^^xsd:int)
}
```

Listing 4: Example for querying tasks based on time constraints and reward expectations

6.1.3 Monitoring and Managing Crowdsourcing Tasks

The last use case scenario focuses on aspects of how the CSM ontology can be applied to monitor, control, and manage the crowdsourcing tasks assembled in a workflow. In order to check if the desired outcome of the crowdsourcing activity will be achieved, requesters need assistance in tracking the current progress of the crowdsourcing project. Example 5 shows a query for identifying the number of participants who carried out a software test on an Android mobile device.

Example 5: How many and which participants conducted a software test on an Android operating system? (query applied to sample data 2)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX csm: <http://purl.org/csm/1.0#>
SELECT ?Participant
WHERE {
   <http://example.org/#project> csm:includes ?Task .
   ?Task csm:hasActionType csm:Test .
   ?Task csm:imposes ?Requirement .
   ?Requirement rdf:type csm:TechnicalRequirement .
   ?Requirement csm:operatingSystem "Android"@en .
   ?Task csm:owns ?Contribution .
   ?Participant csm:submits ?Contribution
}
```

Listing 5: Example for querying the number of contributors for a particular task

Likewise, requesters want to discover and eliminate weak spots or bottlenecks in the overall crowdsourcing process. The last example locates tasks that have not gained the necessary attention since none of the users has contributed toward a solution. To engage more participants, these tasks might require adjustments of the task specification, the reward scheme, or the evaluation mechanism.

Example 6: "Which tasks can not be processed because their preliminary tasks lack a contribution?" (query applied to sample data 2)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX csm: <http://purl.org/csm/1.0#>
SELECT ?PreliminaryTask ?SubsequentTask ?Contribution
WHERE {
   ?PreliminaryTask csm:hasNext ?SubsequentTask .
   FILTER NOT EXISTS { ?PreliminaryTask csm:owns ?Contribution }
}
```

Listing 6: Example for querying preliminary tasks without contributions

6.2 COMPARING THE ONTOLOGY TO OTHER SEMANTIC VOCABULARIES

Researchers and practitioners have produced a wide range of semantic vocabularies for different purposes. In section 5.3, the author examined several potential candidates that might be adopted or reused in the domain of enterprise crowdsourcing. It could be shown that some of them are capable to fulfill certain single aspects of the requirements with regard to the representation of enterprise crowdsourcing data. Apparently, none of them covers the whole set of functional requirements stated in section 4.2.1, and thus may only be applied to the domain of enterprise crowdsourcing with significant modification. This is not surprising as these vocabularies are either developed for a different application domain or they are too general to meet the specific requirements of enterprise crowdsourcing.

To get a better judgment of how the designed ontology will perform in reality compared to other semantic vocabularies by means of meeting the functional requirements, each vocabulary is ranked against a set of aspects worth considering for a certain requirement (Section 4.2.1). Although this evaluation approach offers only a quantitative subjective estimate and makes no statement on the actual performance of the vocabulary, it depicts the chances of future success in an enterprise crowdsourcing environment.

Figure 7 illustrates a summary of the comparison results whereas the number of dots within each rectangle represents the expressive power in terms of relevant semantic entities (no dot = no relevant semantic entities exist, 1 dot = very few relevant semantic entities exist, 2 dots = few relevant semantic entities exist, 3 dots = many relevant semantic entities exist, 4 dots = very many relevant semantic entities exist). A detailed overview about the relevant semantic entities can be found in Appendix D. The number within a table cell points to a Semantic Web standard that is reused, for instance, the SIOC ontology reuses the FOAF vocabulary to describe additional information about the creator of a post (see row F-02.a, column 12). The last column represents the evaluation of the CSM ontology.

Functional Requirement		**Aspect**		1 Activity Streams	2 CAM	3 DOAP	4 DC	5 FOAF	6 GR	7 hCalendar/h-event	8 hCard/h-card, vCard	9 Open Social	10 PROV	11 schema.org	12 SIOC	13 WS-BPEL	14 XPDL	15 XFN	16 CSM
F-01	Task specification	F-01.a	Task description (goal, instruction, description, action type)	●●	●	●		●	●	●		●●	●	●●		●	●●		●●●
		F-01.b	Incentive mechanism						●										●●●
		F-01.c	Evaluation mechanism																●●●
		F-01.d	Human requirement																●●
		F-01.e	Technical requirement			●●								●					●●
		F-01.f	Contribution and external resources	●●	●●	●	●●●	●				●●	4	●●					4
F-02	Task allocation	F-02.a	User description	●	●	5	●	●●	●●		●●●●	●●●	5	●●●●	5		●		5, 11
		F-02.b	Suitability (interest, skills, experience, knowledge)					●●			●			●●					5, 11
		F-02.c	Availability (time and place)	●					●●		●●	●●		●●					5, 11
F-03	Team building	F-03.a	Social network (friendship, work relationship, membership)	●●	15	5		●●	11		●	●	5	●●	5			●●●	5, 11
		F-03.b	Online community (social interaction, communication, activity streams)	●●	●●							●●	●●		●●●				12
F-04	Transaction transparency and quality control	F-04.a	Access rights (permission, status)				●								●●		●●		●
		F-04.b	Sequence description	●												●●●	●●●		●●

Figure 7: Comparison of the CSM ontology with other semantic vocabularies

Several conclusions can be drawn from the evaluation. First, the comparison reveals that also some of the vocabularies offer semantic entities to make simple task descriptions, yet none of them has the expressive power to define detailed incentive schemes, evaluation mechanisms, or specific human and technical requirements of a crowdsourcing task. This may be due to the fact that vocabularies, such as Activity Streams or Open Social, are mainly designed to record past social activities rather than to announce and advertise future crowdsourcing activities. Thus, the CSM ontology provides classes and properties for defining crowdsourcing tasks. Second, the analysis indicates that user profiles and social networks are already well represented by one or more vocabularies. As a consequence, FOAF and schema.org are included in the CSM ontology to describe crowdsourcing users and their social networks. SIOC and DC are not implemented in the CSM ontology yet, but are promising candidates for representing social interactions and describing contributions in an enterprise crowdsourcing environment. Third, standardized formats for specifying and interchanging business processes, such as XPDL and WS-BPEL, provide a language to model very complex process logics and event handling mechanisms. Future research may investigate the potentials of handling complex processes as well as considering different roles and input conditions.

6.3 CALCULATING ONTOLOGY METRICS

Ontology metrics offer a quick and straightforward way to assess ontologies during their ontology engineering process and their subsequent evolution (Vrandečić & Sure, 2007). In this technical report, ontology metrics are only briefly considered to describe the general characteristic and structure of the CSM ontology. Due to not having a representative set of instances yet, the content of the ontology is evaluated using three schema metrics (Tartir, Arpinar, Moore, Sheth, & Aleman-Meza, 2005). For measuring the ontology, the relationship richness, the attribute richness, and the inheritance richness of the ontology are calculated (Table 13).

Table 13: Ontology metrics (relationship, attribute, and inheritance richness)

Metric	Formula	Description	Value
Relationship Richness	$RR = \frac{\lvert P \rvert}{\lvert H \rvert + \lvert P \rvert}$	The relationship richness (RR), also named as the relationship diversity, represents the ratio between the number of non-inheritance relationships (P) and the sum of all inheritance relationships (H) and non-inheritance relationships. The CSM ontology tends to have a high diversity of relations other than class-subclass relations.	RR=0.79
Attribute Richness	$AR = \frac{\lvert att \rvert}{\lvert C \rvert}$	The attribute richness (AR) is defined as the average number of attributes (att) per class (C). The result indicates that the CSM ontology has a reasonable but not an excessive amount of knowledge about the classes.	AR=1.25
Inheritance Richness	$IR = \frac{\lvert H \rvert}{\lvert C \rvert}$	The inheritance richness (IR) or schema depth is defined as the average number of subclasses per class. The result implies that the CSM ontology covers a specific domain in a detailed manner.	IR=0.25

Although these metrics provide rather less information about the quality of an ontology and should therefore be carefully interpreted, they indicate at least the potential of an ontology for knowledge representation in the application domain (García, García-Peñalvo, & Therón, 2010). Additionally, these metrics can serve as an orientation value or benchmark to compare the current version of the CSM ontology with future changes and advancements. However, the metrics are subject to criticism due to the fact that there are no appropriate reference values for an ideal ontology (Sicilia, Rodríguez, García-Barriocanal, & Sánchez-Alonso, 2012).

7 CONCLUSION

This technical report provides researchers and practitioners who are interested in deploying Semantic Web technologies to enhance automation and interoperability in enterprise crowdsourcing environments. The main research objective was to develop a lightweight and extensible ontology for capturing, storing, utilizing, and sharing crowdsourcing data. As a methodology to guide the overall design process, ontology engineering was chosen.

Research contributions were made on three different layers of abstraction. At the conceptual layer, a preliminary set of shared semantic entities was derived from a review of existing crowdsourcing systems and a review of recent crowdsourcing literature. This kind of consensus building approach supports the identification of well-accepted semantic entities. Thereupon, at the intermediate logical layer, a data dictionary and a corresponding data schema were built. Finally, at the physical layer, the CSM ontology was implemented in OWL.

A set of different methods was carried out to evaluate the CSM ontology. It included the transformation of informal to formal competency questions, the comparison to other semantic vocabularies, and the calculation of ontology metrics. The development of the ontology constituted a proof of concept, which demonstrated the feasibility via a functioning prototype. Through the evaluation, evidence was shown that the adoption of Semantic Web technologies promises to enhance the automation and interoperability in enterprise crowdsourcing environments. In the long run, however, a proof of demonstration in terms of successful adoption in a real-life context is required. This calls for evaluating the CSM ontology not only based on the schema but also on the instances (populated ontologies).

Recommendations for future efforts could be given in several directions. One step to demonstrate and unleash the full potentials of the ontology is the implementation of CSM metadata exports from existing web-based enterprise crowdsourcing systems and business applications. For example, a CSM wrapper could be developed that transforms the crowdsourcing data residing in closed database environments into RDF triples. Another step is to provide query facilities. This necessitates replicating the structured crowdsourcing data in a data repository or native RDF data store, which can handle the queries. To replicate the data, either a web crawler automatically updates the RDF data store or the crowdsourcing application itself pushes changes after the crowdsourcing data has been created or modified.

Even though the CSM ontology covers the key representational needs of the enterprise crowdsourcing domain, some challenges are worth highlighting. One of the main challenges is how to reach a wide adoption of the CSM ontology in practice and how to create an incentive for people to publish crowdsourcing data and to develop applications for the proposed CSM ontology. This also asks for engaging more researchers and practitioners in future improvement cycles of the ontology that leads into a standardization process. Another challenge that will also impact the future design and adoption of the ontology is the issue of privacy and trust of crowdsourcing data that is available and viewable to the public, such as the participants' personal data or extremely business-critical information of the company. Thus, solutions are required that allow for the exchange of sensitive information not only in an open manner, but also in a closed network of trusted crowd workers.

REFERENCES

Adepetu, A., Ahmed, K., & Abd, Y. Al. (2012). CrowdREquire: a requirements engineering crowdsourcing platform. In *AAAI Spring Symposium Series*.

Adida, B., Birbeck, M., McCarron, S., & Herman, I. (2013). RDFa Core 1.1 – Second Edition: syntax and processing rules for embedding RDF through attributes (W3C Recommendation 22 August 2013). World Wide Web Consortium (W3C). Retrieved from http://www.w3.org/TR/2013/REC-rdfa-core-20130822/

Brabham, D. C. (2013). *Crowdsourcing*. Cambridge, Mass.: MIT Press.

Brank, J., Grobelnik, M., & Mladenić, D. (2005). A survey of ontology evaluation techniques. In *Proceedings of the 8th International Multi Conference Information Society*.

Casellas, N. N. (2011). Methodologies, tools and languages for ontology design. In *Legal Ontology Engineering* (Vol. 3, pp. 57–107). Springer Netherlands. doi:10.1007/978-94-007-1497-7_3

Chandler, D., & Kapelner, A. (2013). Breaking monotony with meaning: motivation in crowdsourcing markets. *Journal of Economic Behavior & Organization*, *90*, 123–133. doi:10.1016/j.jebo.2013.03.003

Corney, J. R., Torres-Sanchez, C., Jagadeesan, A. P., & Regli, W. C. (2009). Outsourcing labour to the cloud. *International Journal of Innovation and Sustainable Development*, *4*(4), 294–313.

Corney, J. R., Torres-Sánchez, C., Jagadeesan, A. P., Yan, X. T., Regli, W. C., & Medellin, H. (2010). Putting the crowd to work in a knowledge-based factory. *Advanced Engineering Informatics*, *24*(3), 243–250. doi:10.1016/j.aei.2010.05.011

Cosley, D., Frankowski, D., Terveen, L., & Riedl, J. (2007). SuggestBot: using intelligent task routing to help people find work in Wikipedia. In *Proceedings of the 12th International Conference on Intelligent User Interfaces* (pp. 32–41). New York, NY, USA: ACM. doi:10.1145/1216295.1216309

Dai, P., Mausam, & Weld, D. S. (2010). Decision-theoretic control of crowd-sourced workflows. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence* (pp. 1168–1174).

Della Valle, E., & Ceri, S. (2011). Querying the Semantic Web: SPARQL. In J. Domingue, D. Fensel, & J. A. Hendler (Eds.), *Handbook of Semantic Web Technologies* (pp. 299–363). Berlin: Springer.

Devedzić, V. (2002). Understanding ontological engineering. *Communications of the ACM*, *45*(4), 136–144. doi:10.1145/505248.506002

Doan, A., Ramakrishnan, R., & Halevy, A. Y. (2011). Crowdsourcing systems on the World-Wide Web. *Communications of the ACM*, *54*(4), 86–96. doi:10.1145/1924421.1924442

Erickson, L. B. (2012). Leveraging the crowd as a source of innovation: does crowdsourcing represent a new model for product and service innovation? In *Proceedings of the 50th Annual Conference on Computers and People Research* (pp. 91–96). New York, NY, USA: ACM. doi:10.1145/2214091.2214117

Erickson, L. B., Petrick, I., & Trauth, E. M. (2012a). Hanging with the right crowd: matching crowdsourcing need to crowd characteristics. In *Proceedings of the 18th Americas Conference on Information Systems (AMCIS)* (pp. 1–9).

Erickson, L. B., Petrick, I., & Trauth, E. M. (2012b). Organizational uses of the crowd: developing a framework for the study of crowdsourcing. In *Proceedings of the 50th Annual Conference on Computers and People Research* (pp. 155–158). New York, NY, USA: ACM. doi:10.1145/2214091.2214133

Erickson, T. (2011). Some thoughts on a framework for crowdsourcing. In *Workshop on Crowdsourcing and Human Computation* (pp. 1–4).

Estellés-Arolas, E., & González-Ladrón-de-Guevara, F. (2012). Towards an integrated crowdsourcing definition. *Journal of Information Science*, *38*(2), 189–200. doi:10.1177/0165551512437638

García, J., García-Peñalvo, F., & Therón, R. (2010). A Survey on ontology metrics. In M. Lytras, P. Ordonez De Pablos, A. Ziderman, A. Roulstone, H. Maurer, & J. Imber (Eds.), *Knowledge Management, Information Systems, E-Learning, and Sustainability Research SE - 4* (Vol. 111, pp. 22–27). Springer Berlin Heidelberg. doi:10.1007/978-3-642-16318-0_4

Gassmann, O. (2010). *Crowdsourcing: Innovationsmanagement mit Schwarmintelligenz*. Hanser.

Geiger, D., Fielt, E., Rosemann, M., & Schader, M. (2012). Crowsourcing information systems – definition, typology, and design. In *Proceedings of the 33rd International Conference on Information Systems* (pp. 1–11). Orlando.

Geiger, D., Rosemann, M., & Fielt, E. (2011). Crowdsourcing information systems – a systems theory perspective. In *Proceedings of the 22nd Australasian Conference on Information systems*. Sydney.

Geiger, D., Seedorf, S., Schulze, T., Nickerson, R., & Schader, M. (2011). Managing the crowd: towards a taxonomy of crowdsourcing processes. In *Proceedings of the 17th Americas Conference on Information Systems* (pp. 1–11).

Gómez-Pérez, A. (1996). Towards a framework to verify knowledge sharing technology. *Expert Systems with Applications*, *11*(4), 519–529. doi:10.1016/S0957-4174(96)00067-X

Gómez-Pérez, A., Fernández, M., & de Vicente, A. J. (1996). Towards a method to conceptualize domain ontologies. In *Proceedings of the 12th European Conference on Artificial Intelligence (ECAI-96), Workshop on Ontological Engineering* (pp. 41–52).

Gómez-Pérez, A., Fernández-López, M., & Corcho, O. (2005). *Ontological engineering – with examples from the areas of knowledge management, e-commerce and the Semantic Web* (4th ed.). London, Berlin, Heidelberg: Springer.

Grimm, S., Abecker, A., Völker, J., & Studer, R. (2011). Ontologies and the Semantic Web. In J. Domingue, D. Fensel, & J. A. Hendler (Eds.), *Handbook of Semantic Web Technologies* (pp. 507–570). Berlin: Springer.

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies*, *43*(5–6), 907–928. doi:10.1006/ijhc.1995.1081

Gruber, T. R. (2009). Ontology. In L. Liu & M. T. Özsu (Eds.), *Encyclopedia of database systems* (pp. 1963–1965). Berlin: Springer.

Grüninger, M., & Fox, M. S. (1995). Methodology for the design and evaluation of ontologies. In *Proceedings of the International Joint Conference on AI (IJCAI-95), Workshop on Basic Ontological Issues in Knowledge Sharing* (pp. 1–10). Montreal.

Harris, S., & Seaborne, A. (2013). SPARQL 1.1 Query Language (W3C Recommendation 21 March 2013). World Wide Web Consortium (W3C). Retrieved from http://www.w3.org/TR/2013/REC-sparql11-query-20130321/

Haslhofer, B., & Klas, W. (2010). A survey of techniques for achieving metadata interoperability. *ACM Computing Surveys*, *42*(2), 1–37. doi:10.1145/1667062.1667064

Hepp, M. (2007). Possible ontologies: How reality constrains the development of relevant ontologies. *Internet Computing, IEEE*, *11*(1), 90–96. doi:10.1109/MIC.2007.20

Hepp, M. (2008). GoodRelations: An ontology for describing products and services offers on the Web. In A. Gangemi & J. Euzenat (Eds.), *Proceedings of the 16th International Conference on Knowledge Engineering: Practice and Patterns (EKAW2008)* (Vol. 5268, pp. 329–346). Acitrezza: Springer Berlin Heidelberg. doi:10.1007/978-3-540-87696-0_29

Herman, I., Adida, B., Sporny, M., & Birbeck, M. (2013). RDFa 1.1 Primer – Second Edition – Rich Structured Data Markup for Web Documents (W3C Working Group Note 07 June 2012). The World Wide Web Consortium (W3C). Retrieved from http://www.w3.org/TR/2013/NOTE-rdfa-primer-20130822/

Hetmank, L. (2013a). Components and functions of crowdsourcing systems – a systematic literature review. In *11th International Conference on Wirtschaftsinformatik* (pp. 55–69). Leipzig.

Hetmank, L. (2013b). Towards a semantic standard for enterprise crowdsourcing – a scenario-based evaluation of a conceptual prototype. In *21st European Conference on Information Systems*. Utrecht.

Hetmank, L. (2014a). A synopsis of enterprise crowdsourcing literature. In *22nd European Conference on Information Systems*. Tel Aviv.

Hetmank, L. (2014b). Developing an ontology for enterprise crowdsourcing. In *Multikonferenz Wirtschaftsinformatik 2014*. Paderborn.

Howe, J. (2008). *The rise of crowdsourcing: why the power of the crowd is driving the future of business*. New York: Three Rivers Press.

Jones, D., Bench-Capon, T., & Visser, P. (1998). Methodologies for ontology development. In *Proceedings of 15th IFIP World Computer Congress* (pp. 20–35). Budapest.

Kayed, A. (2013). Ontology evaluation: which test to use. In *Proceedings of the 5th International Conference on Computer Science and Information Technology (CSIT)* (pp. 45–48). doi:10.1109/CSIT.2013.6588756

Kearns, M. (2012). Experiments in social computation. *Communications of the ACM*, *55*(10), 56–67. doi:10.1145/2347736.2347753

Khasraghi, H. J., & Tarokh, M. J. (2012). Efficient business process reengineering with crowdsourcing. *International Journal of Applied Information Systems (IJAIS)*, *2*(7).

Khazankin, R., Satzger, B., & Dustdar, S. (2012). Optimized execution of business processes on crowdsourcing platforms. In *Proceedings of the 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing* (pp. 443–451).

Kittur, A., Chi, E. H., & Suh, B. (2008). Crowdsourcing user studies with Mechanical Turk. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (pp. 453–456). ACM.

Kittur, A., Khamkar, S., André, P., & Kraut, R. (2012). CrowdWeaver: visually managing complex crowd work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work* (pp. 1033–1036). New York, NY, USA: ACM. doi:10.1145/2145204.2145357

Kittur, A., Nickerson, J. V, Bernstein, M., Gerber, E., Shaw, A., Zimmerman, J., Lease, M. & Horton, J. (2013). The future of crowd work. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work* (pp. 1301–1318). New York, NY, USA: ACM. doi:10.1145/2441776.2441923

Kulkarni, A., Can, M., & Hartmann, B. (2012). Collaboratively crowdsourcing workflows with Turkomatic. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (pp. 1003–1012). New York, NY, USA: ACM. doi:10.1145/2145204.2145354

Law, E., & von Ahn, L. (2011). *Human Computation*. (R. J. Brachman, W. W. Cohen, & T. Dietterich, Eds.) (Vol. 5, pp. 1–121). doi:10.2200/S00371ED1V01Y201107AIM013

Liu, D., Bias, R. G., Lease, M., & Kuipers, R. (2012). Crowdsourcing for usability testing. *Proceedings of the American Society for Information Science and Technology*, *49*(1), 1–10. doi:10.1002/meet.14504901100

Lopez, M., Vukovic, M., & Laredo, J. (2010). PeopleCloud service for enterprise crowdsourcing. In *Proceedings of the International Conference on Services Computing (SCC)* (pp. 538–545). doi:10.1109/SCC.2010.74

Lykourentzou, I., Vergados, D., Papadaki, K., & Naudet, Y. (2013). Guided crowdsourcing for collective work coordination in corporate environments. In C. Bădică, N. Nguyen, & M. Brezovan (Eds.), *Proceedings of the 5th International Conference on Computational Collective Intelligence Technologies and Applications* (Vol. 8083, pp. 90–99). Berlin, Heidelberg: Springer. doi:10.1007/978-3-642-40495-5_10

Malone, T. W., Laubacher, R., & Dellarocas, C. (2010). The collective intelligence genome. *Engineering Management Review, IEEE*, *38*(3), 38–52. doi:10.1109/EMR.2010.5559142

Nielsen, M. (2011). *Reinventing discovery: the new era of networked science*. Princeton University Press.

OASIS. (2010). WS-BPEL Extension for People (BPEL4People) Specification Version 1.1. Organization for the Advancement of Structured Information Standards (OASIS). Retrieved from http://docs.oasis-open.org/bpel4people/bpel4people-1.1.html

Obrst, L., Ceusters, W., Mani, I., Ray, S., & Smith, B. (2007). The evaluation of ontologies. In C. J. O. Baker & K.-H. Cheung (Eds.), *Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences* (pp. 139–158). New York: Springer.

Quinn, A. J., & Bederson, B. B. (2011). Human computation: survey and taxonomy of a growing field. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)* (pp. 1403–1412). Vancouver.

Rahm, E., & Bernstein, P. a. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal*, *10*(4), 334–350. doi:10.1007/s007780100057

Rouse, A. C. (2010). A preliminary taxonomy of crowdsourcing. In *Proceedings of the 21st Australasian Conference on Information Systems*. Brisbane.

Satzger, B., Psaier, H., Schall, D., & Dustdar, S. (2013). Auction-based crowdsourcing supporting skill management. *Information Systems*, *38*(4), 547–560.

Saxton, G. D., Oh, O., & Kishore, R. (2013). Rules of crowdsourcing: models, issues, and systems of control. *Information Systems Management*, *30*(1), 2–20. doi:10.1080/10580530.2013.739883

Schall, D. (2012). *Service-Oriented Crowdsourcing: Architecture, Protocols and Algorithms*. New York: Springer.

Schenk, E., & Guittard, C. (2011). Towards a characterization of crowdsourcing practices. *Journal of Innovation Economics*, *7*(1), 93–107. doi:10.3917/jie.007.0093

Sicilia, M. A., Rodríguez, D., García-Barriocanal, E., & Sánchez-Alonso, S. (2012). Empirical findings on ontology metrics. *Expert Systems with Applications*, *39*(8), 6706–6711. doi:10.1016/j.eswa.2011.11.094

Skopik, F., Schall, D., & Dustdar, S. (2012). Discovering and managing social compositions in collaborative enterprise crowdsourcing systems. *International Journal of Cooperative Information Systems*, *21*(4), 297–341.

Sommerville, I. (2011). *Software Engineering* (9th ed.). Boston: Pearson.

Sporny, M. (2013). HTML+RDFa 1.1 – Support for RDFa in HTML4 and HTML5 (W3C Recommendation 22 August 2013). The World Wide Web Consortium (W3C). Retrieved from http://www.w3.org/TR/rdfa-in-html/

Sure, Y., Staab, S., & Studer, R. (2009). Ontology engineering methodology. In S. Staab & R. Studer (Eds.), *Handbook on Ontologies* (2nd ed., pp. 135–152). Berlin, Heidelberg: Springer.

Tartir, S., Arpinar, I. B., Moore, M., Sheth, A. P., & Aleman-Meza, B. (2005). OntoQA: metric-based ontology quality analysis. In *IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources* (Vol. 9).

Uschold, M., & Grüninger, M. (1996). Ontologies: principles, methods and applications. *Knowledge Engineering Review*, *11*(2).

Uschold, M., & King, M. (1995). Towards a methodology for building ontologies. In *Proceedings of the International Joint Conference on AI (IJCAI-95), Workshop on Basic Ontological Issues in Knowledge Sharing*. Budapest.

Vrandečić, D. (2009). Ontology evaluation. In S. Staab & R. Studer (Eds.), *Handbook on Ontologies* (2nd ed., pp. 293–313). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-92673-3

Vrandečić, D., & Sure, Y. (2007). How to design better ontology metrics. In E. Franconi, M. Kifer, & W. May (Eds.), *Proceedings of the 4th European Semantic Web Conference (ESCW 2007)* (Vol. 4519, pp. 311–325). Berlin, Heidelberg: Springer. doi:10.1007/978-3-540-72667-8_23

Vukovic, M. (2009). Crowdsourcing for enterprises. In *Proceedings of the 2009 Congress on Services - I* (pp. 686–692). Los Angeles: IEEE. doi:10.1109/SERVICES-I.2009.56

Vukovic, M., Laredo, J., & Rajagopal, S. (2010). Challenges and experiences in deploying enterprise crowdsourcing service. In B. Benatallah, F. Casati, G. Kappel, & G. Rossi (Eds.), *Proceedings of the 10th International Conference on Web Engineering* (Vol. 6189, pp. 460–467). Berlin, Heidelberg: Springer.

WfMC. (2012). Process Definition Interface – XML Process Definition Language (XPDL) Version 2.2. The Workflow Management Coalition.

Ye, H., & Kankanhalli, A. (2013). Leveraging crowdsourcing for organizational value co-creation. *Communications of the Association for Information Systems*, *33*(13), 225–244.

Yuen, M.-C., King, I., & Leung, K.-S. (2011). A survey of crowdsourcing systems. In *Proceedings of the International Conference on Privacy, Security, Risk, and Trust, and*

IEEE International Conference on Social Computing (pp. 766–773). doi:10.1109/PASSAT/SocialCom.2011.203

Zhao, Y., & Zhu, Q. (2012). Evaluation on crowdsourcing research: current status and future direction. *Information Systems Frontiers*, 1–18.

Zwass, V. (2010). Co-Creation: toward a taxonomy and an integrated research perspective. *International Journal of Electronic Commerce*, *15*(1), 11–48. doi:10.2753/JEC1086-4415150101

APPENDIX A (SYSTEM REVIEW)

Platform	Task properties							User properties (requester and participant)
	Task specification (F-01)				Task allocation (F-02)		Transaction transparency and quality control (F-04)	
	Task description	Time and priority	Reward	Evaluation	Requester-oriented	Participant-oriented		
Amazon mTurk	project name, task title, task description, keywords, task type (categorize, collect data, moderate, get sentiment, survey, tag, transcribe, create content), instructions	duration, expiration, approval time after completion,	reward per assignment	-	qualification type, approval rate, number of approved tasks	creation date, task available, reward amount, expiration date, duration	number of assignments per task, status (in progress, for review, reviewed)	name, login name, contact address information, prepaid balance
Atizio	title, description, image, additional information (text, document), important information, acceptance criteria, thank-you text, visibility	duration (start and end date/time)	amount of (alternative) reward,	-	-	reward, accepted languages (de, fr, en), duration	user activity (ideas, projects, comments, comment evaluation, idea evaluation, time of membership)	first name, last name, address (street, zip code, city, country), age, about me, website, interests, profession, job status, educational level, languages, references, career/CV, contact list
crowdSPRING	project title, project description, external resources	end date	amount of payment	-	specialization, country, language	product category, activity score, award, time, contributions, status	user activity (reputation score, projects, awarded projects)	first name, last name, about me, address (city, state, postal code, country), language, time zone, specialization, profile image, email, portfolio items
CrowdWorx	prediction name, description, survey configuration, detailed information, additional information, diagrams, tables, references	-	-	-	-	-	-	user name, name, position, department, telephone, fax, about me
designenlassen	project name, company description, task description, additional information	duration	amount of payment	-	-	time left, number of proposals, reward, buyer, project language, project category, project type	user activity (projects won, projects involved, success rate)	first name, last name, company name, address (street, zip code, city, country, telephone, email, user type (designer, employer), tax id, value-added tax id
elance	task name, task description, external resources, category, subcategory, skills, task visibility (public, private)	workload per week, validity, start date (immediate, date)	type of payment (hourly, fixed), amount of payment	-	type of agent (individual, company), category, preferred location, skills, repu-	categories, type of payment, amount of payment, workload, location (region,	-	user name, first name, last name, address, email, telephone, company name, job title, description, video profile, time zone, team member, social

Platform	Task properties							User properties (requester and participant)
	Task specification (F-01)				Task allocation (F-02)		Transaction transparency and quality control (F-04)	
	Task description	Time and priority	Reward	Evaluation	Requester-oriented	Participant-oriented		
					tation	country), time left		network, portfolio, skills, overview description, service description, payment terms, certifications, licenses, employment, education, references, keywords, group membership
Gengo	project title, translation text, instructions, original and target language	-	-	quality level, preferred translators	-	-	-	address (full name, address line, town/city, state/prefecture, zip/postal code, country, telephone), time zone, display name, email, password, additional information, taxpayer status
Innocentive	project title, brief description, detailed description, image, type of challenge (internal, invitational, external), management of intellectual property	-	award amount	-	expertise, interests, country, type of user (has submitted solutions, winning solver)	expertise, interests	-	first name, last, name, email, address (city state/province, country, zip/postal code, phone number, fax number), academic degree (Baccalaureate, Master, Ph.D., Post doctoral, High school, others), work status (self-employed, independent consultant, small company employee, corporate employee, government employee, non-profit employee, professor/educator, unemployed, retired, undergraduate student, graduate student), employer, contract service (research, manufacturing, product development), expertise, interests, personal headline, biography, link to social network or website LinkedIn, Facebook, Twitter, Website), education list, publication list
MobileWorks	project name, instructions, fields (text, multiple choice, email, telephone, date, number, comma separated	priority	payment	pre-test tasks	blocked worker, language, location, minimum age, maximum	-	workflow (iterative, parallel, survey, manual), redundancy	user name, first name, last name, email, account balance, country, native language, number of completed tasks, accura-

Platform	Task properties							User properties (requester and participant)
	Task specification (F-01)				Task allocation (F-02)		Transaction transparency and quality control (F-04)	
	Task description	Time and priority	Reward	Evaluation	Requester-oriented	Participant-oriented		
	values), link to resource, resource type (text, image, link, audio)				age, gender		(number of redundant workers), user activity (task completed, accuracy, earnings, rank)	cy, earnings, rank
oDesk	task category, task sub category, task description, contractor type (independent, agency), marketplace visibility (all, internal, invited), link to external resource	duration, workload	type of payment (hourly, fixed), amount of payment	pre-evaluation	category, skills required, type of payment, contractor type (none, independent, agency), marketplace visibility, feedback score, hourly rate, hours billed, last activity, location, English level, test score of skill	hourly rate, minimum feedback score, participation, location, English level	status (open, filled, closed)	profile access, job title, years of experience, English level, overview, video profile, individual skills, skill category, employment history, education, portfolio projects, certifications, experiences
Seedmatch	project name, website, description, funding threshold, funding limit, discount	end date	-	-	-	-	status (open, closed)	first name, last name, gender, academic title, email, address (street, zip code, city, country, area code, telephone number, birthday, tax id), profile image, link to social network (Facebook, Twitter, Xing), about me
Startnext	project title, short name, category, address, funding goal, keywords, detailed information (about, goal, motivation, investment decision, responsible persons), image	end date	-	-	category, keywords	recommended projects, new projects, most supported projects, expiration date, location	project status (created, started, deleted, feedback)	first name, last name, gender, academic title, display name, company name, profile image, about me, link to social network or website (website, twitter, Facebook, MySpace), birthday, address (street, city, zip code, country)
TopCoder	project name, project description, contest introduction, contest description, round information, billing account, project type,	project duration, task priority, start date, end date, checkpoint duration	project budget, contest prizes for each place, checkpoint prizes (for each	review style (user selection)	-	payment, bonus, number of submissions	project status (draft, active, on hold, cancelled, completed), accessibility, task	name, company, address, country, time zone, phone number, email, photo

Platform	Task properties							User properties (requester and participant)
	Task specification (F-01)				Task allocation (F-02)		Transaction transparency and quality control (F-04)	
	Task description	Time and priority	Reward	Evaluation	Requester-oriented	Participant-oriented		
	project category, SVN address, bug tracker address, project rating (business impact, risk level, cost, difficulty), external resources (url, category, description, access restriction), confidentiality agreement, visibility		submission, up to number of submission)				status (not started, in progress, waiting on dependency, completed), contest round type (single round, multiple rounds)	
unserAller	project name, project title, question, description, image	end date (continuously or terminated)	bonus type (discount in percentage, coupon, lottery, buy one get one free), bonus criteria (top10, top20, top50, top100, all, winners), bonus detail, bonus code	-	-	interests, location	-	first name, last name, email, gender, address (street, zip code, city), company name, industry sector, short name, contact person, email, webpage, link to imprint, profile image, link to social network (Facebook)
uTest	project name, project description, scope, out of scope	start date, end date	-	-	-	product type, location, language, audience (consumer, business), industry	-	first name, last name, email, country, city, postal code, phone number, native language, birth year, gender, profile image, resume file, about me, testing expertise (usability, language, industry, hobby, hardware, software), link to social network (Twitter, Facebook, LinkedIn, Google+), users' availability (hours, dates), testing experience (in years)
ziptask	project title, instructions	start date end date, urgent (yes/no)	budget range	-	required skills	-	status (open, closed, archived)	email, username, first name, last name, profile image, mobile number, account name, account type (business, non-profit, team, household, school, other)

APPENDIX B (CROWDSOURCING TAXONOMIES)

Article	Research method	Dimension (characteristic)	Component (function)	Semantic entity
Crowdsourcing systems on the World-Wide Web (Doan et al., 2011)	Based on an empirical analysis	• Nature of collaboration (implicit, explicit)	User management (enable coordination)	Interaction mode
		• Type of target problem (evaluate, share, network, build artifacts, execute tasks)	Task management (design task)	Type of action
		• Design of incentive mechanism (by authority, pay users, ask for volunteers, make users pay for service, piggyback, instant gratification, gamification, reputation, competition, ownership situations)	Task management (design task)	Reward and incentive mechanism (nature of the reward)
		• Task complexity (simple, cognitively complex)	Task management (design task)	Complexity level
		• Impact of contribution (low, high)	Task management	Impact level
		• Approach to combine solutions (none, manual, automatic)	Workflow management (define workflow)	Type of aggregation (with respect to automation)
		• Method to evaluate users (block, detect, punish)	User management (evaluate user)	Evaluation mechanism
		• Degree and distribution of manual effort (manual: user or system owner, automatic)	Workflow management (define and manage workflow)	Type of aggregation, evaluation mechanism (source)
		• Role of human users (slaves, perspective providers, content providers, component providers)	Task management (design task) User management (form user group)	Human requirement
		• Type of architecture (standalone, piggyback)	Task management (design task)	Technical requirement
Crowdsourcing Information Systems – A Systems Theory Perspective (Geiger et al., 2011); Crowdsourcing Information Systems - Definition, Typology, and Design (Geiger et al., 2012)	Based on a system-theoretical approach	• Treatment of external elements (homogenous, heterogeneous)	Task management (design task) User management (form user group)	Human requirements (demographic characteristics, qualification type and level)
		• Value of the relationship with an external element (individual, collective)	Workflow management (define workflow)	Type of aggregation
Some Thoughts on a Framework for Crowdsourcing (Erickson, 2011)	Based on literature of computer supported cooperative work (CSCW)	• Distribution over time (same time, different times)	Task management (design task)	Time, latency
		• Distribution over space (same place, different places)	Task management (design task)	Location
A Survey of Crowdsourcing Systems (Yuen et al., 2011)	Based on a literature review	• Application (vote, share, play, create)	Task management (design task)	Type of action

Article	Research method	Dimension (characteristic)	Component (function)	Semantic entity
A Preliminary Taxonomy of Crowdsourcing (Rouse, 2010)	Based on a review of largely non-academic publications	• Nature of the crowdsourcing task / supplier capabilities (simple, sophisticated, moderate)	Task management (design task)	Complexity level
		• Distribution of benefits (individualistic, community, mixed)	Task management (design task)	Reward and incentive mechanism
		• Nature of the motivation to participate (self-marketing, social status, instrumental, altruism, token compensation, market compensation, personal achievement and learning)	Task management (design task)	Reward and incentive mechanism
Towards a characterization of crowdsourcing practices (Schenk & Guittard, 2011)	Based on different cases of crowdsourcing	• Nature of the crowdsourcing process (integrative, selective)	Workflow management (define workflow)	Type of aggregation
		• Type of task (simple, complex, creative)	Task management (define task)	Complexity level
Outsourcing labor to the cloud (Corney et al., 2009)	Based on current applications, platforms, and academic literature	• Nature of the task (creation, evaluation, organization)	Task management (design task)	Type of action
		• Nature of the crowd (any individual, most people, or expert)	Task management (design task) User management (form user group)	Human requirement (qualification type and level)
		• Nature of the payment (voluntary, rewarded at a flat rate, rewarded with a bonus or prize)	Task management (design task)	Reward mechanism (nature of the reward)
Managing the Crowd: Towards a Taxonomy of Crowdsourcing Processes (Geiger et al., 2011)	Based on dimensions used in existing crowdsourcing literature and insights gained by applying these dimensions on real application	• Pre-selection (qualification-based, context-specific, both, none)	User management (evaluate user)	Evaluation mechanism, Target audience, Human Requirements
		• Accessibility (modify, assess, view, none)	Workflow management (define workflow)	Visibility, Accessibility
		• Aggregation (integrative, selective)	Workflow management (define workflow)	Type of aggregation
		• Remuneration (fixed, success-based, none)	Task management (design task)	Reward and incentive mechanism (nature of the reward)
Hanging with the right crowd: Matching crowdsourcing need to crowd characteristics (Erickson et al., 2012a); Organizational uses of the crowd: developing a framework for the study of crowdsourcing (Erickson et al., 2012b); Leveraging the crowd as a source of innovation: does crowdsourcing represent a new model for product and service innovation?	Based on a literature review and grounded theory	• Organizational uses of the crowd (marketing/branding, productivity, product/service innovation, and knowledge capture)	Task management (design task)	Category, goal
		• Common task (ideation, filtration, evaluation, design, development, complex problem solving, tasks difficult for computers but easy for humans, data collection, knowledge sharing)	Task management (design task)	Type of action
		• Crowd knowledge (general, situational, product/service, specialized, domain expertise, problem solving)	Task management (design task) User man-	Human requirement (qualification type and level)

Article	Research method	Dimension (characteristic)	Component (function)	Semantic entity
(Erickson, 2012)			agement (form user group)	
		• Value of the crowd (diversity, distributed knowledge, large numbers)	Task management (design task) User management (form user group)	Human requirement (diversity of qualifications, number of contributors)
		• Preferred crowd location (internal, external)	Task management (design task) User management (form user group)	Target audience
Human computation: a survey and taxonomy of a growing field (Quinn & Bederson, 2011)	Based on a review of human computation literature and examples found in industry	• Motivation (pay, altruism, enjoyment, reputation, implicit work)	Task management (design task)	Reward and incentive mechanism (type of the reward)
		• Quality control (output agreement, input agreement, economic models, defensive task design, redundancy, statistical filtering, multilevel review, automatic check, reputation system)	Contribution management (evaluate contribution)	Evaluation mechanism (evaluation method)
		• Aggregation (collection, wisdom of crowds, search, iterative improvement, genetic algorithm, node)	Workflow management (define workflow)	Type of aggregation
		• Human skill (visual recognition, language understanding, basic human communication)	Task management (design task) User management (form user group)	Human requirement (qualification type and level)
		• Process order (computer-worker-requester, worker-requester-computer, computer-worker-requester-computer, requester-worker)	Workflow management (define workflow)	Sequence of work
		• Task-request cardinality (one-to-one, many-to-many, many-to-one, few-to-one)	Workflow management (define workflow)	Number of assignments
The collective intelligence genome (Malone et al., 2010)	Based on examples of web-enabled collective intelligence	• Goal (create, decide)	Task management (define task)	Type of action
		• Staffing (hierarchy, crowd)	Workflow management (define workflow)	Interaction mode
		• Incentives (money, love, glory)	Task management (design task)	Reward and incentive mechanism (type of the reward)
		• Process (create: collection, contest, collaboration; group: decision: voting, averaging, consensus, prediction market; individual decisions: market, social network)	Workflow management (define workflow)	Type of aggregation
Co-Creation: Toward a Taxonomy and an Integrated Research Perspective (Zwass, 2010)	Based on a literature review	• Performers (world, prequalified individuals, community members, skilled contributors)	Task management (design task) User management	Target audience, Human requirements (qualification type and level), evaluation mecha-

Article	Research method	Dimension (characteristic)	Component (function)	Semantic entity
			(evaluate user, form user group)	nism
		• Motivation (altruistic, monetary)	Task management (design task)	Reward and incentive mechanism (type of the reward)
		• Structural task complexity (high, low)	Task management (design task)	Complexity level
		• Intellective demands (high, low)	Task management (design task)	Human requirement (qualification type and level)
		• Effort intensity (high, low)	Task management (design task)	Estimated time of duration
		• Time frame (indefinite, tight)	Task management (design task)	Submission and closure time, latency
		• Aggregation (searchable corpus, hyperlinking, statistical ratings and rankings, competition and voting, information markets, bottom-up taxonomy, moderators)	Workflow management (define workflow)	Type of aggregation

APPENDIX C (DATA DICTIONARY)

Concept	Description
User	the person or organization that is involved in a crowdsourcing activity
Requester	the initiator of a crowdsourcing project (any individual, company, or public organization)
Participant	the person that submits a contribution
Project	a carefully planned crowdsourcing activity that includes one or more tasks
Task	the smallest indivisible unit of work that is clearly described by a single instruction
Target Audience	an initial general restriction of the size of the crowd (internal, external, both)
Action Type	the type of action that is required to solve a crowdsourcing task
Aggregation Type	the mechanism of how the contributions of the crowd are combined
Latency Type	the time when the solution of a crowdsourcing task can be expected (immediate, untimed)
External Resource	all additional inputs that are required to accomplish a crowdsourcing task (e.g., applications, documents, or datasets)
Contribution	all data records, content items, documents, or code fragments that are part of the solution of a crowdsourcing task
Requirement	any human or technical aspect that is required to solve a crowdsourcing task
Human Requirement	any aspect that supports the composition of the crowd (demographic characteristics, qualification, etc.)
Technical Requirement	any aspect that specifies the required system of a crowd member
Reward Mechanism	the configuration of a certain type of a reward mechanism
Reward Nature	the definition of how a worker's contribution is rewarded (fixed, performance-based, proportional, voluntary)
Reward Type	the specification of what kind of reward is offered (coupon, discount, good, lottery, money, no reward, resource access, virtual points)
Payout Method	the specification of which participants are rewarded (all, the winner only, or the top ten participants)
Evaluation Mechanism	the configuration that contains aspects of how a crowdsourcing user or contribution is evaluated
Evaluation Time	the point of time when the participants or the contributions are evaluated (after, before, simultaneously)
Evaluation Source	the agent that is engaged in the evaluation process (requester, participant, third party organization, algorithm)
Evaluation Method	the method that is used for the evaluation (majority decision, control group)
Evaluation Specificity	the specificity of the evaluation (an acceptance or rejection, a rating, an assessment form, or a free text response)
Evaluation Subject	the subject of the evaluation (contribution, participant)

Attribute	Description
title	a short phrase to describe the instance
description	one or more sentences to describe the instance
goal	the desired result of a crowdsourcing activity
category	the class of a crowdsourcing activity (e.g., idea generation, problem-solving, or content creation)
instruction	a direction that lead the crowd towards a common goal
submission time	the time when the task is accessible for the crowd
closure time	the time when the task expires
confidentiality level	a value that limits the access to certain types of information
complexity level	the amount of skills, experiences, and knowledge that is required to solve a crowdsourcing task
priority level	the importance of a crowdsourcing task compared to other concurrent tasks
visibility mode	a specification of whether the workers can or cannot see each other's contributions
interaction mode	the configuration that enables or disables peer-to-peer collaboration
estimated duration	the estimated time required to complete a crowdsourcing task
number of assignments	the number of participants who will be assigned to one crowdsourcing task
amount	the number of points or the monetary value of the reward
currency	the currency of the monetary reward
gender	the gender that a participant should have

Attribute	Description
minimum age (minAge)	the minimum age that a participant should have
maximum age (maxAge)	the maximum age that a participant should have
language	the language that a participant should speak
country	the country in which a participant should live
job title	the job title that a participant should have
qualification type	the academic degree, certificate, or skill that is required to solve a crowdsourcing task
qualification level	the proficiency level that is required for a certain qualification
approval rate	the ratio of properly solved tasks to the number of submitted tasks
number of approved tasks	the number of properly solved tasks
hardware device	any hardware device (e.g., personal computer, mobile phone, smartphone, tablet computer)
hardware feature	any build-in feature of the hardware device (e.g., processor, memory, GPS, camera, accelerometer, gyrometer)
operating system	any operating system for personal computers or mobile devices (e.g., Microsoft Windows, OS X, Linux, Android, Windows Phone, iOS)
software platform	any operating system independent platform (e.g., Java, Firefox)

Relation	Description
issues (requester, project)	the project that a requester issues
submits (participant, contribution)	the contribution that a participant submits
includes (project, task)	the crowdsourcing tasks that are included in a project
has next (task, task)	the subsequent crowdsourcing task
has previous (task, task)	the preliminary crowdsourcing task
owns (task, contribution)	the contributions that belong to a crowdsourcing task

APPENDIX D (SEMANTIC VOCABULARIES)

Functional requirement		Aspect		Activity Streams	CAM	DOAP	DC	FOAF	GR	hCalendar/ h-event	hCard/ h-card, vCard
F-01	Task specification	F-01.a	Task description (goal, instruction, description, action type)	activity (title, content), action type (mainly social web actions), event (start time, end time), location	action type (without predefined action types), event (time, duration)	project (name)		project	business function (construction, installation, dispose, lease out, maintain, provide service, sell, buy)	event (start time, end time, duration, location)	
		F-01.b	Incentive mechanism						price specification, payment method		
		F-01.c	Evaluation mechanism								
		F-01.d	Human requirement								
		F-01.e	Technical requirement			operating system, platform, programming languages					
		F-01.f	Contribution and external resources	display name, content, summary, object type (article, audio, badge, bookmark, collection, comment, file, image, note, product, question, review, service, video)	item (title, type), device, application	repository	web resource (title, description, format, type, source)	document, image			
F-02	Task allocation	F-02.a	User description	person (display name)	user information (user name, email, discipline)	(reuse FOAF)	creator	person, (family name, given name, age, gender, and several other social web properties), organization	business entity (name, description, legal name, category, etc.), business entity type (business, end user, public institution, reseller) (reuse sche-		people (extensive set of properties to describe a user), company, organization

Functional requirement		Aspect		Activity Streams	CAM	DOAP	DC	FOAF	GR	hCalendar/ h-event	hCard/ h-card, vCard
									ma.org)		
		F-02.b	Suitability (interest, skills, experience, knowledge)					interest, made, publications, current project			job-title
		F-02.d	Availability (time and place)	location					location, opening hours specification		address, geographic location
F-03	Team building	F-03.a	Social network (friendship, work relationship, membership)	follow (person), join (group), leave (group), make-friend, remove-friend request-friend		(reuse FOAF)		knows (person), member (of organization)	(reuse schema.org)		member (of organization)
		F-03.b	Online community or Activity streams	activity (actor, verb, object, target)	event (action, session, item, context)						
F-04	Workflow and quality control	F-04a	Access rights (permission, status)				access rights				
		F-04b	Sequence description	prerequisites (activity)							

Functional Requirement		Aspect		Open Social	PROV	schema.org	SIOC	WS-BPEL	XPDL	XFN	CSM
F-01	Task specification	F-01.a	Task description (goal, instruction, description, action type)	activity (title, body, posted time, priority)	activity (start time, end time, location)	action (name, description, image, start time, end time, location, result, several specific action types), event (start date, end date, duration),		process definition (name, priority, description, subject), user task	process (process name, description), human task (priority, duration, time estimation, valid from, valid to, waiting time, working time)		project (goal, category), task (action type, latency type, instruction, submission time, closure time, complexity level, priority level, visibility mode, interaction mode, estimated duration)
		F-01.b	Incentive mechanism								reward (amount, currency, nature, type, payout method)
		F-01.c	Evaluation mechanism								evaluation mechanism (time, source, method, subject, specificity)
		F-01.d	Human requirement								human requirement

Functional Requirement		Aspect		Open Social	PROV	schema.org	SIOC	WS-BPEL	XPDL	XFN	CSM
											(gender, age min, age max, language, country, job title, qualification type, qualification level, approval rate, number of approved tasks)
		F-01.e	Technical requirement			instrument					technical requirement (hardware device, hardware feature, operating system, software platform)
		F-01.f	Contribution and external resources	messages, media items, additional objects may be defined	entity	creative work (article, book, code, comment, dataset, map, review, etc.), media objects (audio, data, image, music, video)					contribution, external resource (reuse of DC)
F-02	Task allocation	F-02.a	User description	person (display name, alternate names, about me, name, native name, preferred name, preferred username, and several social web properties), organization (department, type, field, etc.)	(reuse FOAF)	person (name, additional name, gender, given name, family name, nationality), organization (legal name, brand, makes offer, DUNS, GLN)	user account (implemented by using FOAF) user group (set of user accounts), role		participants (name, description, type)		(reuse of FOAF, GR, and schema.org)
		F-02.b	Suitability (interest, skills, experience, knowledge)			job title, honorific title, economic activity (isicV4, NAICS)					(reuse of FOAF, GR, and schema.org)
		F-02.d	Availability (time and place)	address, location		address, location (home location, work location)					(reuse of FOAF, GR, and schema.org)
F-03	Team	F-03.a	Social network	member (of	(reuse	affiliation, same	(reuse			diverse	(reuse of FOAF, GR,

Functional Requirement		Aspect		Open Social	PROV	schema.org	SIOC	WS-BPEL	XPDL	XFN	CSM
	building		(friendship, work relationship, mem-bership)	organization, group)	FOAF)	as, alumni of, children, col-league, follows, knows, member of, parent, related to, sibling, spouse, works for	FOAF)			types of relationships (friendship, physical and geographical relations, professional contacts, family mem-bership)	and schema.org)
		F-03.b	Online community or Activity streams	activity (actor, generator, object, target, verb)	provenance (agent, entity, activity)		forum (item, post, site, space, thread, container)				(reuse of SIOC)
F-04	Workflow and quality control	F-04a	Access rights (permission, status)				permission, status		access level (private, public), Status (none, ready, active, cancelled, abort-ing, aborted, completing, completed), publication status (under revision, re-leased, under test)		target audience, confi-dentiality level
		F-04b	Sequence description					several ele-ments to describe the workflow sequence and logic (repeat until, for each, while, if, etc.); as-signment to people (only in parallel or in sequence)	different gate-way types (xor, or, and, exclu-sive, inclusive, parallel, com-plex)		aggregation type, num-ber of assignments, includes (relation) has next (relation), has previous (relation)

APPENDIX E (CSM ONTOLOGY SOURCE CODE)

```
<?xml version="1.0"?>


<!DOCTYPE rdf:RDF [
    <!ENTITY dcterms "http://purl.org/dc/terms/" >
    <!ENTITY foaf "http://xmlns.com/foaf/0.1/" >
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY dc "http://purl.org/dc/elements/1.1/" >
    <!ENTITY gr "http://purl.org/goodrelations/v1#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>


<rdf:RDF xmlns="http://purl.org/csm/1.0#"
     xml:base="http://purl.org/csm/1.0"
     xmlns:dc="http://purl.org/dc/elements/1.1/"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:foaf="http://xmlns.com/foaf/0.1/"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:gr="http://purl.org/goodrelations/v1#"
     xmlns:dcterms="http://purl.org/dc/terms/">
    <owl:Ontology rdf:about="http://purl.org/csm/1.0">
        <dc:date rdf:datatype="&xsd;string">2013-09-22</dc:date>
        <dc:title rdf:datatype="&xsd;string">CSM Ontology – An Enterprise
Crowdsourcing Ontology</dc:title>
        <dc:creator rdf:datatype="&xsd;string">Lars Hetmank</dc:creator>
        <owl:versionInfo rdf:datatype="&xsd;string">Revision: 1.0</owl:versionInfo>
        <dc:rights xml:lang="en">This work is distributed under a Creative Commons
Attribution License (http://creativecommons.org/licenses/by/3.0/).</dc:rights>
        <owl:imports rdf:resource="http://purl.org/goodrelations/v1"/>
        <owl:imports rdf:resource="http://xmlns.com/foaf/0.1/"/>
    </owl:Ontology>

<!--///////////////////////////////////////////////////////////////////////////////////
//
// Object Properties
//
///////////////////////////////////////////////////////////////////////////////////-->

```

```
    <!-- http://purl.org/csm/1.0#hasActionType -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#hasActionType">
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#ActionType"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#hasAggregationType -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#hasAggregationType">
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#AggregationType"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#hasEvaluationMechanism -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#hasEvaluationMechanism">
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#EvaluationMechanism"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#hasEvaluationMethod -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#hasEvaluationMethod">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#EvaluationMechanism"/>
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#EvaluationMethod"/>
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#hasEvaluationSource -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#hasEvaluationSource">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#EvaluationMechanism"/>
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#EvaluationSource"/>
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#hasEvaluationSpecificity -->
    <owl:ObjectProperty
rdf:about="http://purl.org/csm/1.0#hasEvaluationSpecificity">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#EvaluationMechanism"/>
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#EvaluationSpecificity"/>
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#hasEvaluationSubject -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#hasEvaluationSubject">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#EvaluationMechanism"/>
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#EvaluationSubject"/>
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#hasEvaluationTime -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#hasEvaluationTime">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#EvaluationMechanism"/>
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#EvaluationTime"/>
```

```
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#hasExternalResource -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#hasExternalResource">
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#ExternalResource"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#hasLatencyType -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#hasLatencyType">
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#LatencyType"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#hasNext -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#hasNext">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#Task"/>
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#hasPayoutMethod -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#hasPayoutMethod">
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#PayoutMethod"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#RewardMechanism"/>
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#hasPrevious -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#hasPrevious">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#Task"/>
        <owl:inverseOf rdf:resource="http://purl.org/csm/1.0#hasNext"/>
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#hasRewardMechanism -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#hasRewardMechanism">
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#RewardMechanism"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#hasRewardNature -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#hasRewardNature">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#RewardMechanism"/>
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#RewardNature"/>
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#hasRewardType -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#hasRewardType">
```

```
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#RewardMechanism"/>
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#RewardType"/>
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#hasTargetAudience -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#hasTargetAudience">
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#TargetAudience"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#hasTask -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#hasTask">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Project"/>
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#Task"/>
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#imposes -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#imposes">
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#Requirement"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#issues -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#issues">
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#Project"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Requester"/>
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#owns -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#owns">
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#Contribution"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
    </owl:ObjectProperty>

    <!-- http://purl.org/csm/1.0#submits -->
    <owl:ObjectProperty rdf:about="http://purl.org/csm/1.0#submits">
        <rdfs:range rdf:resource="http://purl.org/csm/1.0#Contribution"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Participant"/>
    </owl:ObjectProperty>

<!--///////////////////////////////////////////////////////////////////////////////
//
// Data properties
//
////////////////////////////////////////////////////////////////////////////////-->

    <!-- http://purl.org/csm/1.0#ageMax -->
```

```
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#ageMax">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#HumanRequirement"/>
        <rdfs:range rdf:resource="&xsd;unsignedByte"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#ageMin -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#ageMin">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#HumanRequirement"/>
        <rdfs:range rdf:resource="&xsd;unsignedByte"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#amount -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#amount">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#RewardMechanism"/>
        <rdfs:range rdf:resource="&xsd;float"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#approvalRate -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#approvalRate">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#HumanRequirement"/>
        <rdfs:range rdf:resource="&xsd;float"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#category -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#category">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Project"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#closureTime -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#closureTime">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Project"/>
        <rdfs:range rdf:resource="&xsd;dateTime"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#complexityLevel -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#complexityLevel">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#confidentialityLevel -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#confidentialityLevel">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#country -->
```

```
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#country">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#HumanRequirement"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#currency -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#currency">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#RewardMechanism"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#description -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#description">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Contribution"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#EvaluationMechanism"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#ExternalResource"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Project"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Requirement"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#RewardMechanism"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#estimatedDuration -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#estimatedDuration">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
        <rdfs:range rdf:resource="&xsd;int"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#gender -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#gender">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#HumanRequirement"/>
        <rdfs:range rdf:resource="&xsd;unsignedByte"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#goal -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#goal">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Project"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#hardwareDevice -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#hardwareDevice">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#TechnicalRequirement"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
    </owl:DatatypeProperty>
```

```
    <!-- http://purl.org/csm/1.0#hardwareFeature -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#hardwareFeature">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#TechnicalRequirement"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#instruction -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#instruction">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#interactionMode -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#interactionMode">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
        <rdfs:range rdf:resource="&xsd;boolean"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#jobTitle -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#jobTitle">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#HumanRequirement"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#language -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#language">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#HumanRequirement"/>
        <rdfs:range rdf:resource="&xsd;language"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#numberOfApprovedTask -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#numberOfApprovedTask">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#HumanRequirement"/>
        <rdfs:range rdf:resource="&xsd;unsignedInt"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#numberOfAssignments -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#numberOfAssignments">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
        <rdfs:range rdf:resource="&xsd;unsignedInt"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#operatingSystem -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#operatingSystem">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#TechnicalRequirement"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
    </owl:DatatypeProperty>
```

```
    <!-- http://purl.org/csm/1.0#priorityLevel -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#priorityLevel">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#qualificationLevel -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#qualificationLevel">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#HumanRequirement"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#qualificationType -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#qualificationType">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#HumanRequirement"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#softwarePlatform -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#softwarePlatform">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#TechnicalRequirement"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#submissionTime -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#submissionTime">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Project"/>
        <rdfs:range rdf:resource="&xsd;dateTime"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#title -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#title">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Contribution"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#EvaluationMechanism"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#ExternalResource"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Project"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Requirement"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#RewardMechanism"/>
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
    </owl:DatatypeProperty>

    <!-- http://purl.org/csm/1.0#visibilityMode -->
    <owl:DatatypeProperty rdf:about="http://purl.org/csm/1.0#visibilityMode">
        <rdfs:domain rdf:resource="http://purl.org/csm/1.0#Task"/>
        <rdfs:range rdf:resource="&xsd;boolean"/>
    </owl:DatatypeProperty>
```

```

<!--///////////////////////////////////////////////////////////////////////////////////
//
// Classes
//
///////////////////////////////////////////////////////////////////////////////////-->

    <!-- http://purl.org/csm/1.0#ActionType -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#ActionType"/>

    <!-- http://purl.org/csm/1.0#AggregationType -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#AggregationType"/>

    <!-- http://purl.org/csm/1.0#Contribution -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#Contribution">
        <rdfs:subClassOf rdf:resource="http://purl.org/csm/1.0#EvaluationSubject"/>
    </owl:Class>

    <!-- http://purl.org/csm/1.0#EvaluationMechanism -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#EvaluationMechanism"/>

    <!-- http://purl.org/csm/1.0#EvaluationMethod -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#EvaluationMethod"/>

    <!-- http://purl.org/csm/1.0#EvaluationSource -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#EvaluationSource"/>

    <!-- http://purl.org/csm/1.0#EvaluationSpecificity -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#EvaluationSpecificity"/>

    <!-- http://purl.org/csm/1.0#EvaluationSubject -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#EvaluationSubject"/>

    <!-- http://purl.org/csm/1.0#EvaluationTime -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#EvaluationTime"/>

    <!-- http://purl.org/csm/1.0#ExternalResource -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#ExternalResource"/>

    <!-- http://purl.org/csm/1.0#HumanRequirement -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#HumanRequirement">
        <rdfs:subClassOf rdf:resource="http://purl.org/csm/1.0#Requirement"/>
    </owl:Class>

    <!-- http://purl.org/csm/1.0#LatencyType -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#LatencyType"/>

```

```
    <!-- http://purl.org/csm/1.0#Participant -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#Participant">
        <rdfs:subClassOf rdf:resource="http://purl.org/csm/1.0#EvaluationSubject"/>
        <rdfs:subClassOf rdf:resource="http://purl.org/csm/1.0#User"/>
    </owl:Class>

    <!-- http://purl.org/csm/1.0#PayoutMethod -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#PayoutMethod"/>

    <!-- http://purl.org/csm/1.0#Project -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#Project"/>

    <!-- http://purl.org/csm/1.0#Requester -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#Requester">
        <rdfs:subClassOf rdf:resource="http://purl.org/csm/1.0#User"/>
    </owl:Class>

    <!-- http://purl.org/csm/1.0#Requirement -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#Requirement"/>

    <!-- http://purl.org/csm/1.0#RewardMechanism -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#RewardMechanism"/>

    <!-- http://purl.org/csm/1.0#RewardNature -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#RewardNature"/>

    <!-- http://purl.org/csm/1.0#RewardType -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#RewardType"/>

    <!-- http://purl.org/csm/1.0#TargetAudience -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#TargetAudience"/>

    <!-- http://purl.org/csm/1.0#Task -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#Task"/>

    <!-- http://purl.org/csm/1.0#TechnicalRequirement -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#TechnicalRequirement">
        <rdfs:subClassOf rdf:resource="http://purl.org/csm/1.0#Requirement"/>
    </owl:Class>

    <!-- http://purl.org/csm/1.0#User -->
    <owl:Class rdf:about="http://purl.org/csm/1.0#User">
        <owl:equivalentClass rdf:resource="&gr;BusinessEntity"/>
        <owl:equivalentClass rdf:resource="&foaf;Person"/>
        <owl:equivalentClass>
            <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
```

```
                    <rdf:Description rdf:about="http://schema.org/Organization"/>
                    <rdf:Description rdf:about="http://schema.org/Person"/>
                </owl:unionOf>
            </owl:Class>
        </owl:equivalentClass>
        <rdfs:subClassOf rdf:resource="http://purl.org/csm/1.0#EvaluationSource"/>
    </owl:Class>

    <!-- http://purl.org/goodrelations/v1#BusinessEntity -->
    <rdf:Description rdf:about="&gr;BusinessEntity"/>

    <!-- http://xmlns.com/foaf/0.1/Person -->
    <rdf:Description rdf:about="&foaf;Person"/>

<!--///////////////////////////////////////////////////////////////////////////////////
//
// Individuals
//
///////////////////////////////////////////////////////////////////////////////////-->

    <!-- http://purl.org/csm/1.0#AcceptReject -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#AcceptReject">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#EvaluationSpecificity"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#After -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#After">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#EvaluationTime"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Algorithm -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Algorithm">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#EvaluationSource"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#AllParticipants -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#AllParticipants">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#PayoutMethod"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#AssessmentForm -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#AssessmentForm">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#EvaluationSpecificity"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Before -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Before">
```

```
        <rdf:type rdf:resource="http://purl.org/csm/1.0#EvaluationTime"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Categorize -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Categorize">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#ActionType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Code -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Code">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#ActionType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#ControlGroup -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#ControlGroup">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#EvaluationMethod"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Coupon -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Coupon">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#RewardType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Create -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Create">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#ActionType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Design -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Design">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#ActionType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Discount -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Discount">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#RewardType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Execute -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Execute">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#ActionType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#External -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#External">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#TargetAudience"/>
    </owl:NamedIndividual>
```

```
    <!-- http://purl.org/csm/1.0#Fixed -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Fixed">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#RewardNature"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#FreeForm -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#FreeForm">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#EvaluationSpecificity"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#GenerateIdea -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#GenerateIdea">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#ActionType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Good -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Good">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#RewardType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Hybrid -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Hybrid">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#TargetAudience"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Immediate -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Immediate">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#LatencyType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Integrative -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Integrative">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#AggregationType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Internal -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Internal">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#TargetAudience"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Iterative -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Iterative">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#AggregationType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Label -->
```

```
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Label">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#ActionType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Lottery -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Lottery">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#RewardType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#MajorityDecision -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#MajorityDecision">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#EvaluationMethod"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Money -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Money">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#RewardType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#NoReward -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#NoReward">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#RewardType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#PerformanceBased -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#PerformanceBased">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#RewardNature"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Play -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Play">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#ActionType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Proportional -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Proportional">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#RewardNature"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Rank -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Rank">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#ActionType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Rate -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Rate">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#ActionType"/>
```

```
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Rating -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Rating">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#EvaluationSpecificity"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#ResourceAccess -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#ResourceAccess">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#RewardType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Selective -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Selective">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#AggregationType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Share -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Share">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#ActionType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Simultaneously -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Simultaneously">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#EvaluationTime"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#SolveProblem -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#SolveProblem">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#ActionType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Test -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Test">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#ActionType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#ThirdParty -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#ThirdParty">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#EvaluationSource"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#TopXParticipants -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#TopXParticipants">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#PayoutMethod"/>
    </owl:NamedIndividual>

```

```
    <!-- http://purl.org/csm/1.0#Transcribe -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Transcribe">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#ActionType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Translate -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Translate">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#ActionType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Untimed -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Untimed">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#LatencyType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Verify -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Verify">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#ActionType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#VirtualPoints -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#VirtualPoints">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#RewardType"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Voluntary -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Voluntary">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#RewardNature"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#WinnerParticipants -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#WinnerParticipants">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#PayoutMethod"/>
    </owl:NamedIndividual>

    <!-- http://purl.org/csm/1.0#Write -->
    <owl:NamedIndividual rdf:about="http://purl.org/csm/1.0#Write">
        <rdf:type rdf:resource="http://purl.org/csm/1.0#ActionType"/>
    </owl:NamedIndividual>
</rdf:RDF>



<!-- Generated by the OWL API (version 3.4.2) http://owlapi.sourceforge.net -->
```

APPENDIX F (SAMPLE DATA INSTANCE 1)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="author" content="LH">
  <title>Example: Translate a technical specification</title>
</head>

<body prefix="csm: http://purl.org/csm/1.0# foaf: http://xmlns.com/foaf/0.1/">
  <div typeof="csm:Requester" about="http://example.org/#requester" >
    <div rel="foaf:page" resource="http://www.example.org/#company"></div>
    <div rel="csm:issues">
      <div typeof="csm:Project" about="http://example.org/#project">
        <div property="csm:title">
          Translation project
        </div>
        <div property="csm:category">
          Research and product development
        </div>
        <div rel="csm:includes">

          <!-- Task 01 -->
          <div typeof="csm:Task" about="http://example.org/#task01">
            <div property="csm:title">
              Translate technical specification
            </div>
            <div property="csm:instruction">
              Translate the content module from French to German.
            </div>
            <div property="csm:submissionTime" content="2014-09-03T09:00:00Z"
datatype="xsd:dateTime">
              The task will be available on 3rd of September, 2014 (at 9.00 am).
            </div>
            <div property="csm:closureTime" content="2014-09-17T16:00:00Z"
datatype="xsd:dateTime">
              The task will be closed on 17th of September, 2014 (at 4.00 pm).
            </div>
            <div property="csm:estimatedDuration" content="30" datatype="xsd:int">
              Estimated time of duration: 30 min.
            </div>
            <div property="csm:numberOfAssignments" content="1" datatype="xsd:int">
              Number of participants that will be assigned to the task: 1.
            </div>
```

```
            <div rel="csm:hasTargetAudience"
resource="http://purl.org/csm/1.0#Hybrid"></div>
            <div rel="csm:hasActionType"
resource="http://purl.org/csm/1.0#Translate"></div>
            <div rel="csm:imposes">
              <div typeof="csm:HumanRequirement"
about="http://example.org/#language_fr_de">
                <div property="csm:language" content="fr"
datatype="xsd:language"></div>
                <div property="csm:language" content="de"
datatype="xsd:language"></div>
              </div>
            </div>
            <div rel="csm:hasRewardMechanism">
              <div typeof="csm:RewardMechanism"
about="http://example.org/#reward01">
                <div property="csm:amount" content="80" datatype="xsd:float"></div>
                <div property="csm:currency" content="EUR"
datatype="xsd:string"></div>
                <div rel="csm:hasRewardNature"
resource="http://purl.org/csm/1.0#Fixed"></div>
                <div rel="csm:hasRewardType"
resource="http://purl.org/csm/1.0#Money"></div>
                <div rel="csm:hasPayoutMethod"
resource="http://purl.org/csm/1.0#AllParticipants"></div>
              </div>
            </div>
          </div>

          <!-- Task 02 -->
          <div typeof="csm:Task" about="http://example.org/#task02">
            <div property="csm:title">
              Translate memo
            </div>
            <div property="csm:instruction">
              Translate the memo from German into Polish.
            </div>
            <div property="csm:submissionTime" content="2014-09-03T09:00:00Z"
datatype="xsd:dateTime">
              The task will be available on 3rd of September, 2014 (at 9.00 am).
            </div>
            <div property="csm:closureTime" content="2014-09-17T16:00:00Z"
datatype="xsd:dateTime">
              The task will be closed on 17th of September, 2014 (at 4.00 pm).
            </div>
            <div property="csm:estimatedDuration" content="30" datatype="xsd:int">
              Estimated time of duration: 15 min.
```

```
      </div>
      <div property="csm:numberOfAssignments" content="1" datatype="xsd:int">
        Number of participants that will be assigned to the task: 1.
      </div>
      <div rel="csm:hasTargetAudience"
resource="http://purl.org/csm/1.0#Internal"></div>
      <div rel="csm:hasActionType"
resource="http://purl.org/csm/1.0#Translate"></div>
      <div rel="csm:imposes">
        <div typeof="csm:HumanRequirement"
about="http://example.org/#language_de_pl">
          <div property="csm:language" content="de"
datatype="xsd:language"></div>
          <div property="csm:language" content="pl"
datatype="xsd:language"></div>
        </div>
      </div>
      <div rel="csm:hasRewardMechanism">
        <div typeof="csm:RewardMechanism"
about="http://example.org/#reward02">
          <div property="csm:amount" content="5" datatype="xsd:float"></div>
          <div rel="csm:hasRewardNature"
resource="http://purl.org/csm/1.0#Fixed"></div>
          <div rel="csm:hasRewardType"
resource="http://purl.org/csm/1.0#VirtualPoints"></div>
          <div rel="csm:hasPayoutMethod"
resource="http://purl.org/csm/1.0#AllParticipants"></div>
        </div>
      </div>
     </div>

    </div>
   </div>
  </div>
 </div>
</body>
</html>
```

APPENDIX G (SAMPLE DATA INSTANCE 2)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="author" content="LH">
  <title>Example: Develop and test a software application</title>
</head>

<body prefix="csm: http://purl.org/csm/1.0# foaf: http://xmlns.com/foaf/0.1/">
  <div typeof="csm:Requester" about="http://example.org/#requester" >
    <div rel="csm:issues">
      <div typeof="csm:Project" about="http://example.org/#project">
        <div property="csm:title">
          CSM Annotator
        </div>
        <div property="csm:category">
          Software development
        </div>
        <div rel="csm:includes">

          <!-- Task 01 -->
          <div typeof="csm:Task" about="http://example.org/#task01">
            <div property="csm:title">
              Create graphical user interface
            </div>
            <div property="csm:instruction">
              Code graphical user interface
            </div>
            <div rel="csm:hasActionType"
resource="http://purl.org/csm/1.0#Code"></div>
            <div property="csm:numberOfAssignments" content="1" datatype="xsd:int">
              Number of participants that will be assigned to the task: 1.
            </div>
            <div rel="csm:imposes">
              <div typeof="csm:HumanRequirement" about="http://example.org/#html5">
                <div property="csm:qualificationType">HTML5</div>
                <div property="csm:qualificationLevel">Expert</div>
                <div property="csm:jobTitle" content="151130"
datatype="xsd:string">
                  Software Developers and Programmers
                </div>
              </div>
            </div>
```

```
            <div rel="csm:hasEvaluationMechanism">
              <div typeof="csm:EvaluationMechanism"
about="http://example.org/#evaluation01">
                <div rel="csm:hasEvaluationSubject"
resource="http://purl.org/csm/1.0#Participant"></div>
                <div rel="csm:hasEvaluationTime"
resource="http://purl.org/csm/1.0#After"></div>
                <div rel="csm:hasEvaluationSource"
resource="http://purl.org/csm/1.0#Participant"></div>
              </div>
            </div>
            <div rel="csm:owns"
resource="http://example.org/#contribution01"></div>
            <div rel="csm:hasNext" resource="http://example.org/#task02"></div>
          </div>

          <!-- Task 02 -->
          <div typeof="csm:Task" about="http://example.org/#task02">
            <div property="csm:title">
              Establish database connection
            </div>
            <div property="csm:instruction">
              Code database access
            </div>
            <div rel="csm:hasActionType"
resource="http://purl.org/csm/1.0#Code"></div>
            <div property="csm:numberOfAssignments" content="1" datatype="xsd:int">
              Number of participants that will be assigned to the task: 1.
            </div>
            <div rel="csm:imposes">
              <div typeof="csm:HumanRequirement" about="http://example.org/#php">
                <div property="csm:qualificationType">PHP</div>
                <div property="csm:qualificationLevel">Expert</div>
                <div property="csm:jobTitle"content="151141" datatype="xsd:string">
                  Specialist, Database Management System
                </div>
              </div>
            </div>
            <div rel="csm:hasEvaluationMechanism">
              <div typeof="csm:EvaluationMechanism"
about="http://example.org/#evaluation02">
                <div rel="csm:hasEvaluationSubject"
resource="http://purl.org/csm/1.0#Participant"></div>
                <div rel="csm:hasEvaluationTime"
resource="http://purl.org/csm/1.0#Before"></div>
                <div rel="csm:hasEvaluationSource"
resource="http://purl.org/csm/1.0#Requester"></div>
```

```
          </div>
        </div>
        <div rel="csm:hasNext" resource="http://example.org/#task03"></div>
        <div rel="csm:hasNext" resource="http://example.org/#task04  "></div>
      </div>

      <!-- Task 03 -->
      <div typeof="csm:Task" about="http://example.org/#task03">
        <div property="csm:title">
          Usability and functionality test
        </div>
        <div property="csm:instruction">
          Test web application on your Android mobile device
        </div>
        <div rel="csm:hasActionType"
resource="http://purl.org/csm/1.0#Test"></div>
        <div property="csm:numberOfAssignments" content="5" datatype="xsd:int">
          Number of participants that will be assigned to the task: 5.
        </div>
        <div rel="csm:imposes">
          <div typeof="csm:TechnicalRequirement"
about="http://example.org/#android"><div
property="csm:operatingSystem">Android</div>
          </div>
        </div>
        <div rel="csm:owns"
resource="http://example.org/#contribution02"></div>
        <div rel="csm:owns"
resource="http://example.org/#contribution03"></div>

      </div>

      <!-- Task 04 -->
      <div typeof="csm:Task" about="http://example.org/#task04">
         <div property="csm:title">
          Usability and functionality test
        </div>
        <div property="csm:instruction">
          Test web application on your iOS mobile device.
        </div>
        <div rel="csm:hasActionType"
resource="http://purl.org/csm/1.0#Test"></div>
        <div property="csm:numberOfAssignments" content="5" datatype="xsd:int">
          Number of participants that will be assigned to the task: 5.
        </div>
        <div rel="csm:imposes">
```

```
              <div typeof="csm:TechnicalRequirement"
about="http://example.org/#iOS">
                <div property="csm:operatingSystem">iOS</div>
              </div>
            </div>
          </div>

        </div>
      </div>
    </div>
  </div>

  <!-- Participant 01 -->
  <div typeof="csm:Participant" about="http://example.org/#participant01" >
    <div rel="csm:submits">
      <div typeof="csm:Contribution"
about="http://example.org/#contribution01"></div>
    </div>
  </div>

  <!-- Participant 02 -->
  <div typeof="csm:Participant" about="http://example.org/#participant02" >
    <div rel="csm:submits">
      <div typeof="csm:Contribution"
about="http://example.org/#contribution02"></div>
    </div>
  </div>

  <!-- Participant 03 -->
  <div typeof="csm:Participant" about="http://example.org/#participant03" >
    <div rel="csm:submits">
      <div typeof="csm:Contribution"
about="http://example.org/#contribution03"></div>
    </div>
  </div>

</body>
</html>
```