Request-Driven Social Computing: Towards Next Generation Crowdsensing Systems

Mathijs de Weerdt Virginia Dignum M. Birna van Riemsdijk Martijn Warnier

Delft University of Technology, The Netherlands
{m.m.deweerdt,v.m.dignum,m.b.vanriemsdijk,m.e.warnier}@tudelft.nl

Abstract. In this position paper we first argue that increasingly, crowdsourcing and mobile computing are being used to collect, analyse, and share data in domains such as environment sensing and transportation. However, in current systems users can only be a sensing data contributor as part of the crowd, or they can pose a single type of request to the crowd. This means that only small, independent, and homogeneous forms of interaction can be accommodated. Moreover, usually the crowd has no control over the way their resources are used. We argue that current crowdsourcing systems, and in particular, crowdsensing systems do not sufficiently respect the autonomy of users regarding sharing of data and resources (contribution autonomy), nor regarding retrieval of data and information (request autonomy).
The major goal is therefore to build systems that facilitate more flexible interaction, while respecting users' autonomy both regarding control over their resources, as well as control over the types of requests users can pose. Next generation crowdsensing systems should thus carefully balance interdependency with autonomy.

1 Introduction

"These are busy days. At work my boss is pressing me to finish the report, at home I'm finishing the new roof of our garden shed, and tonight we're expected at my mom's birthday. After lunch I quickly sneak out the office to get her a present, when (of course) it starts raining. I'm worried about the unfinished rooftop; if more rain falls, I should ask a friend to check whether the temporary cover is attached correctly. I quickly check rainfall near home on my mobile. It seems that there are a few showers scattered around, but there is no detailed information on whether it is raining in my own street, nor how much has fallen so far."

The story used to end here. However, nowadays sensing devices exist which can be attached to (e.g.) mobile phones, and observations can be shared via dedicated services. Typically, crowdsourcing approaches are used to (rapidly) mobilise large numbers of people to accomplish tasks on a global scale. This opens a world of opportunities (the first of which are already visible), but also new types of problems and challenges: Why would anyone share? It may cost time and probably battery power. How can privacy be achieved for users where data may contain sensitive (but relevant) information such as users' locations? How can one be ensured that the data being shared is protected and secure? Is it efficient to collect everything? If not, can data collection and processing be customised to specific user requests? How can users set conditions on the use of their resources, and how can compliance be monitored? What happens with the data being collected? Who owns it, where is it stored?

In current systems, these user concerns are only addressed to a limited extent. In this paper we first discuss such existing crowdsourcing systems and derive two major concerns regarding users' autonomy in these systems (Section 2). Next we discuss the concrete challenges that arise when attempting to address these concerns (Section 3). We argue that the field of autonomous agents and multi-agent systems is well-positioned to address these challenges, since establishing a balance between interdependency and autonomy in a way that creates value for all stakeholders is at the heart of research in this area. We conclude the paper in Section 4.

2 Requirements for the Next Generation

Crowdsourcing is "the practice of obtaining needed services, ideas, or content by soliciting contributions from a large group of people and especially from the online community rather than from traditional employees or suppliers".[1] In the traditional sense of the term, an organisation or individual puts out a request for contributions (see, e.g., [15]), typically small, independent, and homogeneous tasks that are manually performed by the crowd. Amazon's Mechanical Turk is a typical example of a platform that facilitates requesters to post tasks, and providers to find and address tasks, but other systems exist where users are fulfilling both roles, such as DuoLingo,[2] and JigSaw.[3]

With the increasing pervasiveness of mobile computing devices, a novel type of crowdsourcing is emerging called *mobile crowdsensing* [13]. In such applications, people use mobile devices such as smartphones to collect and share information to measure and map phenomena of common interest. Contributions can be provided manually through the mobile device (participatory sensing), or automatically through sensors connected to the mobile device (opportunistic sensing) [16].

Mobile crowdsensing applications have been developed for measuring phenomena in different domains, including applications for environmental and infrastructure phenomena. For example, smartphone apps for participatory crowdsourcing the weather (e.g., Minutely and Weathermob) allow citizens to contribute data about the weather in their location. This can give more accurate and higher resolution information, since weather phenomena have very high spatial and temporal variability that cannot always be measured accurately with traditional sensor technologies. An example of an opportunistic sensing platform is Common Sense, which is being developed for monitoring air quality by connecting sensors for measuring pollutants to mobile phones [11].

In the traffic domain, real-time information about the current traffic situation obtained through mobile crowdsensing is very valuable in densely populated areas where traffic congestion costs time and money.[4] For example, the TomTomLive service offered by GPS company TomTom displays information about traffic jams and can adapt

---

[1] `http://www.merriam-webster.com/dictionary/crowdsourcing`
[2] `http://www.duolingo.com/`
[3] `http://www.jigsaw.com/`
[4] The costs of congestion in urban areas in 2011 in the US were estimated to $121 billion [20].

its route suggestion accordingly. The data to achieve this is obtained from drivers who in order to use the service are required to dispatch their own location information to TomTom as they are driving. A (non-commercial) mobile app called Waze is a community-based traffic and navigation app that provides a similar service.[5]

All these crowdsensing systems provide powerful functionality by using data gathered by the crowd to provide (real-time) information services to users. However, in these applications the crowd has very little control over the way their resources are used. Especially in opportunistic sensing such as in the traffic examples the user has only two options: opt-in and continuously share data whenever the user is driving the car, or opt-out and never share data in which case the system's services cannot be used. Moreover, in current crowdsensing applications requesters have no control over the data that is being gathered. For example, if contributors to a weather crowdsourcing app are at a given moment not providing the needed data, the requester has no means to connect and ask them to provide this data.

We thus argue that current crowdsourcing systems, and in particular, crowdsensing systems do not sufficiently respect the autonomy of users regarding sharing of data and resources (*contribution autonomy*), nor regarding retrieval of data and information (*request autonomy*). Contribution autonomy means user control over which data is *shared* when and with whom, and request autonomy similarly means that the user can influence which data is being *gathered* when and by whom. To illustrate our vision of how these systems could facilitate interaction while respecting users' autonomy, let us get back to the scenario from Section 1.

"I usually allow my phone to share the information from the humidity sensor I have connected to it. I'm not worried that my boss (who may also be using this system) will now get to know my location and thus what I'm doing, because I've set it up to share humidity only outside office hours. I've built up sufficient credentials to be able to make specific requests. I therefore decide to put out a request for an accurate measurement of past rainfall and a prediction on future rainfall near my home. After a few minutes I receive information from sources nearby home, as well as from within the area upwind."

This example illustrates how the system respects the user's contribution autonomy by allowing to specify when data from the humidity sensor may be shared, and the user's request autonomy by allowing the user to pose requests that influence which data is gathered and shared. Resources nearby the user's home might not want to share their data continuously due to energy constraints, but upon request they may be willing to do so in return for credits. Traditional crowdsourcing systems typically do not fully exploit the potential outlined above.

3 Design Challenges

In crowdsourcing systems there are typically two user roles: a requesting user, who is interested in some of the data or services offered by the system, and a providing user, who shares some of his or her resources, knowledge, or data. Sometimes a user can fulfil both roles. In the general design of future crowdsourcing systems, in our view an agent

[5] `https://www.waze.com/`

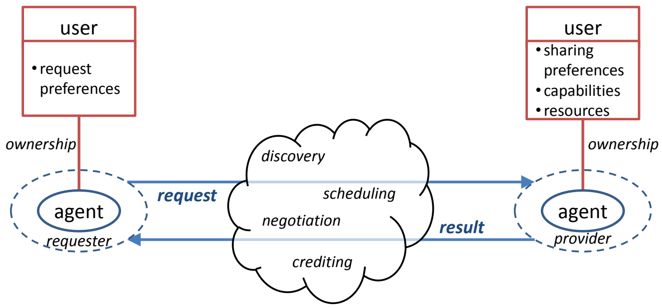

Fig. 1. General design of a next generation crowdsourcing system.

will represent a user and her preferences (see Figure 1). This agent communicates with the user (e.g., via a mobile application) both to have an accurate representation of the user's preferences, as well as to present the results of requests. Respecting contribution and request autonomy has important consequences for the design of the organisation of the process from a user's request to the appropriate result, which we also discuss below. Towards the end of this section we discuss the (dis)advantages of a completely decentralised design versus a cloud-based approach as is commonly done.

3.1 Contribution autonomy

In our vision, end users should have the autonomy to decide what to contribute, how much, with whom and when. This has consequences for preference elicitation, and the design of appropriate incentives.

Preference elicitation and representation. Future crowdsourcing systems respecting contribution autonomy require a transparent manner for end users to define which resources can be used at a particular time. Individuals have preferences and norms related to benefits for self as well as related to benefits for others [19]. In fact, individuals mostly dislike unequal outcomes of interactions (inequity aversion) and will adopt norms that increase their sense of fairness and justice. However, not all individuals have the same norms, and perceptions of a situation and of 'the right thing' may differ substantially. Moreover, it is a challenge to design a normative language that has sufficient expressivity and yet can be understood by users, in particular considering that norms may interact in unforeseen ways during execution of the system. Current work on the representation of preferences, use policies, and norms can provide a basis for resolving this issue [3].

Incentives. The challenge of these crowdsourcing systems is that it can be seen as a commons [18], bringing with it the challenges associated with creating effective commons, managing contributions and requests such that the needed quality and quantity are ensured. These include challenges associated with governance, i.e. the rules to solve conflicts between actors and adopt decision. Although contribution autonomy is based on the voluntary sharing of resources by the crowd, there is an inherent tension between individual rationality and collective functionality that threatens the viability of this type of systems, encouraging free-riding and/or 'sloppy' contributions, especially when contributing is costly. For example in early file-sharing systems without proper incentive mechanisms, only a fraction of the users displayed the desired contributing behaviour, significantly reducing effectiveness of such systems (see e.g. Gnutella [1]). Even though theory suggests that cooperation between unrelated individuals is costly and as a result, without externally generated incentives, rational actors should not cooperate, there is substantial evidence to suggest that cooperation is a common human behaviour, particularly if reciprocation can be expected. Incentives schemes should therefore exploit the capability for reciprocity, e.g. through contractual approaches between (groups of) participants.

For static settings, mechanism design theory provides some results that may be useful, e.g., in settings where payments (rewards) are used. Important open challenges remain in the case requests arrive dynamically, or resources or information become available over time [6], as is typically the case with crowd-sensing systems.

3.2 Request autonomy

Request autonomy means that users have control over the type of requests they can 'ask' the system. At the system level, this requires that the system is flexible enough to continuously reconfigure its resources in such a way that it can meet the various requests of its users, leading to challenges in task decomposition, coordination, online planning and scheduling, and dynamic reorganisation.

Task decomposition. In some cases a request cannot be handled by a single agent. It would then first need to be decomposed into tasks that can. Existing work on task decomposition (see e.g., the overview in [14]) can be of use. However, major challenges remain: which agent should make the decomposition and how does this agent know what the (current) abilities are of others? Can agents act as mediators, forwarding (sub)tasks to others? How then to prevent endless forwarding? When there are multiple possible decompositions, which one should be chosen? Should they be tried one after another? What protocols to use in case some of the responding agents do fulfil their tasks, while others do not?

Coordination. For some requests, agents may need to coordinate their actions: perhaps a picture and the temperature need to be taken around the same time and place, or the counting of birds throughout the country should be done in the same period to prevent birds being counted twice. In general, one can distinguish two types of existing solutions to such coordination problems. *Decoupling* is the approach where coordination is dealt with before the agents start, and in such a way that no further interaction is required. A typical example is setting a fixed time at which to complete a certain activity, or agreeing on a so-called social law [22, 23].

The other approach is where coordination takes place during or sometimes after planning. Existing work on PGP related approaches [17] or plan merging [7] can then be applied. However, the specifics of the distributed and dynamic setting of next generation crowdsensing bring new research questions. How can one be ensured that coordination, planning, and execution are all completed before a deadline? How can a decisions be made (in case coordination turns out to be difficult) to work with another set of agents? How can actions be coordinated of both artificial agents and humans?

Online planning and scheduling. Answering a single request by a single agent can then be done using developments in the field of planning and scheduling. Challenges arise when many such requests appear over time: in the fields of distributed and online task/resource allocation, and online planning and scheduling there are still many issues regarding efficient algorithms and uncertain future demand [8].

Dynamic reorganisation. Related to online planning, users may come and go, introducing or retracting resources, or resources may change because users move around. These are known, but unresolved issues especially relevant to active sensing [24]. Also, recent work in the area of Multi-Agent Organisations can help in the design of next generation crowdsourcing systems [9], specifying structures and rules (e.g. contracts) that facilitate and manage interactions between agents.

3.3 Decentralised systems

In centralised systems users need to send all their data to a server (or nowadays, the cloud) for processing and storage. On the one hand, the advantage of this is that users do not have to process or store any data locally (on their own devices). Such systems are relatively easy to build and maintain. On the other hand, these systems have the disadvantage that users have to trust the server. Some users will be less willing to do this because of privacy concerns. Other issues typically relate to scalability and resilience to failure, but these are mostly solved by current cloud computing techniques [4]. Nevertheless, even clouds have weaknesses and can fail, which is a second major argument for setting up a decentralised system. Moreover, a decentralised system architecture provides the desired autonomy for users to decide at which aggregation level information is shared. However, a decentralised system design where any user can contribute computing and communication resources to the “cloud” brings an additional dimension of complexity to the above mentioned challenges related to trust, privacy, incentives, and planning and scheduling.

Trust. First, the issue of trusting a single responsible party moves in the decentralized situation to trusting a system composed of the resources of many different users. This problem has been well studied in the field of trust and reputation, but no single perfect solution is known to date [21].

Privacy. There are known examples of distributed applications that provide more privacy for their users than centralised designs, such as Freenet [5], an anonymous peer-to-peer file sharing client, and Tor [10], an anonymous communication application. Both applications require users to provide resources: storage for Freenet and (communication) bandwidth for Tor. An additional challenge for crowd-sensing systems is that often data is location related, and users usually do not like to share their own location.

Distributed incentives. As discussed earlier, designing a mechanism to provide the right incentives to users is already a challenging problem, but when this must be realised using a decentralized system, this problem becomes even harder. The field in which this is studied is distributed mechanism design [12], but consists mostly of very hard open problems. A possible alternative approach might be to let go of the view where the whole systems is seen as one (distributed) mechanism (and let go of global efficiency), but enable separate negotiation sessions for each request, possibly making use of (fixed) negotiation templates. Although there are good result when private knowledge on one side is known, situations where two agents meet and hardly have any knowledge on each others’ costs or values are notoriously hard to deal with [2].

4 Conclusion

Easy access to mobile computing and sensing has introduced new interesting and often very useful collective data gathering and processing systems [24]. Current implementations of crowdsensing systems, however, show severe disadvantages in terms of autonomy: data is collected without a user knowing or being able to influence what exactly is shared, and is then (after agreeing to a license nobody reads) controlled by a single company. We argue that current and future work on multiagent systems can contribute to mitigating these disadvantages.

The challenges discussed in this paper in fact lay out a careful balancing act: the system has its own goals, but should also continuously adapt to meet the users request (and thus request autonomy) without lowering the users' contribution autonomy. Moreover, architectures for such systems should not be dependent on the design of the diverse and heterogeneous entities involved, whose number, characteristics, and architecture are unknown to the system designer [25].

As we have argued, to realise next generation crowdsensing systems a multitude of challenges needs to be addressed, combining many subareas from multi-agent systems research. We believe that these challenges need to be addressed not only in these respective subareas, but that it is crucial to investigate how they can be combined into a comprehensive approach. Only in this way can it be investigated how the combination of several techniques affects overall system behaviour. Moreover, we believe it is important to investigate this in a simulated setting where all system parameters can be controlled, as well as in realistic field experiments with large groups of users.

In this paper, we discussed the characteristics and requirements to the design of a crowdsourcing systems that respect the autonomy of participants. That is, autonomy must be an assumed property of participants, and the system must be able to combine and use the potential of the crowd without ignoring the individual preferences of the autonomous participants. We believe that this upcoming domain may turn out to be the 'silver bullet' that the multi-agent systems research community is searching for.

Acknowledgements

This work has been supported by the project SHINE, the flagship project of DIRECT (Delft Institute for Research on ICT) at Delft University of Technology.

References

1. E. Adar and B. A. Huberman. Free riding on Gnutella. *First Monday*, 5(10), 2000.
2. B. An, N. Gatti, and V. Lesser. Bilateral bargaining with one-sided uncertain reserve prices. *Autonomous Agents and Multi-Agent Systems*, 5 2012.
3. G. Andrighetto, G. Governatori, P. Noriega, and L. W. N. van der Torre, editors. *Normative Multi-Agent Systems*, volume 4 of *Dagstuhl Follow-Ups*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2013.
4. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.

5. I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Designing Privacy Enhancing Technologies*, pages 46–66. Springer, 2001.
6. R. K. Dash, N. R. Jennings, and D. C. Parkes. Computational-mechanism design: A call to arms. *IEEE intelligent systems*, pages 40–47, 2003.
7. M. M. de Weerdt. *Plan Merging in Multi-Agent Systems*. PhD thesis, Delft Technical University, Delft, The Netherlands, 2003.
8. M. E. DesJardins, E. H. Durfee, C. L. Ortiz, and M. J. Wolverton. A survey of research in distributed, continual planning. *AI Magazine*, 20(4):13–22, 2000.
9. V. Dignum, editor. *Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models*. Information Science Reference, 2009.
10. R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. Technical report, DTIC Document, 2004.
11. P. Dutta, P. M. Aoki, N. Kumar, A. Mainwaring, C. Myers, W. Willett, and A. Woodruff. Common sense: participatory urban sensing using a network of handheld air quality monitors. In *Proc. of the 7th ACM Conference on Embedded Networked Sensor Systems*, pages 349–350, New York, NY, USA, 2009. ACM.
12. J. Feigenbaum and S. Shenker. Distributed algorithmic mechanism design: Recent results and future directions. In *Proc. of the International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, 2002.
13. R. Ganti, F. Ye, and H. Lei. Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine*, 49(11):32–39, 2011.
14. M. Ghallab, D. Nau, and P. Traverso. *Automated Planning, theory and practice*. Morgan Kaufmann Publishers, 2004.
15. A. Kittur, J. V. Nickerson, M. Bernstein, E. Gerber, A. Shaw, J. Zimmerman, M. Lease, and J. Horton. The future of crowd work. In *Proc. of the 2013 conference on Computer supported cooperative work*, pages 1301–1318, New York, NY, USA, 2013. ACM.
16. N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell. A survey of mobile phone sensing. *Communications Magazine, IEEE*, 48(9):140–150, 2010.
17. V. Lesser, K. Decker, N. Carver, A. Garvey, D. Neimen, M. Prassad, and T. Wagner. Evolution of the GPGP domain independent coordination framework. Technical Report UMASS CS TR 1998-005, University of Massachusetts, 1998.
18. E. Ostrom. *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press, Cambridge, 1990.
19. E. Ostrom. Beyond markets and states: Polycentric governance of complex economic systems. *The American Economic Review*, 100(3):641–672, 2010.
20. D. Schrank, B. Eisele, and T. Lomax. TTI's 2012 urban mobility report. Technical report, Texas A&M Transportation Institute, 2012.
21. S. Sen. A comprehensive approach to trust management. In *Proc. of the 12th International Conference on Autonomous agents and multi-agent systems*, pages 797–800, 2013.
22. Y. Shoham and M. Tennenholtz. On social laws for artificial agent societies: Off-line design. *Artificial Intellligence*, 73(1–2):231–252, 1995.
23. J. M. Valk, M. M. de Weerdt, and C. Witteveen. Algorithms for coordination in multi-agent planning. In Vlahavas and Vrakas, editors, *Intelligent Techniques for Planning*, pages 194–224. Idea Group Inc., 2005.
24. M. Vinyals, J. A. Rodriguez-Aguilar, and J. Cerquides. A survey on sensor networks from a multiagent perspective. *The Computer Journal*, 54(3):455–470, 2011.
25. H. Weigand and V. Dignum. I am autonomous, you are autonomous. In M. Nickles, M. Rovatsos, and G. Weiss, editors, *Agents and Computational Autonomy*, volume 2969 of *LNCS*, pages 227–236. Springer, 2004.