2013

Components and Functions of Crowdsourcing Systems – A Systematic Literature Review

Lars Hetmank
TU Dresden, Dresden, Germany, lars.hetmank@tu-dresden.de

Components and Functions of Crowdsourcing Systems – A Systematic Literature Review

Lars Hetmank

TU Dresden, Dresden, Germany
lars.hetmank@tu-dresden.de

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.

Keywords: crowdsourcing, crowdsourcing system, crowdsourcing application, crowdsourcing platform, systematic literature review

1 Introduction

The research of crowdsourcing is a vigorous research area that has been steadily increasing over the last several years [1] and there is still an ongoing need for scientific engagement in this field [2], [3]. 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 [4]. 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 [5]. Besides using existing external crowdsourcing solutions, such as Amazon Mechnical 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

11th International Conference on Wirtschaftsinformatik,
27th February – 01st March 2013, Leipzig, Germany

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 [6]. Since then a lot of various *crowdsourcing definitions* have been proposed. Recently, Estellés-Arolas and González-Ladrón-de-Guevara analyzed existing definitions of crowdsourcing and created an integrated definition that considers several specific aspects of the crowd, the initiator and the underlying process [7].

The *process perspective* on crowdsourcing was examined in detail by Geiger et al. who developed a taxonomic framework for crowdsourcing processes [8]. The authors identified four dimensions that describe how crowdsourcing processes can be configured, ranging from pre-selection of contributors, accessibility of contributors, and aggregation of contributors to remuneration for contribution.

Several authors have drawn their attention to *crowdsourcing taxonomies*. Rouse, for example, decomposed the term “crowdsourcing” into several subtypes [9]. 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 [10]. 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 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) [11]. Yuen et al. 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 [12]. The most recent, sophisticated classification of CSSs was proposed by Doan et al. [13]. 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, 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 [14]. 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 et al. specify a further conceptual framework. Their proposed framework contains four building blocks that are important in designing collective intelligence systems [15]. 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 [6], [7], (ii) the characterization of the crowdsourcing process [8], (iii) the development of a crowdsourcing taxonomy [9-13], and (iv) the introduction of a conceptual framework that supports the designing of CSSs [14], [15], 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 [16]. 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 [17].

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 (see Fig. 1).

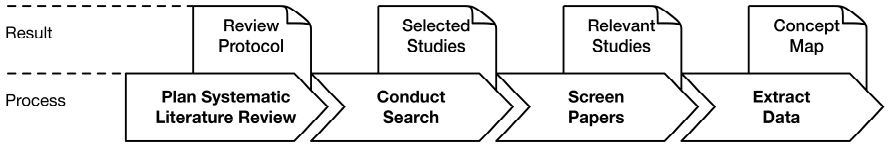

Fig. 1. Systematic literature review procedure

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 Jeff Howe [18]. 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 (see 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 Mechnanical Turk for evaluation research purposes, but that do not discuss any design issues were excluded.

[1] Requested 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.

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 [19]. An article was included in the set of relevant studies if it either met all inclusion criteria or was not rejected by any of the exclusion criteria. Simultaneously, each paper was classified according to publication type and research approach (see Fig. 2) [20].

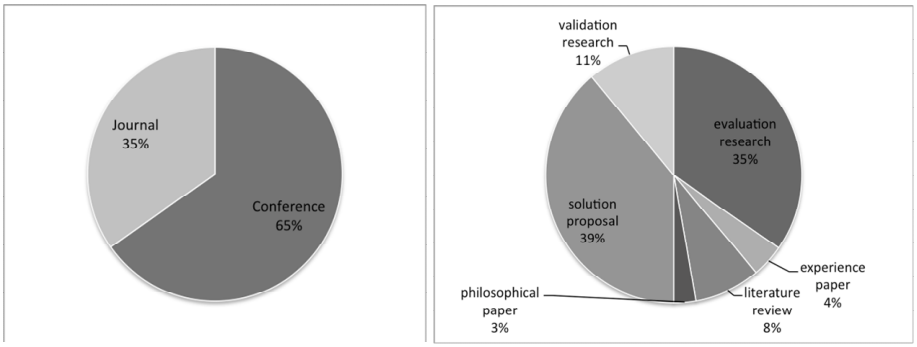

Fig. 2. Article distribution regarding publication type (left) and research type (right)

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 *crowsourcing system, crowdsourcing application* and *crowdsourcing platform*. With the aid of content analysis, all definitions were grouped in different perspectives of CSSs [21]. 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 map 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. I first address the question of existing definitions of CSSs and then draw attention to several design aspects of components and functions of a CSS.

4.1 Crowdsourcing System Definitions (RQ1)

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

[3] see also http://larshetmank.com/documents/wi2013_css_concept_map.pdf for more details of the concept map; the terms finally found are represented in Fig. 3

crowdsourcing platform (see Table 2). 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 [21]:

- 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 [34] addresses at least the organizational, the technical, and the process perspective, the definition of Treiber et al. [32] 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 map, 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 (Fig. 3).

Table 2. Collected definitions

Article	Definition of crowdsourcing system and its assigned perspective (O, T, P, H)
DiPalantino and Vojnovic [22]	… 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. [13]	… 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 et al. [23]	…creates a marketplace on which requesters offer tasks and workers accept and work on the tasks. (O)
Fraternali et al. [24]	… 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 et al. [25]	Every employer needs a mediator to access the worker crowd. This mediator is called a crowdsourcing platform … (O)
Hirth et al. [26]	… 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 et al. [27]	… distributes the work submitted by an employer among the human worker resources and acts as mediator between worker and employer. (O)
Jayakanthan et al. [28]	… enterprise crowdsourcing applications which aim to utilize the capabilities of members within the organization itself – particularly the employees within a large company. (H)
Karger et al. [29]	… 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 et al. [30]	… an effective tool making human skills and intelligence accessible to machines. (H)
Ross et al. [31]	… 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 et al. [32]	… distribute problem-solving tasks among a group of humans. (only weakly associated to P)
Venetis et al. [33]	… must post tasks for the humans, collect results, and cleanse and aggregate the answers provided by humans. (P)
Vukovic [34]	… 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. [35]	… collaborative cyberinfrastructure that can aggregate scattered resources, including both human brainpower and machine computational capacities. (H)
Zhang and van der Schaar [36]	… 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 [1]	… 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)

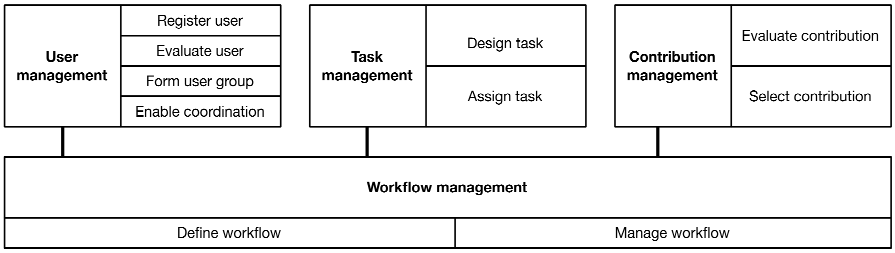

Fig. 3. Components and functions of crowdsourcing systems

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 [37].
- *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, prequalification tasks or gold standard data to determine the expertise or skill level of a worker [38]. The latter considers acceptance and rejection decisions of historic contributions [39]. Sometimes a certain user's answer will be directly compared to the answers of the other users responding to the same question [29]. 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 [40], [41], [42].
- *Form User Group.* Different types of users are motivated differently and hence need specific incentive mechanisms [43]. 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 [43]. 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 [24].
- *Enable Coordination.* A CSS needs appropriate mechanisms to facilitate collaboration and coordination [44]. 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)

[45]. 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 [38]. Furthermore, the type (e.g., straightforward, novel), the size, the reward or incentive scheme [22], [46], the submission time, the latency (e.g., immediate, waitable) [47], [48], the degree of confidentiality and the designated crowd should be carefully defined [49], [26]. 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 [50] or knowledge that is gained by applying social network analysis techniques to the existing crowd network [24]. 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 [51]. Finally, a requester may configure if the contributions of the solver can be seen by the other users or not [52].
- *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 [53]. 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 [47].

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 [54]. 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. [25]. 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 [55], [33]. 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 [56].

Workflow Management. A workflow management component is of crucial importance when designing complex tasks with global requirements and constraints [57], and helps to secure contribution quality [35]. 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 [57]. Workflows are either defined by the requester of the task or the crowd itself [58].
- *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 [59]. 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 [59].

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 designing of a semantic model for corporative knowledge-intensive problem solving in crowdsourcing environments that will be used to improve data portability between different CSSs as well as to connect to other business application software.

References

1. Zhao, Y., Zhu, Q.: Evaluation on crowdsourcing research: Current status and future direction. Information Systems Frontiers. 1–18 (2012)
2. Hammon, L., Hippner, H.: Crowdsourcing. Wirtschaftsinformatik. 54, 165–168 (2012)
3. Leimeister, J.M., Huber, M., Bretschneider, U., Krcmar, H.: Leveraging Crowdsourcing: Activation-Supporting Components for IT-Based Ideas Competition. Journal of Management Information Systems. 26, 197–224 (2009)
4. Greengard, S.: Following the crowd. Communications of the ACM. 54, 20–22 (2011)

5. Vukovic, M., Bartolini, C.: Towards a Research Agenda for Enterprise Crowdsourcing. In: Margaria, T., Steffen, B. (eds.): Leveraging Applications of Formal Methods, Verification, and Validation. pp. 425–434. Springer Berlin / Heidelberg (2010)
6. Brabham, D.C.: Crowdsourcing as a Model for Problem Solving: An Introduction and Cases. Convergence: The International Journal of Research into New Media Technologies. 14, 75–90 (2008)
7. Estellés-Arolas, E., González-Ladrón-de-Guevara, F.: Towards an integrated crowdsourcing definition. Journal of Information Science. 38 , 189–200 (2012)
8. Geiger, D., Seedorf, S., Schulze, T., Nickerson, R., Schader, M.: Managing the Crowd: Towards a Taxonomy of Crowdsourcing Processes. Proceedings of the Seventeenth Americas Conference on Information Systems. pp. 1–11 (2011)
9. Rouse, A.C.: A preliminary taxonomy of crowdsourcing. ACIS 2010 Proceedings (2010)
10. Schenk, E., Guittard, C.: Towards a characterization of crowdsourcing practices. Journal of Innovation Economics. 7, (2010)
11. Erickson, T.: Some Thoughts on a Framework for Crowdsourcing. Workshop on Crowdsourcing and Human Computation. pp. 1–4 (2011)
12. Yuen, M.-C., King, I., Leung, K.-S.: A Survey of Crowdsourcing Systems. Privacy, security, risk and trust (passat), 2011 IEEE third International Conference on Social Computing (socialcom). pp. 766–773 (2011)
13. Doan, A., Ramakrishnan, R., Halevy, A.Y.: Crowdsourcing systems on the World-Wide Web. Communications of the ACM. 54, 86–96 (2011)
14. Kazman, R., Chen, H.-M.: The metropolis model a new logic for development of crowdsourced systems. Communications of the ACM. 52, 76–84 (2009)
15. Malone, T.W., Laubacher, R., Dellarocas, C.: The collective intelligence genome. Engineering Management Review, IEEE. 38, 38–52 (2010)
16. Kitchenham, B.: Guidelines for performing Systematic Literature Reviews in Software Engineering. (2007)
17. Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., Linkman, S.: Systematic literature reviews in software engineering – A systematic literature review. Information and Software Technology. 51, 7–15 (2009)
18. Howe, J.: The rise of crowdsourcing. Wired magazine. 1–5 (2006)
19. Brereton, P., Kitchenham, B. a., Budgen, D., Turner, M., Khalil, M.: Lessons from applying the systematic literature review process within the software engineering domain. Journal of Systems and Software. 80, 571–583 (2007)
20. Wieringa, R., Maiden, N., Mead, N., Rolland, C.: Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. Requirements Engineering. 11, 102–107 (2006)
21. Bortz, J., Döring, N.: Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler. Springer, Heidelberg (2009)
22. DiPalantino, D., Vojnovic, M.: Crowdsourcing and all-pay auctions. In: Proceedings of the 10th ACM Conference on Electronic Commerce. pp. 119–128. ACM, New York (2009)
23. Franklin, M.J., Kossmann, D., Kraska, T., Ramesh, S., Xin, R.: CrowdDB: answering queries with crowdsourcing. In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of data. pp. 61–72. ACM, New York (2011)
24. Fraternali, P., Castelletti, A., Soncini-Sessa, R., Ruiz, C.V., Rizzoli, A.E.: Putting humans in the loop: Social computing for Water Resources Management. Environmental Modelling; Software. 37, 68–77 (2012)
25. Hirth, M., Hoßfeld, T., Tran-Gia, P.: Analyzing costs and accuracy of validation mechanisms for crowdsourcing platforms. Mathematical and Computer Modelling (2012)

26. Hirth, M., Hossfeld, T., Tran-Gia, P.: Anatomy of a Crowdsourcing Platform - Using the Example of Microworkers.com. Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS). In: 2011 Fifth International Conference on, pp. 322–329 (2011)
27. Hossfeld, T., Hirth, M., Tran-Gia, P.: Modeling of crowdsourcing platforms and granularity of work organization in future internet. In: Proceedings of the 23rd International Teletraffic Congress. pp. 142–149. ITCP (2011)
28. Jayakanthan, R., Sundararajan, D.: Enterprise crowdsourcing solutions for software development and ideation. Proceedings of the 2nd international workshop on Ubiquitous crowdsouring. pp. 25–28. ACM, New York (2011)
29. Karger, D.R., Oh, S., Shah, D.: Budget-optimal crowdsourcing using low-rank matrix approximations. Communication, Control, and Computing (Allerton). In: 49th Annual Allerton Conference on. pp. 284–291 (2011)
30. Lofi, C., Selke, J., Balke, W.T.: Information Extraction Meets Crowdsourcing: A Promising Couple. Datenbank-Spektrum. 12, 109–120 (2012)
31. Ross, J., Irani, L., Silberman, M.S., Zaldivar, A., Tomlinson, B.: Who are the crowdworkers?: shifting demographics in mechanical turk. In: Proceedings of the 28th of the International Conference extended abstracts on Human factors in Computing Systems. pp. 2863–2872. ACM, New York (2010)
32. Treiber, M., Schall, D., Dustdar, S., Scherling, C.: Tweetflows: flexible workflows with twitter. In: Proceedings of the 3rd International Workshop on Principles of Engineering Service-Oriented Systems. pp. 1–7. ACM, New York (2011)
33. Venetis, P., Garcia-Molina, H., Huang, K., Polyzotis, N.: Max algorithms in crowdsourcing environments. In: Proceedings of the 21st International Conference on World Wide Web. pp. 989–998. ACM, New York (2012)
34. Vukovic, M.: Crowdsourcing for Enterprises. Services - I, 2009 World Conference on. pp. 686–692 (2009)
35. Zhai, Z., Sempolinski, P., Thain, D., Madey, G., Wei, D., Kareem, A.: Expert-Citizen Engineering: “Crowdsourcing” Skilled Citizens. Dependable, Autonomic and Secure Computing (DASC). In: IEEE Ninth International Conference on. pp. 879–886 (2011)
36. Zhang, Y., Van der Schaar, M.: Reputation-based incentive protocols in crowdsourcing applications. INFOCOM. In: 2012 Proceedings IEEE. pp. 2140–2148 (2012)
37. Klinger, J., Lease, M.: Enabling trust in crowd labor relations through identity sharing. Proceedings of the American Society for Information Science and Technology. 48, 1–4 (2011)
38. Corney, J.R., Torres-Sánchez, C., Jagadeesan, A.P., Yan, X.T., Regli, W.C., Medellin, H.: Putting the crowd to work in a knowledge-based factory. Advanced Engineering Informatics. 24, 243–250 (2010)
39. Mashhadi, A.J., Capra, L.: Quality control for real-time ubiquitous crowdsourcing. In: Proceedings of the 2nd International Workshop on Ubiquitous Crowdsouring. pp. 5–8. ACM, New York (2011)
40. Ipeirotis, P.G., Provost, F., Wang, J.: Quality management on Amazon Mechanical Turk. Proceedings of the ACM SIGKDD Workshop on Human Computation. pp. 64–67. ACM, New York (2010)
41. Schall, D.: Expertise ranking using activity and contextual link measures. Data & Knowledge Engineering. 71, 92–113 (2012)
42. Archak, N.: 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. ACM, New York (2010)

43. Heipke, C.: Crowdsourcing geospatial data. ISPRS Journal of Photogrammetry and Remote Sensing. 65, 550–557 (2010)
44. Gao, H., Barbier, G., Goolsby, R.: Harnessing the Crowdsourcing Power of Social Media for Disaster Relief. Intelligent Systems, IEEE. 26, 10–14 (2011)
45. Liu, Y., Lehdonvirta, V., Alexandrova, T., Nakajima, T.: Drawing on mobile crowds via social media. Multimedia Systems. 18, 53–67 (2012)
46. Liu, Y., Alexandrova, T., Nakajima, T.: Gamifying intelligent environments. Proceedings of the 2011 International ACM workshop on Ubiquitous meta user interfaces. pp. 7–12. ACM, New York (2011)
47. Liu, Y., Lehdonvirta, V., Kleppe, M., Alexandrova, T., Kimura, H., Nakajima, T.: A crowdsourcing based mobile image translation and knowledge sharing service. In: Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia. pp. 6:1–6:9. ACM, New York (2010)
48. Bernstein, M.S., Brandt, J., Miller, R.C., Karger, D.R.: Crowds in two seconds: enabling realtime crowd-powered interfaces. Proceedings of the 24th annual ACM Symposium on User interface Software and Technology. pp. 33–42. ACM, New York (2011)
49. Eickhoff, C., De Vries, A.: Increasing cheat robustness of crowdsourcing tasks. Information Retrieval. 1–17 (2011)
50. Shao, B., Shi, L., Xu, B., Liu, L.: Factors affecting participation of solvers in crowdsourcing: an empirical study from China. Electronic Markets. 22, 73–82 (2012)
51. Kazai, G., Kamps, J., Koolen, M., Milic-Frayling, N.: 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. ACM, New York (2011)
52. Aparicio, M., Costa, C.J., Braga, A.S.: Proposing a system to support crowdsourcing. In: Proceedings of the Workshop on Open Source and Design of Communication. pp. 13–17. ACM, New York (2012)
53. Govindaraj, D., K.V.M., N., Nandi, A., Narlikar, G., Poosala, V.: MoneyBee: Towards enabling a ubiquitous, efficient, and easy-to-use mobile crowdsourcing service in the emerging market. Bell Labs Technical Journal. 15, 79–92 (2011)
54. Dow, S., Kulkarni, A., Klemmer, S., Hartmann, B.: Shepherding the crowd yields better work. In: Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work. pp. 1013–1022. ACM, New York (2012)
55. Marcus, A., Wu, E., Karger, D., Madden, S., Miller, R.: Human-powered sorts and joins. Proc. VLDB Endow. 5, 13–24 (2011)
56. Barbier, G., Zafarani, R., Gao, H., Fung, G., Liu, H.: Maximizing benefits from crowdsourced data. Computational & Mathematical Organization Theory. 1–23 (2012)
57. Zhang, H., Law, E., Miller, R., Gajos, K., Parkes, D., Horvitz, E.: Human computation tasks with global constraints. In: Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems. pp. 217–226. ACM, New York (2012)
58. Kulkarni, A., Can, M., Hartmann, B.: Collaboratively crowdsourcing workflows with turkomatic. In: Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work. pp. 1003–1012. ACM, New York, USA (2012)
59. Kittur, A., Khamkar, S., André, P., Kraut, R.: CrowdWeaver: visually managing complex crowd work. In: Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work. pp. 1033–1036. ACM, New York (2012)