DOOR: A Data Model for Crowdsourcing with Application to Emergency Response

To Tu Cuong[1(✉)], Paras Mehta[1], and Agnès Voisard[1,2]

[1] Department of Mathematics and Computer Science, Freie Universität Berlin, Berlin, Germany
{to.cuong,paras.mehta,agnes.voisard}@fu-berlin.de
[2] Fraunhofer FOKUS, Berlin, Germany

Abstract. Crowdsourcing allows us to employ collective human intelligence and resources in completing tasks in a wide variety of domains, such as mapping, translation, emergency response, and even fund raising. It first involves identification of a problem that can be solved using crowdsourcing and then its decomposition into tasks that workers can finish in a timely manner. Worker engagement analysis and data quality analysis are done afterwards. Such analysis activities are not supported by current platforms and are done in an ad-hoc fashion leading to duplicate efforts. As a first step towards realizing such analysis mechanisms, we propose a *D*ata m*O*del for cr*O*wdsou*R*cing (DOOR), which is based on a fuzzy Entity-Relationship model in order to capture the uncertainty that is inherent in any crowdsourcing process. To illustrate its application, we have chosen the problem of collection of data about incidents for emergency response.

Keywords: Data model · Emergency response

1 Introduction

The collective power of the crowd is being utilized today in a wide variety of applications, ranging from digitizing text [18], through image tagging [17], to mapping the world [13]. A considerable amount of research has already been done to review and categorize existing crowdsourcing systems and technologies [5,8]. All these systems require significant upfront investment into designing tasks and dealing with technicalities related to tasks. In other words, the infrastructures that support crowdsourcing are limited [2]. As a consequence, although programming frameworks for crowdsourcing are coming up [1,10,14], developers often end up spending a great deal of effort in dealing with operational aspects, rather than in designing the crowdsourcing tasks. Currently, the common workflow for experimenters is that they design the experiment, run it, collect the results, and finally analyze them. As a result, experimenters develop a set of tools around a crowdsourcing platform, e.g. Amazon Mechanical Turk [3], to ensure that experiments are successful. While this is fine for "one time"

R. Giaffreda et al. (Eds.): IoT360 2014, Part II, LNICST 151, pp. 265–270, 2015.
DOI: 10.1007/978-3-319-19743-2_37

experiments, it is certainly not suitable for experiments that run on a continued basis. Moreover, experimenters may want to conduct analysis while crowdsourcing is being done. The reason for doing this is that if experimenters realize that the data quality is low, they can increase the payment or change the description of tasks. As a first step towards addressing the aforementioned challenges, we propose a *D*ata *M*odel for cr*O*wdsou*R*cing *(DOOR)*.

The proposed model is generic enough to be applied to different real-world crowdsourcing tasks, particularly in the geospatial domain. One of these is response to the occurrence of incidents, such as fires, traffic accidents, and crimes in urban areas. Such incidents can happen suddenly, without much warning, and do not leave much time to prepare and act. Our aim is to mitigate the impact of such events by detecting them early and by informing the affected population on time with relevant, actionable information. Crowdsourcing can assist in different tasks, starting from collection of data about incidents, through delegating tasks to volunteers, to receiving feedback. As the number of people equipped with smartphones and wearable devices grow, citizens become increasingly capable of capturing snapshots of their surroundings through the host of sensors embedded in these devices. Thus, crowdsourcing assumes a progressively prominent role in emergency response. In this paper, we describe the use of *DOOR* for the collection of geotagged reports about incidents.

The remainder of this paper is organized as follows. Sections 2 and 3 describe the workflow and the entities involved in crowdsourcing, respectively. Section 4 presents a brief discussion and concludes the paper.

2 Crowdsourcing Process

In crowdsourcing, we have two main parties: *requesters* and *workers*. A requester has a problem and wants to solve it with the help of the crowd. An example could be the image tagging problem, where a requester would like to have tags for her images. These tags help build image search services. Due to the nature of crowdsourcing, a problem needs to be divided into subtasks that can be finished quickly, e.g., in less than a minute. Some problems (like image labeling) can be divided easily. For others, this can be tricky, e.g., composing a poem. There have been researches into automatically decomposing a problem using recursive crowdsourcing [9], i.e., crowdsourcing the decomposition of a problem itself. This line of research is orthogonal to our research. Here, we assume the problem decomposition is already provided.

Once the problem is divided into subproblems, these subproblems are assigned to multiple users to collect their contributions. These contributions are then aggregated into a "result" for the original problem. The mechanism to aggregate answers (contributions) from workers into the result is called a *rewarding model*. The most popular rewarding model is majority voting where answers that receive the majority of votes are deemed correct and its corresponding workers are rewarded financially or through other means.

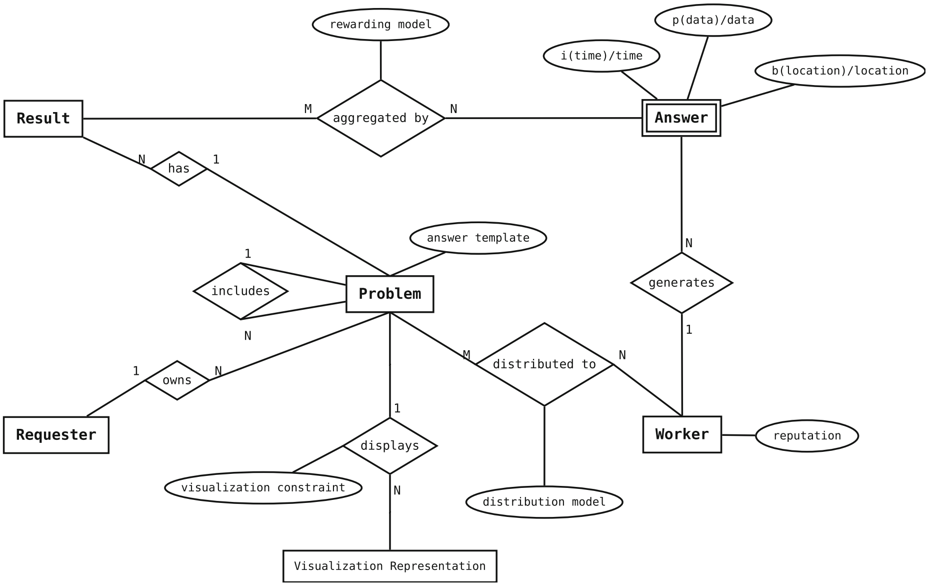

Fig. 1. The DOOR model

We also note that in crowdsourcing, it can be either requesters or workers who initiate the crowdsourcing process. In our previous example, i.e. image tagging, a requester is the active side. On the other hand, Volunteered Geographic Information (VGI) [6] or user generated geospatial content, such as geotagged tweets, is also a form of crowdsourced data which can be processed to produce useful information (e.g., epidemic predictions). In these cases, the workers (who tweet) are the active entities.

3 Crowdsourcing Data Model

In Fig. 1, we use an Entity-Relationship diagram to depict the *DOOR* model. In this model, we have 5 main entities, namely *Requester*, *Worker*, *Answer*, *Result*, and *Problem*. The *Requester* and *Worker* entities in the model represent the task provider and the task worker in the crowdsourcing process, respectively, while the *Problem* and the *Result* entities model the task and the end result, respectively. The recursive nature of problems (or tasks) are captured in our model as the relationship *Includes* between *Problems* themselves. They are distributed to *Workers* using *distribution model* which relies on statistics, such as workers *reputation* and problem type. When a problem is presented to a worker, she can choose an answer and possibly provide ancillary data, such as location, photo, or any input generated by her mobile's sensors. These contributions are captured by the weak entity *Answer*. Since it is an important entity, we describe its attributes in details as follows.

- *i(time)/time: time* represents the instant a worker makes a contribution, whereas $i(time)$ is an interval defined as $[time - \epsilon, time + \epsilon]$ where ϵ is application dependent. This interval is used to specify in which period a worker's contribution is considered "relevant". For example, when a worker indicates that there is an traffic time at 8 am at a specific location, the interval could be $[8 - 0.5, 8 + 0.5]$ which means the traffic jam could have happened between 7:30 am and 8:30 am.
- *b(location)/location: location* represents the location of a worker when she contributes, while $b(location)$ represents a neighborhood around the location (e.g., a circle or a polygon). Similar to the time interval, the contribution is considered "relevant" within this neighborhood.
- *p(data)/data:* The data contributed by workers is inherently noisy, which is why we need to model its uncertainty. $p(data)$ is the probability of that *data* being accepted. How this probability is calculated is application dependent. For instance, we can employ the *reputation* of a worker as this probability. One way to compute a worker's *reputation* is to divide the number of her correct contributions by the total number of contributions made by her.

Answers once collected from users can then be aggregated using a chosen *rewarding model*. Another important aspect in crowdsourcing is how to present a problem to a worker. This is modeled by the *answer template* attribute of the *Problem* entity and by the relationship *Displays* with the *Visualization Representation* entity. An answer template could be an HTML (HyperText Markup Language) file which is used to generate a user interface for workers. Since workers can take part in a crowdsourcing process using different devices, the answer template needs to be adapted based on the *visualization constraint*s. For example, CSS (Cascading Style Sheets) can be used to display an HTML file properly on different devices.

In the scope of collection of data for emergency response, crowdsourced incident reporting can help in the early detection of the incident occurrence and in the reinforcement and validation of data collected through hardware infrastructures, like sensor networks. Moreover, updates from the field from citizens can assist in maintaining constant situational awareness. For example, in the event of a fire outbreak, people on the site can report on the extent of the fire perimeter with pictures, video footages, estimates of the damage, and the number of occupants in the affected area. This can improve the response time of the authorities, reduce or help prioritize their workload, and increase citizen participation and transparency in crisis response. It also allows the people themselves in staying updated by receiving regular information about the incident from other members of the community.

We now focus on a specific scenario of a fire in a building complex in an urban area which is detected by the fire department through smoke alarms and calls to the emergency services. The fire department sends out notifications to the people to inform and advise them, and to ask the people at the scene for updates. Our proposed model can be applied to this scenario where people respond back with information about the fire, as depicted in Fig. 2.

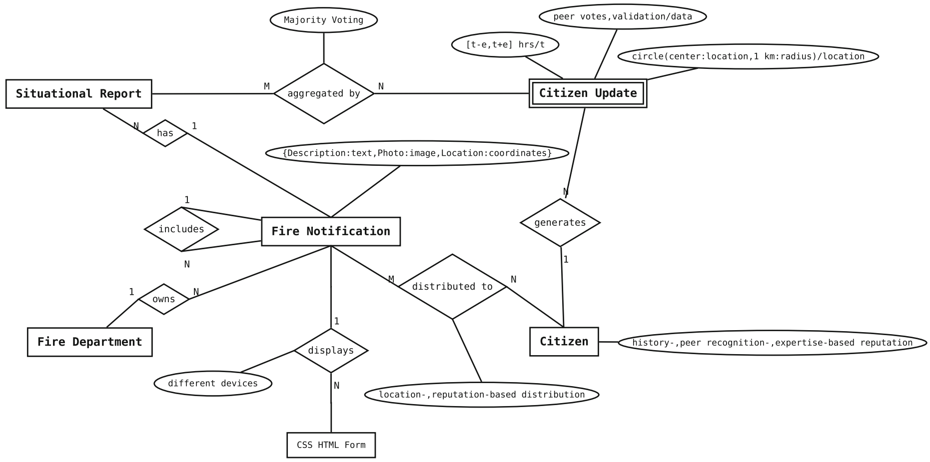

Fig. 2. The DOOR model in emergency response

4 Discussion and Conclusions

The potential of crowdsourcing in aiding emergency response efforts has been widely discussed [4,5,7,12,15]. For gathering field data and maintaining constant situational awareness, sharing data using smartphones can be employed [11, 19]. Platforms, such as Ushahidi [16], have proven useful time and again in situations of crises. All of these efforts share a common workflow. It first involves identification of a problem that can be solved using crowdsourcing and then its decomposition into tasks that workers can finish in a timely manner. Job distribution, worker engagement analysis, and data quality analysis are done afterwards. Such analysis activities are not supported by current platforms and are done in an ad-hoc fashion leading to duplicate efforts. As a first step towards realizing such analyses, in this paper, we proposed the DOOR data model. The model was designed to cope with data uncertainty with a bias towards spatial and temporal data. Moreover, in order to illustrate that DOOR is generic enough to be used in real-world applications, we instantiated it in a disaster management scenario. A software prototype is being developed at our group using DOOR.

Acknowledgements. This research is carried out in the framework of the GEOCROWD project funded by the European Commission.

References

1. Ahmad, S., Battle, A., Malkani, Z., Kamvar, S.: The jabberwocky programming environment for structured social computing. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, pp. 53–64. ACM (2011)

2. Alonso, O.: Perspectives on infrastructure for crowdsourcing. In: Crowdsourcing for Search and Data Mining (CSDM 2011), p. 7 (2011)
3. amazon.com: Amazon Mechanical Turk. https://www.mturk.com/. Accessed 24 April 2014
4. Butler, D.: Crowdsourcing goes mainstream in typhoon response. Nature (2013)
5. Fuchs-Kittowski, F., Faust, D.: Architecture of mobile crowdsourcing systems. In: Baloian, N., Burstein, F., Ogata, H., Santoro, F., Zurita, G. (eds.) CRIWG 2014. LNCS, vol. 8658, pp. 121–136. Springer, Heidelberg (2014)
6. Goodchild, M.F.: Citizens as sensors: the world of volunteered geography. GeoJournal **69**(4), 211–221 (2007)
7. Goodchild, M.F., Glennon, J.A.: Crowdsourcing geographic information for disaster response: a research frontier. Int. J. Digit. Earth **3**(3), 231–241 (2010)
8. Heipke, C.: Crowdsourcing geospatial data. ISPRS J. Photogrammetry Remote Sens. **65**(6), 550–557 (2010)
9. Kulkarni, A.P., Can, M., Hartmann, B.: Turkomatic: automatic recursive task and workflow design for mechanical turk. In: CHI 2011 Extended Abstracts on Human Factors in Computing Systems, pp. 2053–2058. ACM (2011)
10. Little, G., Chilton, L.B., Goldman, M., Miller, R.C.: Turkit: human computation algorithms on mechanical turk. In: Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology, pp. 57–66. ACM (2010)
11. Mehta, P., Müller, S., Voisard, A.: Movesafe: A framework for transportation mode-based targeted alerting in disaster response. In: Proceedings of the Second ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information, GEOCROWD 2013, pp. 15–22. ACM, New York (2013). http://doi.acm.org/10.1145/2534732.2534735
12. Munro, R.: Crowdsourced translation for emergency response in Haiti: the global collaboration of local knowledge. In: AMTA Workshop on Collaborative Crowdsourcing for Translation (2010)
13. OpenStreetMap Community: OpenStreetMap. http://www.openstreetmap.org/. Accessed 04 April 2014
14. pybossa.com: Pybossa. https://pybossa.com/. Accessed 01 June 2014
15. TechCrunch: Help Me Help uses crowdsourcing to make disaster response more efficient. http://techcrunch.com/2013/07/04/. Accessed 24 March 2014
16. Ushahidi Inc.: Ushahidi. http://ushahidi.com/. Accessed 04 April 2014
17. Von Ahn, L., Dabbish, L.: Labeling images with a computer game. In: Proceedings of the SIGCHI Conference on Human Factors In Computing Systems, pp. 319–326. ACM (2004)
18. Von Ahn, L., Maurer, B., McMillen, C., Abraham, D., Blum, M.: recaptcha: Human-based character recognition via web security measures. Science **321**(5895), 1465–1468 (2008)
19. Zheng, L., Shen, C., Tang, L., Li, T., Luis, S., Chen, S.C.: Applying data mining techniques to address disaster information management challenges on mobile devices. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2011, pp. 283–291. ACM, New York (2011). http://doi.acm.org/10.1145/2020408.2020457