Pers Ubiquit Comput (2014) 18:2025–2034
DOI 10.1007/s00779-014-0758-3

ORIGINAL ARTICLE

Providing real-time assistance in disaster relief by leveraging crowdsourcing power

Dingqi Yang · Daqing Zhang · Korbinian Frank ·
Patrick Robertson · Edel Jennings ·
Mark Roddy · Michael Lichtenstern

Received: 21 August 2013 / Accepted: 13 January 2014 / Published online: 30 January 2014

Abstract Crowdsourcing platforms for disaster management have drawn a lot of attention in recent years due to their efficiency in disaster relief tasks, especially for disaster data collection and analysis. Although the on-site rescue staff can largely benefit from these crowdsourcing data, due to the rapidly evolving situation at the disaster site, they usually encounter various difficulties and have requests, which need to be resolved in a short time. In this paper, aiming at efficiently harnessing crowdsourcing power to provide those on-site rescue staff with real-time remote assistance, we design and develop a crowdsourcing disaster support platform by considering three unique features, viz., selecting and notifying relevant off-site users for individual request according to their expertise; providing collaborative working functionalities to off-site users; improving answer credibility via "crowd voting." To evaluate the platform, we conducted a series of experiments with three-round user trials and also a System Usability Scale survey after each trial. The results show that the platform can effectively support on-site rescue staff by leveraging crowdsourcing power and achieve good usability.

Keywords Disaster management platform · Crowdsourcing · System Usability Scale · SUS

D. Yang (✉) · D. Zhang
Department of Telecommunication Network and Services,
Institut Mines-TELECOM/TELECOM SudParis, 91011 Evry,
France
e-mail: Dingqi.Yang@it-sudparis.eu

D. Zhang
e-mail: Daqing.Zhang@it-sudparis.eu

K. Frank · P. Robertson · M. Lichtenstern
Institute of Communications and Navigation,
German Aerospace Center, Oberpfaffenhofen, Germany
e-mail: Korbinian.Frank@dlr.de

P. Robertson
e-mail: Patrick.Robertson@dlr.de

M. Lichtenstern
e-mail: M.Lichtenstern@dlr.de

E. Jennings · M. Roddy
Telecommunications Software and Systems Group,
Waterford Institute of Technology, Waterford, Ireland
e-mail: ejennings@tssg.org

M. Roddy
e-mail: mroddy@tssg.org

1 Introduction

In recent years, a number of environmental disasters have been seen all over the world, such as the Haiti earthquake on January 12, 2010 and Fukushima nuclear disaster on March 11, 2011. In both cases, crowd power played an important role in disaster management. For example, Ushahidi,[1] an open-source crisis map platform founded in 2007, illustrated the power of the crowd in disaster data collection in the Haiti earthquake. Ushahidi Haiti Project gathered nearly 40,000 independent reports of which nearly 4,000 distinct events were plotted just after a few hours [12]. Similarly, after the severe earthquake triggered the devastating tsunami and Fukushima nuclear disaster, some Japanese engineers collaborated and launched Web services such as Sinsai[2] to assist the victims [11].

By leveraging power of the crowds, those crowdsourcing disaster management platforms have shown their

[1] http://www.ushahidi.com.

[2] sinsai.info.

efficiency in disaster relief tasks, such as disaster data collection and analysis. On the one hand, crowdsourcing data, such as on-site damage assessment, are collected in a very short time after a disaster, with very low costs. On the other hand, remote users of these crowdsourcing platforms can provide assistance such as disaster data analysis and online support to rescuers. By considering the location of crowdsourcing disaster management platform users, we classify them into two categories, viz., on-site users and off-site users.

- *On-site users*, including on-site rescuers and field command in On-Site Operations Coordination Center (OSOCC), are located at the disaster site. They can provide help such as victim searching, medication, damage assessment, and reporting.
- *Off-site users*, i.e., off-site volunteers,[3] are located off-site from the disaster and are willing to provide *on-site users* with remote assistance such as data analysis, navigation, translation, remote mental health counseling, Insarag[4] marking decoding, and disaster image analysis.

A well-designed crowdsourcing disaster management platform should empower both on-site and off-site users to efficiently provide help. In the current literature, most of the existing works focus on disaster data collection and incident reporting via crowdsourcing such as in [3, 5, 8, 13, 15, 16], which can provide timely disaster reports and damage assessment to disaster command staff. However, in disaster relief tasks such as victim searching and damage assessment, the on-site users usually suffer from unfamiliar situations and lack up-to-date information. Therefore, it is crucial to provide those on-site users with real-time assistance. In this paper, we focus on leveraging off-site users' efforts to provide on-site users with real-time assistance. Concretely, the following scenario extracted from EU SOCIETIES[5] (Self-Orchestrating Community ambIEnT IntElligent Spaces) project shows an example of such a use case.

An earthquake of magnitude 7.8 with epicenter about 20 km south–west of a coastal city has caused severe damage and casualties. The local response capacity is exceeded, and the government has requested international assistance. Several international rescue teams have been sent on-site to support the local disaster relief staff.

- Request 1: An on-site rescue team needs to go from City Hall to the East Coast Family Clinic, 121 Meyer Rd. But the common itineraries including ECP, Tanjong Rhu Flyover, Merdeka Bridge, Kallang Rd, and Sims Ave are impassable at the moment. Thus, they send a request for navigation.
- Request 2: To assess the damage on the transportation system, an on-site rescue team needs to know the names and locations of all ferry terminals and piers within 2-km radius from Keppel Island.
- Request 3: An on-site rescue team are looking for kindergartens and possibly other information that might be relevant (e.g., location and building type).

In the above scenario, on-site users generate requests that need to be resolved with a minimum delay. Using crowdsourcing efforts from off-site users, we are able to provide such help. However, such an approach still faces several challenges. Firstly, on-site users may generate various requests that require different types of expertise such as navigation, translation, or information search. The most straightforward approach is to let off-site users select the requests to which they can provide their help. However, dealing with a large amount of requests that need to be resolved in a short time, such an approach is inefficient. Therefore, selecting and notifying the most relevant off-site users about specific requests is necessary. Secondly, some requests may need the collaboration of off-site users in order to get comprehensive answers within satisfactory delay. For example, it is very difficult for a single off-site user to resolve *Request 3* in the above scenario within reasonable time, because such a request usually requires significant information search efforts and would be more than one individual user can provide. Hence, it is necessary to develop a collaboration mechanism for off-site users. Last but not least, as identified by most of the crowdsourcing research works, the credibility of such crowdsourcing data is not guaranteed. In other words, the answers provided by off-site users to the requests are not always accurate.

In this paper, we develop CDSP, a Crowdsourcing Disaster Support Platform, which is designed specifically to provide real-time assistance to on-site users by efficiently leveraging the crowdsourcing power of off-site users. Concretely, CDSP integrates a user selection feature to identify the most relevant off-site volunteers for a specific request according to user expertise, obtained from various sources such as manual input, extracting from one's social networks, and learning from one's historical usage records. Moreover, we adopt the "divide-and-conquer" approach for the requests that need collaboration of off-site volunteers and provide an instant message communication service to them for discussion. Finally, in order to alleviate the credibility concern, we implement a "voting" feature for users to vote for the best answer to a specific request.

[3] Since "off-site users" of crowdsourcing disaster management platforms often voluntarily provide their help, we do not differentiate the two terms throughout this paper.

[4] http://www.usar.nl/upload/docs/insarag_guidelines_july_2006.pdf.

[5] http://www.ict-societies.eu/.

To evaluate CDSP, we conducted experiments with three rounds of user trials by simulating the disaster situation described above and we continuously improve the platform over the three trials. Moreover, we conducted a System Usability Scale (SUS) survey after each trial. By quantitatively analyzing user logs and the SUS survey, the results show that using CDSP, off-site users can efficiently resolve the requests from on-site users and thus effectively support them in disaster relief tasks. The overall SUS score also shows that CDSP achieves good usability.

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 illustrates the CDSP design from both system and functionality perspectives. Section 4 presents the experiments including the three user trials and the results analysis, followed by discussion in Sect. 5. We conclude our work in Sect. 6.

2 Related work

Crowdsourcing has been widely studied in the disaster management domain, especially for crowdsourcing disaster data collection and analysis. Firstly, early efforts in disaster data collection focused on gathering volunteered geographic information, which is defined by Goodchild [7] in 2007. Then, Goodchild et al. [8] conducted a study on crowdsourcing geographic information for disaster response and identified that despite the various advantages of crowdsourcing data, such as time efficiency and low cost, improving data quality is still a challenge. Roche et al. [15] studied the perspectives of using volunteered geographic information, both in terms of technologies and applications, against crisis management processes. Chu et al. [3] developed a crowdsourcing support system for disaster surveillance that focused specifically on maintaining a comprehensive view of disaster sites with the help of on-site users. Palmer et al. [14] designed and implemented the RAVEN framework, which supports applications for collaboratively collecting structured disaster data on the Android OS. Secondly, in the domain of leveraging crowdsourcing efforts for disaster data analysis, Vivacqua et al. [17] discussed the possibilities of harnessing collective knowledge in disaster relief and presented an architecture and examples of how this could be accomplished. Ortmann et al. [13] studied information triage by leveraging Linked Open Data[6] in disaster data analysis. By leveraging machine learning methods, Schulz et al. [16] aimed at minimizing the human efforts for turning unstructured crowdsourcing disaster data into relevant information supporting the rescuers in making better decisions. These research works focused on efficiently leveraging crowdsourcing efforts to handle disaster data collection and analysis tasks.

Although the on-site rescue teams can largely benefit from these crowdsourcing disaster data, facing rapidly changing situations on disaster site, they may encounter various situations and have requests that need to be resolved in a short time. In our work, we focus on a specific use case that involves off-site users collaboratively providing real-time remote assistance to on-site users during their disaster relief tasks. To the best of our knowledge, there are very few research works tackling specifically the issue of providing real-time assistance to on-site users leveraging the collective efforts of off-site users, which is indeed important in disaster relief. However, providing real-time assistance to on-site users still faces several challenges for crowdsourcing disaster management. For example, Gao et al. [6] discussed both advantages and shortfalls of using crowdsourcing data in disaster relief and mentioned that the credibility of crowdsourcing data and the collaboration in the crowdsourcing tasks is still the major concern. Weaver et al. [18] studied the trust issue in disaster crowdsourcing tasks and proposed several solutions. Yu et al. [19] investigated the efficiency of organizing off-site users efforts in resolving crowdsourcing tasks and proposed a community-based crowdsourcing approach. In this paper, aiming at building an efficient crowdsourcing disaster support platform providing real-time assistance to on-site users, we identify three challenges, viz., how to find relevant off-site users for individual requests, how to organize collaborative work of off-site users, and how to improve the credibility of the crowdsourcing answers. We then design and develop CDSP by addressing these challenges.

3 Platform design

In this section, we describe the design of CDSP by introducing three unique features, viz., identification of relevant off-site users for individual requests, collaboration of off-site users, and improvement of credibility of the crowdsourcing answers. In the following, we first briefly present the SOCIETIES Project within which CDSP is developed and then show system details as well as the prototype screenshots.

3.1 SOCIETIES project overview

Pervasive computing is concerned with connecting to and interacting with devices in the user's environment while social networking is concerned with connecting people. The SOCIETIES project, funded by the European Seventh Framework Programme for Research (FP7), merges social

[6] http://linkeddata.org/.

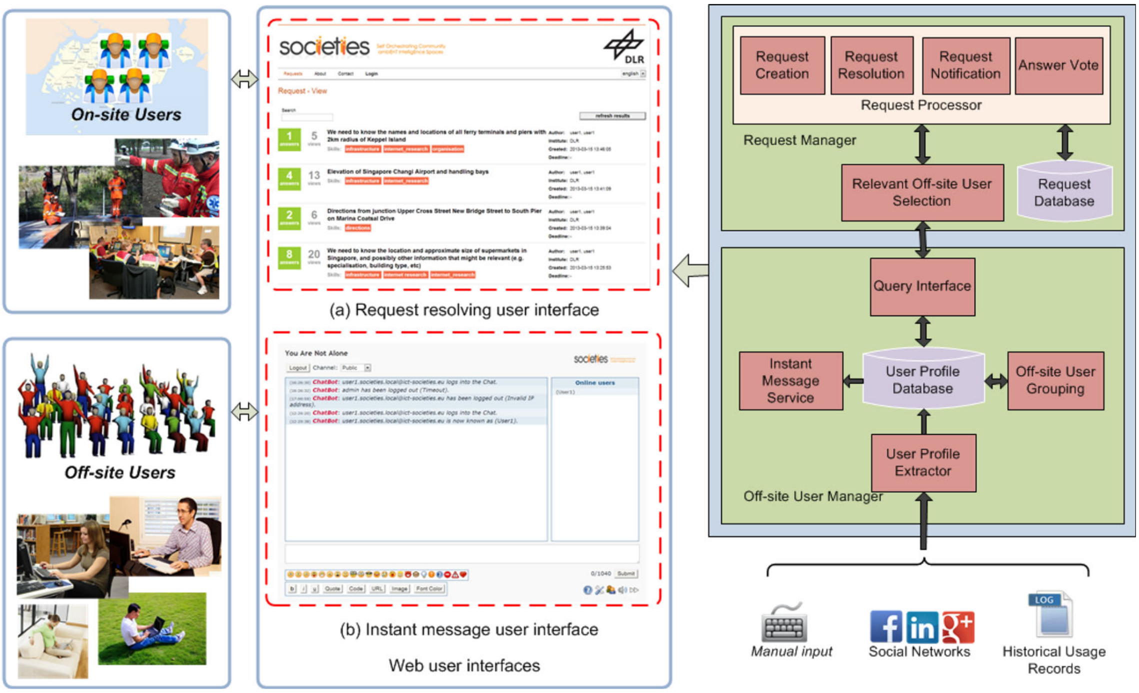


Fig. 1 System overview of CDSP

computing and pervasive computing by designing and implementing a sustainable and personalized platform for mobile social network services. One targeted application domain is crowdsourcing disaster management, which aims at supporting the on-site rescue teams in disaster relief tasks.

3.2 Crowdsourcing disaster support platform

Figure 1 illustrates the system overview of CDSP. The left panel illustrates both on-site and off-site users, as well as the Web user interfaces. The right panel shows the platform components. In the following, we first present the individual components and then describe the specific features of CDSP to resolve the three identified challenges.

3.2.1 Platform components

From the system perspective, CDSP consists of three parts, i.e., the Request Manager and the Off-site User Manager as well as the related Web user interfaces.

The Request Manager is responsible for processing all requests from on-site users. It is composed of five components and a request database. The Request Creation/Request Resolution components take charge of creating new requests and sending back answers from off-site users to on-site users. Request Notification component informs off-site users about the arrival of new requests, which match their expertise. The Answer Vote component is designed to let users vote for the best answer to individual requests, which provides additional information to the on-site users for the decision process. These four components combined are named as the Request Processor in CDSP. The relevant off-site user selection component takes charge of identifying the necessary expertise for a specific request and retrieving the most relevant off-site users information from the Off-site User Manager.

The Off-site User Manager is responsible for processing all off-site volunteer information. It is composed of four components and a user profile database. The Query Interface retrieves the most relevant group of users according to their expertise and sends their information back to the Request Manager. In order to retrieve the relevant off-site users, we need to identify user expertise. To achieve this goal, the User Profile Extractor component can collect user profile information from various data sources, such as user manual input, social network services, as well as their historical usage records of the platform. According to these user profile, the off-site user grouping component clusters off-site users into overlapping groups based on their expertise, which are later used for off-site user information retrieval. In addition, to help off-site users collaborate with each other, the Instant Message Service component provides a Web chat interface.

The Web user interfaces consist of two parts as shown in Fig. 1. The request resolving interface can help users to create new requests, answer requests, vote for an answer to a request and receive new request notification. The instant message user interface provides an on-line chat forum for off-site users to discuss requests, coordinate their efforts, etc. The prototype of CDSP was developed using PHP and Java.

3.2.2 *Specific features*

The design and implementation of CDSP targets on the three challenges, viz., how to find relevant off-site users for individual requests, how to organize collaborative work of off-site users, and how to improve the credibility of the crowdsourcing answers. In the following, we explain how CDSP resolves these problems.

Firstly, in order to help off-site users to put more efforts on resolving requests rather than wasting time in browsing all requests, for a specific request, we select off-site users according to their expertise. When creating new requests, users need to assign one or more expertise (i.e., "skills" in request resolving user interface) tags to the request such as "internet research," "infrastructure," and "navigation". By extracting off-site user profile, we cluster them according to their expertise. Then, the relevant off-site user selection component retrieves a group of users whose expertise is relevant to the requirements of a request. In the implemented prototype, the expertise tags are a finite set of predefined terms (10–20 tags). We let users manually choose them for their profile as well as creating new requests. To retrieve the relevant group of users, we adopt the keyword-matching approach. More sophisticated methods for expertise description and relevant user retrieval, such as using topic models or web ontology languages, can be used for these steps. However, adopting the basic approach can still validate the key idea and functionalities of the platform.

Secondly, in order to collaboratively maximize crowdsourcing efforts, we provide off-site users with the opportunity to work using the "divide-and-conquer" approach. Concretely, for a request that needs collaborative efforts, off-site users can divide the original request into sub-requests and then create them on the platform via the request resolving interface. In such a way, they can work on these sub-requests in paralle, and thus obtain satisfactory answers much more quickly. In addition, they can also have a discussion via the instant message user interface for those requests.

Thirdly, in order to improve the answer credibility, we implement a "crowd voting" feature to let users select the best answer to a request. Such a crowd voting approach has proved to be efficient in obtaining more reliable results in many existing on-line crowdsourcing services, such as Stack Overflow.[7] In our current prototype, we also adopt a user credibility rating mechanism to obtain an individual credibility degree for future extension. Specifically, from long-term usage logs, we can calculate the frequency of the answers voted by others, which implies an individual's credibility degree. Considering such a measurement can enable various new features in the future, such as filtering out low credible off-site users in the relevant user selection process, or building an incentive mechanism to attract more efforts from off-site users.

4 Experiments

To evaluate CDSP, we conducted experiments with three rounds of user trials. We simulated the situation of an earthquake in Singapore and tried to use CDSP to allow off-site users to help on-site users. On-site users involved people working in disaster management domain for many years. Off-site users were recruited volunteers, including colleagues, friends, etc., from different countries. In the following, we first present the user trial setup and then show the analysis results.

4.1 User trial setup

We conducted three user trials in our experiments. The first user trial was held on the March 5, 2013 lasting about two and a half hours. We recruited the participants from the SOCIETIES Project consortium, including one on-site user and four off-site volunteers. All users were located at their respective workplaces. The trial included a preparatory phase (30 min), the actual trial (90 min), and a phase for discussion and review (30 min). The second user trial was held on March 15, 2013. Comparing with the first trial, we first included new sets of disaster scenario requests and deployed a different composition of the off-site volunteers from the SOCIETIES Project consortium. Based on the user feedback from the first trial, we duplicated the system notification feature in the instant message service in order to announce new request/answers and also prepared a comprehensive user guide in this trial. The third user trial was carried out on April 30, 2013. Comparing with the prior trials, we recruited off-site volunteers from outside the project, who had no prior knowledge about the platform before the trial. Moreover, based on the user feedback from the second trial, we added an "off-site volunteer leader" who had participated in the previous trials and been familiar with the platform functionalities. The details about user trial setup are as follows.

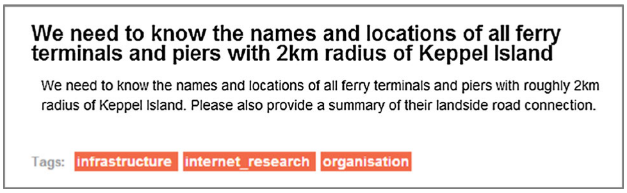
4.1.1 *Participants*

Crowdsourcing disaster management works in a distributed manner, i.e., users are located all over the world and

[7] http://stackoverflow.com.

Table 1 Location of trial participants

User tole	First trial	Second trial	Third trial
On-site user	Germany	Germany	Germany
Off-site user 1	Ireland	France	Ireland
Off-site user 2	Norway	Norway	Ireland
Off-site user 3	France	Ireland	Ireland
Off-site user 4	Greece	Ireland	Germany
Off-site leader	N/A	N/A	France



We need to know the names and locations of all ferry terminals and piers with 2km radius of Keppel Island

We need to know the names and locations of all ferry terminals and piers with roughly 2km radius of Keppel Island. Please also provide a summary of their landside road connection.

Tags: infrastructure internet_research organisation

Fig. 2 An example of requests from on-site users

provide their help via crowdsourcing disaster management platforms. Therefore, to simulate such experimental environments, we recruited the trial participants in Ireland, Greece, France, Norway, and Germany throughout the three trials. In summary, Table 1 shows the location of the participants in the three trials.

Moreover, in our scenario, on-site users initiate requests from the simulated disaster sites. In order to properly simulate the real requests, we assigned this role to a disaster management expert from Germany, who also worked in the SOCIETIES project. In addition, due to the diversity of crowdsourcing users, it may be difficult to reliably achieve high-quality work because some online workers may behave irresponsibly, misunderstand the task, etc., which is a common drawback of crowdsourcing systems [4]. Consequently, in the third user trial, we added an "off-site leader" participant to help resolve requests, which was also suggested by participants in the first and second trials. The "leader" was a participant who was familiar with the platform functionalities and had participated previous trials. The objective of adding this role was to better coordinate off-site users' collaboration such as leading the "divide-and-conquer" process for difficult requests.

4.1.2 Request preparation

Due to the rapidly changing situation of the disaster site, the request from on-site users should be created spontaneously and may be diverse and require various expertise. To simulate such a situation, the disaster management expert continuously created requests during the trial process that might probably be raised by on-site users. The expert also predefined a set of expertise tags which were later assigned to each request and selected by off-site users as part of their profiles. Figure 2 shows a request example.

However, most crowdsourcing disaster management platforms do not have prior information about the expertise of off-site volunteers. Thus, without prior knowledge about the off-site users, defining appropriate expertise tags is a challenging task. A well-defined expertise skill tag set implies that the required expertise is mostly covered by the off-site users' expertise. In our experiments, we continuously considered those tags after each trial and made improvements for the next trials. Figure 3 illustrates the expertise tag clouds for the three trials.

4.1.3 Platform configuration

During the three user trials, we continuously considered the feedback from the users and made improvements to CDSP. In the first trial, off-site users were notified for new requests via the request resolving user interface. However, as users spent most of their time in the instant message service, they would also prefer to be notified for new requests/answers by system notification messages via the instant message user interface. Thus, we added these required features for the second and third trials.

4.1.4 Data collection

We collected user logs in CDSP, which included two parts, viz., request resolving records and the instant messages they communicated. From these data, we were able to quantitatively evaluate the efficiency and effectiveness of CDSP in terms of providing crowdsourcing support to on-site users. Furthermore, in order to evaluate the usability of CDSP, we interviewed the participants to get their direct feedback and conducted a survey for System Usability Scale after each trial. In our study, we adopted the System Usability Scale developed by Brooke [2] in 1996, which had been widely adopted by both academia and industry. This tool contains a ten-item scale questionnaire that provides a global assessment of usability.

4.1.5 Results analysis

Based on the data collected in the user trials, we conducted an analysis from two perspectives. Concretely, we first quantitatively analyzed the user logs including both requests/answers and instant messages. We then studied the usability of CDSP via the SUS survey results.

4.1.6 Analysis of user logs

The user logs provide an important mechanism to understand how the platform works and whether it achieves the

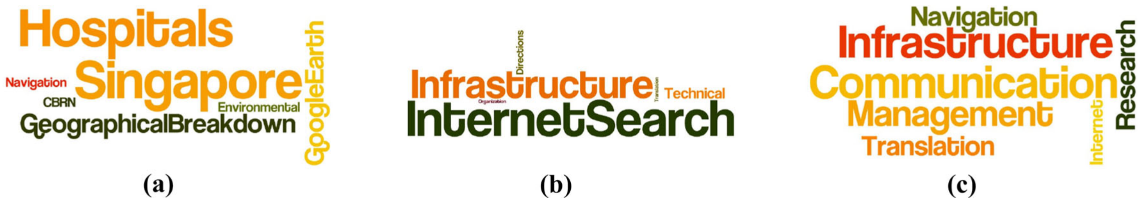


Fig. 3 Expertise tag clouds of requests (larger font size implies higher frequency, and vice versa). **a** First trial. **b** Second trial. **c** Third trial

Table 2 Statistics of user logs

Logs	Metric	First trial	Second trial	Third trial
Requests/ answers	No. of accomplished requests	7	15	19
	No. of created sub-requests	3	3	3
	No. of tags per request	2.75	2.08	1.42
	No. of tags per user	1.75	2	6.2
	Expertise coverage rate (%)	50	83	100
	No. of answer votes by on-site users	7	3	17
	No. of answer votes by off-site users	2	1	0
Instant messages	No. of total messages	108	133	198
	No. of messages from on-site users	35	28	34
	No. of messages from off-site users	73	55	97
	No. of system notification messages	N/A	50	67

design objectives. Thus, we conducted a statistical analysis on the requests/answers logs and instant message logs. Table 2 shows the statistics from these user logs. In general, within the same duration (90 min), the number of accomplished requests increased from the first trial to the third trial and achieved satisfactory results in the second and third trials. Concretely, due to the lack of a comprehensive user guide in the first user trial, four off-site users struggled to understand the scenario, as well as the platform functionalities, which led to the fact that only 7 requests were resolved. In the second trial, four off-site users were well informed by the user guide and resolved 15 requests in total. In the third trial, with an extra leader, 19 requests were resolved by five off-site users. In the last two trials, off-site users efficiently provided help to resolve requests and achieved consistently good performance in terms of the number of accomplished requests per user. In the following, we show the detailed analysis according to the three identified challenges.

Firstly, in order to help off-site users efficiently find relevant requests, we implemented an off-site user selection feature through a set of predefined expertise tags. In Sect. 4.1.2, we discussed the expertise tags that were used to annotate requests as well as characterize user profiles in CDSP. A good tag set should be comprehensive, from which user profiles can maximize the coverage of the required expertise tags in requests. As we continuously improved the tag set throughout the three trials, we illustrate the expertise tag clouds in user profiles for three trials in Fig. 4. By comparing with Fig. 3, which depicts expertise tag clouds for requests, we calculated the expertise coverage rate by off-site users, which is defined as the percentage of the expertise tags required by the requests that can be covered by the expertise tags from off-site users' profiles. As shown in Table 2, the expertise coverage rate increases from the first trial to the third trials and finally reaches 100 % in the third trial. We also observe that the requests use less tags and off-site users add more tags in their profile from the first trial to the third trial. For each trial, we analyze the tags selected by off-site users and user feedback, in order to adjust the tag set more precisely and concisely. For example, in the first trial, the tag "CBRN" (i.e., Chemical, Biological, Radiological, and Nuclear) had not been selected by any users because it is a professional term that off-site volunteers did not associate with. As such, we removed it for the rest of the trials. In addition, due to the limited number of participants in the user trials, the bias of participants existed such as these profession domains, which may also have influenced the expertise tag selection. We have a further discussion about this is Sect. 5.

Secondly, in order to help off-site users collaborate with each other, we implemented the "divide-and-conquer" mechanism in CDSP as well as an instant message service. As shown in Table 2, users had an intensive discussion for each of the trials. We observed clearly an increase in the discussion traffic over the three trials, which correlates with the increasing number of accomplished requests. However, the number of messages from both on-site and off-site users does not exactly follow this increase. By investigating into the message contents, we found that almost half of the messages in the first trial were "noise" messages, which discussed the platform features and the trial scenarios and planning. Consequently, without the consideration of those "noise" messages, the number of messages from both on-site and off-site users also increases over the

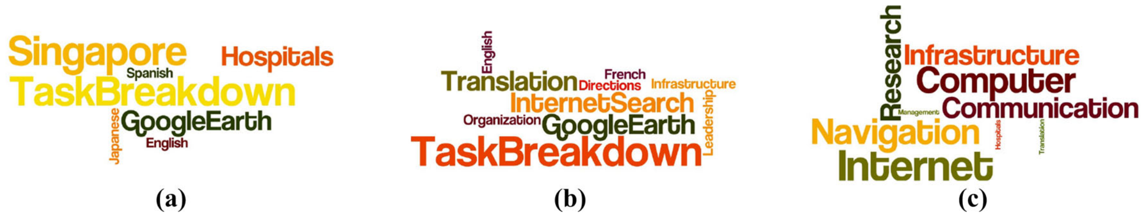


Fig. 4 Expertise tag clouds of user profile (larger font size implies higher frequency, and vice versa). **a** First trial. **b** Second trial. **c** Third trial

three trials. Moreover, the number of sub-requests shows a constant number throughout the three trials. This can probably be explained by two reasons. On the one hand, most of the requests can be resolved individually and therefore do not require collaborative efforts. On the other hand, users are not sufficiently confident to create sub-requests. Although an off-site leader was assigned in the third trial, user feedback shows that he/she was not active enough in terms of leading the "divide-and-conquer" process.

Thirdly, to increase the credibility of the crowdsourcing answers, we implemented an answer voting feature, so that users could vote for the best answer to a specific request. However, an interesting observation was that very limited voting was performed by the off-site users over the three trials. In contrast, the on-site user confirmed most of the answers. This was probably due to the fact that off-site users tried to resolve the maximum number of requests in the limited trial time. At the beginning of each trial, they usually organized themselves via instant messages to tackle different requests. Afterward, when a user finished the answer to a request, he/she usually chose to work on a new task rather than checking the answers of others.

4.1.7 Analysis of the System Usability Scale survey

The System Usability Scale developed by Brooke is a Likert's Scale [10], where a statement is made and respondents are supposed to indicate the degree of agreement with the statement. The SUS consists of ten statements, of which odd-numbered statements are worded positively and even-numbered statements are worded negatively. To use the SUS, participants should indicate their agreement with each statement using a five-point scale from 1 (anchored with "Strongly disagree") to 5 (anchored with "Strongly agree"). Afterward, each statement's score contribution is determined, which ranges from 0 to 4. Concretely, for positively worded statements (1, 3, 5, 7, and 9), the score contribution is the scale position minus 1. For negatively worded statements (2, 4, 6, 8, and 10), it is 5 minus the scale position. Finally, SUS yields a single score representing the overall usability, which is calculated by multiplying the sum of the statement score contributions by 2.5. In addition, Lewis et al. [9] conducted factor analysis on the SUS statement and then defined two dimensions, i.e., learnability and usability. According to their analysis, the learnability dimension includes the statement 4 and 10, while the usability dimension includes the statements 1, 2, 3, 5, 6, 7, 8, and 9. Please refer to [2] and [9] for more details.

Table 3 System Usability Scale scores (higher score implies better performance)

SUS Statements	First trial	Second trial	Third trial
S1: I think that I would like to use this system frequently	2.20	2.67	2.20
S2: I found the system unnecessarily complex	1.67	1.67	2.00
S3: I thought the system was easy to use	2.17	2.67	2.60
S4: I think that I would need the support of a technical person to be able to use this system	2.80	3.00	3.80
S5: I found the various functions in this system were well integrated	1.40	1.00	2.20
S6: I thought there was too much inconsistency in this system	1.80	1.67	3.00
S7: I would imagine that most people would learn to use this system very quickly	2.60	2.33	2.80
S8: I found the system very cumbersome to use	1.20	1.00	2.60
S9: I felt very confident using the system	1.20	2.67	2.40
S10: I need to learn a lot of things before I could get going with this system	2.17	2.00	3.20
Learnability dimension (S4 and S7)	2.49	2.50	3.50
Usability dimension (other 8 statements)	1.78	1.96	2.48
Overall SUS score	48.03	51.67	67.00

We conducted a SUS survey after each user trial for all participants. Table 3 shows the SUS statements as well as the average scores. Higher scores imply better usability and learnability. The overall SUS score increased over the three trials and reached 67 in the third trial. According to the study of Bangor et al. [1] on adjective ratings (i.e, worst imaginable, awful, poor, OK, good, excellent, and best imaginable) and SUS scores (from 0 to 100), CDSP achieved a "good" SUS rating.

We also observed an increase in both of the usability and learnability dimensions over the three trials. Concretely, the learnability dimension achieved high scores throughout the three trials and even reached 3.50 in the third trial, which suggests that CDSP can efficiently enable users to learn how to use it. Although the usability dimension had relatively low scores for the first and second trials, it achieved satisfactory scores in the third trial. In addition, we also observed that the statements 2, 5, 6 and 8 had relatively low scores in first and second trials, but reached high scores in the third trial. It should be noted that we did not add any new features to the platform for the third trial. From the investigation of user feedback, we found that in the first and second trials, users expressed a strong demand for better user guides and a role of a "leader" to coordinate the request resolution. Both of these were available in the third trial, which may probably be the reason that the usability dimension reached high scores in the 3rd trial.

5 Discussion

Through the three user trials and SUS survey, we have shown that CDSP can effectively and efficiently provide on-site users with real-time assistance and achieve good usability. However, we still have some further consideration about the platform design and user trials.

Firstly, from the platform design perspective, the integration of the platform features and the interoperability of CDSP with the existing disaster management platform need to be further improved. From the user feedback and SUS survey results, participants suggested that the request resolving user interface and instant message user interface could be better integrated and merged, because it is sometimes inconvenient to switch between the two user interfaces. Furthermore, a common shortfall of crowdsourcing disaster management platforms is the lack of interoperability. In other words, most of the existing applications do not provide a common mechanism for collaboration and coordination between them [6], which usually causes inefficiency in disaster information sharing and collaborative disaster relief. Thus, it is necessary to specifically design a common mechanism and communication interface for collaboration between disparate disaster management platforms.

Secondly, a larger-scale user trial would provide a more complete evaluation of CDSP. In the first and second trials, participants were recruited from SOCIETIES project consortium, who already have some prior knowledge about the crowdsourcing platform and also disaster management platforms in general. In the third trial, we recruited participants outside the SOCIETIES project, most of whom were our colleagues not getting involved in the SOCIETIES project. It should be noted that those participants all work in the Information Technology (IT)-related domain. Consequently, such a professional bias of the trial participants may not fully fit the fact of diversity of off-site volunteers and may cause bias in expertise tag selection, request preparation, etc. Moreover, existing crowdsourcing disaster management platforms, such as Ushahidi, usually have hundreds of thousands of off-site users. Therefore, a large-scale user trial would be preferable for further evaluation of CDSP.

6 Conclusion

With the increasing popularity of crowdsourcing applications, disaster management also benefits from the power of crowdsourcing in disaster data collecting and analysis. In this paper, we focus specifically on providing on-site users with real-time remote assistance in disaster relief tasks by leveraging crowdsourcing power from off-site volunteers. By analyzing such scenarios, we build CDSP, a crowdsourcing disaster support platform, by introducing three unique features. Firstly, we implement an off-site user selection mechanism to efficiently perform request notification according to user expertise. Secondly, we provide off-site users with the opportunity to collaborate with each other when tackling difficult requests. Thirdly, we implement a crowd voting feature, aiming at improving the crowdsourcing answer credibility. The experiments with the three rounds of user trials shows the effectiveness and efficiency of CDSP. The SUS survey also shows CDSP achieves the good usability.

In the future, we plan to broaden CDSP by adding new features, such as designing concise and efficient user interfaces, integrating with other disaster management platforms, and exploring individual credibility degrees. Furthermore, as suggested in [19], social and context information are also important to improve off-site volunteers' efficiency. Therefore, we could also consider social influence and user context in the relevant user selection process. In addition, based on the analysis of the three user trials, we also plan to conduct a large-scale user trial by including more participants with the consideration of diversity of their professions, ages, locations, etc.

Acknowledgments This work is supported by the EU FP7 Project SOCIETIES (No. 257493). The authors would like to express their great appreciation to Dr. Jacqueline Floch, Dr. Michael Angermann, Yiorgos Bouloudis, Bernhard Perun, Gunther Berthold, as well as all other colleagues and trial participants, for their help in the preparation and execution of the user trials. The authors would also like to thank the editors and reviewers for the valuable comments and suggestions.

References

1. Bangor A, Kortum P, Miller J (2009) Determining what individual sus scores mean: adding an adjective rating scale. J Usability Stud 4(3):114–123
2. Brooke J (1996) Sus-a quick and dirty usability scale. Usability Eval Ind 189:194
3. Chu ETH, Chen YL, Lin JY, Liu JW (2012) Crowdsourcing support system for disaster surveillance and response. In: Proceedings of WPMC. IEEE, pp 21–25
4. Dow S, Kulkarni A, Klemmer S, Hartmann B (2012) Shepherding the crowd yields better work. In: Proceedings of CSCW. ACM, pp 1013–1022
5. Frassl M, Lichtenstern M, Khider M, Angermann M (2010) Developing a system for information management in disaster relief-methodology and requirements. In: Proceesings of ISCRAM
6. Gao H, Barbier G, Goolsby R (2011) Harnessing the crowdsourcing power of social media for disaster relief. IEEE Intell Syst 26(3):10–14
7. Goodchild MF (2007) Citizens as sensors: the world of volunteered geography. GeoJournal 69(4):211–221
8. Goodchild MF, Glennon JA (2010) Crowdsourcing geographic information for disaster response: a research frontier. Int J Digit Earth 3(3):231–241
9. Lewis JR, Sauro J (2009) The factor structure of the system usability scale. In: Human centered design. Springer, Berlin, pp 94–103
10. Likert R (1932) A technique for the measurement of attitudes. Arch Psychol 22(140):1–55
11. Mizushima K, Sugihara T, Ikawa Y (2012) Knowledge management in a volunteer community at the time of disaster. In: Proceedings of PICMET. IEEE, pp 2274–2282
12. Morrow N, Mock N, Papendieck A, Kocmich N (2011) Independent evaluation of the ushahidi haiti project. Dev Inf Syst Int 8:1–36
13. Ortmann J, Limbu M, Wang D, Kauppinen T (2011) Crowdsourcing linked open data for disaster management. In: Proceedings of the Terra Cognita workshop on foundations, technologies and applications of the geospatial web, pp 11–22
14. Palmer N, Kemp R, Kielmann T, Bal H (2012) Raven: using smartphones for collaborative disaster data collection. In: Proceedings of ISCRAM
15. Roche S, Propeck-Zimmermann E, Mericskay B (2013) Geoweb and crisis management: Issues and perspectives of volunteered geographic information. GeoJournal 78(1):21–40
16. Schulz A, Paulheim H, Probst F (2012) Crisis information management in the web 3.0 age. In: Proceedings of ISCRAM
17. Vivacqua AS, Borges MR (2012) Taking advantage of collective knowledge in emergency response systems. J Netw Comput Appl 35(1):189–198
18. Weaver AC, Boyle JP, Besaleva LI (2012) Applications and trust issues when crowdsourcing a crisis. In: Proceesings of ICCCN. IEEE, pp 1–5
19. Yu Z, Zhang D, Yang D, Chen G (2012) Selecting the best solvers: toward community based crowdsourcing for disaster management. In: Proceedings of APSCC. IEEE, pp 271–277