

Engineering Crowdwork for Disaster Events: The Human-Centered Development of a Lost-and-Found Tasking Environment

Mario Barrenechea, Kenneth M. Anderson, Leysia Palen, Joanne White
University of Colorado Boulder
{mario.barrenechea,ken.anderson,leysia.palen,joanne.white}@colorado.edu

Abstract

Human computation and crowdsourcing have become hot topics across several application domains. Indeed, some efforts have been directed toward emergency management to find ways to involve the public in disaster response. However, many tasks in disaster response can put the public in harm's way or introduce problems of liability. Furthermore, some human computation tasks are disconnected from the true needs of emergency response. In this paper we discuss the case of an important lost-and-found task—reuniting displaced pets with their owners after a disaster. We argue why this task is a strong candidate for human computation by “digital volunteers.” The goal of this paper is to articulate the design decisions and software engineering problems faced in designing and developing a web-based crowdwork environment that supports a lost-and-found matching task.

1. Introduction

Disaster events are a new place for technology-based solutions. However, as is often the case, many attempts at technology development do not appropriately serve the needs of users, nor the larger institutional environment for which they are intended [11, 20]. For the safety-critical matters of disaster, this is indeed a problem. “Human computation” [3, 12, 29] is one such technology solution that holds promise as a means to involve “the crowd” which is large and attendant in catastrophes, but such deployment must be carefully done, as it too can introduce new problems in disaster.¹ Specifically, some crowdwork ideas ask people to report on what they see, but if they are directed to report about a specific location, this may mean that people converge *toward* that area. For other problems, it may be that the help digital volunteers engage in—e.g., advising people about situational awareness information—can make the *volunteers* liable

¹ For examples of human computation in disaster, see those offered by Meier in [17].

as their involvement is not covered by “Good Samaritan” laws [26]. In addition, some proposed tasks are so micro as to be uninteresting and disconnected from the larger problem that users bring far less creativity and good human judgment to the task [12].

This is not to say that all human computation problems are problematic. The work supported by Ushahidi [19], for example, has many more successes than problems. Furthermore, people *are* willing to invest time into some micro-tasks, as demonstrated by the people who participated in the effort to find the lost Malaysian jet in 2014. To help crowdwork solutions gain and maintain real ground, however, our concern is that their development must pay sufficient consideration to the task, environment, users, and larger institutional context that they are trying to support. To make such visions successful, understanding user-centered and institution-centered needs around emergency management is critical. This is the goal of crisis informatics research [7, 20-23].

In this paper, we look at a problem that occurs in disaster: pet displacement. Pet displacement refers to the separation of pets from their owners during the course of a disaster and the process that ensues to return found pets to their original owners. We describe the background empirical research we draw upon to articulate this problem and the methods we employed to design and implement a software system that aims to help volunteers with the tasks associated with reuniting displaced pets with their families. We introduce EmergencyPetMatcher (EPM) and outline its features and its design and development process. Finally, we discuss the implications for developing software to address problems such as pet displacement in disaster, with an eye towards including both the public and disaster management personnel in the system.

2. A Lost-and-Found Problem in Disasters: Pet Displacement

Domestic animals represent a large population in American households. A recent national survey indicates that Americans own an estimated 180 million

pets, roughly a quarter of which have been adopted from animal shelters [24]. Shelters face challenges of scale as 6-8 million pets enter them each year. Of these, 3-4 million are adopted while the rest are euthanized. During disasters, the number of pets entering shelters increases drastically over a short period of time, exacerbating the challenges shelters already face. Though common shelter policies put pets up for adoption after 72 hours, some so-called “kill shelters” are considered by pet advocates to be more likely to euthanize as soon as that time is up. This increases the urgency some pet advocates feel when trying to match lost and found pets.

Displacement of pets from families in disaster occurs for a number of reasons: evacuation orders may be issued while owners are away from home, fences may be blown down, or pets get skittish and escape. In 2005, the effects of Hurricane Katrina displaced more than 70,000 pets from their owners. Of these, only 3% were ever reunited with their original owners [32]. Furthermore, these numbers do not take into account the psychological impact on pets and their families, particularly for the young and elderly [10, 15, 18]. After the 2006 Pets Evacuation and Transportation Standards Act (PETS) was passed, disaster management policies have been revised to include accommodations for the transportation and sheltering of domestic animals as well as their owners [18]. Though this ensures that resources are available to families with pets during a disaster, it does nothing to address how pets could be reunited with their families. Pet advocates then turned to social media for solutions.

2.1. Social Behavior in Disaster Events

New attention is being paid to the role that volunteers—including digital volunteers—are playing in disaster [7, 9, 30]. White et al. report on the grassroots use of Facebook in the aftermath of Hurricane Sandy [32] and how the Sandy Pets Facebook page became a way for pet advocates to evolve as an organized group with an improvised case management system to support pet-to-family reunion. That these volunteers exist is not surprising; the public has been seen as a large-scale and collectively intelligent force that makes use of technology during times of emergency to perform sense-making tasks [20]. Digital volunteers have been observed acting in this way to provide relief [21, 30] and situational awareness information [21, 23, 30, 32].

2.2. Pet Matching as a Crowdtwork Solution

Early in the aftermath of the Hurricane Sandy event, online pet advocates and owners started cross-

posting photos of displaced pets—this is something they normally do across a range of social media, but without direct coordination with others [32]. “Cross-posting” is considered a good practice among online pet advocates and is how they normally get the word out about pets that need to find a new home when owners relinquish them to shelters. However, the influx of displaced (rather than unwanted) pets taken into shelters as a result of the hurricane provided pet advocates with a new challenge: Not only did they disseminate photos of pets, but they realized they could engage in the new task of *matching* the photos to reunite pets and their owners.

By transforming Facebook features such as photo albums and comment threads into customized information structures, the digital volunteers created an improvised case management system to post and sort “lost dogs,” “found cats,” and so on, in different geographical regions. Thread-based conversations associated with photos of pets ensued between members. These activities were collaborative: The activity of manually comparing a lost pet to a found pet and sharing results was hailed as valuable work [32].

This intrinsically motivated behavior that became focused around the needs of animals in a disaster is the inspiration for developing a crowdwork platform for disasters that can mobilize volunteers in a safe way around a specific problem. Though the technology of Facebook was able to support this collective behavior, the limitations imposed on communication between people connected only through Facebook Pages and Groups meant personal messages such as those attempting matches and pet owners were not widely visible, limiting opportunities for collaboration.

We know from other disaster events that many are looking for software that provides features that support pet matching tasks directly. Our platform demonstrates how the tendency to engage in lost-and-found matching can be realized computationally with distributed human support in a way that still exploits human creativity without reducing human involvement into unsatisfying micro-tasks. The focus here on pets should be truly helpful on-the-ground in disasters with major human and animal displacements, but we note the strategies can be used for other lost-and-found disaster tasks as well. For example, documents and photographs displaced from people’s homes during tornadoes, floods, and hurricanes could prove to be another use of this type of technological intervention.

3. The EmergencyPetMatcher System

Fig. 1 shows the home page of the EPM system, which presents a photo-centric interface that invites users to perform work right away by showing pictures

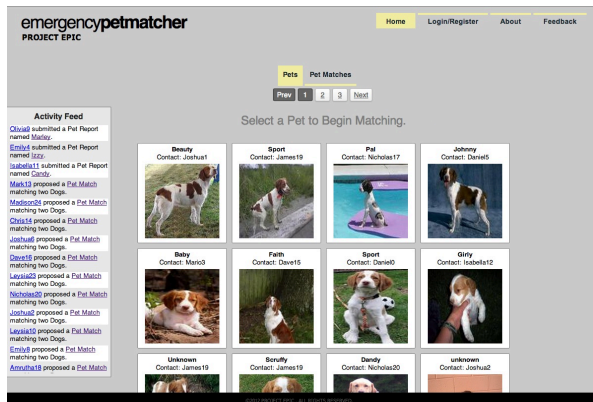


Fig. 1. The EPM home page.

of reported lost and found pets. On the home page, EPM displays a “newly added” section where the user can see all reported pets. By clicking on a thumbnail of a pet, a user can see the report for that pet. To encourage visitors to sign up and participate, anyone can examine the pet reports, but only registered users can help match pets, bookmark reports for later work, or communicate to those who reported pets as lost or found. There are links on the home page for submitting a lost/found pet, the user’s profile page (if logged in), the “about” page, and a feedback form. If an authenticated user is viewing the home page, the activity feed on the left-hand side of the page outlines the different activities performed by other EPM users who are being followed. Otherwise, the activity feed presents a random slice of activities from EPM users.

3.1. Submitting a Pet Report

An EPM user can submit a pet report for a lost or found pet and provide information that can lead others to assist in matching it with other pets in the system. A pet report is described by mandatory attributes such as pet type (i.e. dog or cat), status (lost/found), and date lost/found, but they are supplemented by non-mandatory fields such as name, breed, age, sex, spayed/neutered, coat color(s), size, description, tag collar text, an image that illustrates the pet, and a microchip ID if available. Cross-posting to other social media is supported by encouraging users to submit pet reports and contact information on behalf of original pet owners who may not know about EPM. These “contact fields” are optional but are shown on the actual pet report once submission is complete. Once a pet report is submitted, it is saved in the system database and then displayed in the “newly added” section of the home page. The pet report itself displays all of the entered attributes, as well as any proposed pet matches for that pet, and provides links to bookmark the pet to later work on matching the pet.

3.2. Matching a Pet Report

A registered user can navigate to the matching page (see Fig. 2). On this page, the user can see a list of candidate pet reports from which he/she can select to propose a pet match. Logically filtered and ranked based on supplied pet report attributes, the candidate list shows the most likely pets at the top while providing pagination to scroll through them. Users are able to click and drag pets from the candidates list and compare the attributes for both pets, almost as if playing a “matching game.” The matching activity is supported by the system’s ability to sift the most likely candidates to the top, which makes the activity simple and straightforward. Once the user proposes a pet match by pairing the two pets together, the system will create the pet match and display it for the crowd to evaluate and vote on. All pet matches are automatically linked to the pet reports that they bind together.

3.3. Voting on a Pet Match

On the pet match itself, the user finds buttons for voting “up” and “down.” EPM enables the crowd to up-vote or down-vote pet matches as a means to verify their “correctness.” By this, the system tries to capture how confident the crowd is with respect to a proposed match. EPM has a threshold of up votes versus down votes that is based upon the number of active users (as opposed to registered users) in the system to trigger a “successful match.” Once the threshold has been triggered for a particular match, the system pauses the voting activity and sends email to the contacts, asking for resolution outside of the system to determine if the pets being matched by EPM members are indeed the same pet. Presumably, the coordination of people to verify the match is done without need for system facilitation, since it is clear that EPM cannot guarantee that a successful match will be made. Therefore, EPM leaves it up to the contacts to deliberate and return to the system for final verification. Once both users respond with their final vote of “yes,” the pet match is showcased on EPM as a successful reunion in a separate gallery on the home page, and all pet match suggestions related to the successful match are then closed. If there is inconsistency with the vote, EPM will err on the side of caution and close the match but will allow attempts to match either pet to continue.

3.4. Connecting with Other Users on EPM

A user has access to his/her own profile page and can view the profile pages of other registered users on the system. EPM implements the following model of social connectivity, allowing users to “follow” one

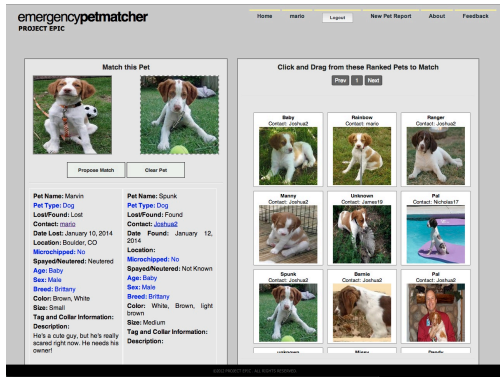


Fig. 2. Matching two pets together.

another and subscribe to activities that they themselves perform. For example, if EPM user Kathy123 starts to follow Tim456, then Kathy123 will be able to view Tim456's activities on the home page activity feed. We believe this will allow users to collectively work together on specific pets and extend their networks both inside and outside of EPM to spread volunteer awareness. EPM users can also personally message each other without revealing email addresses to maintain privacy among the digital volunteer crowd.

These social features promote cooperative efforts to work together in finding and reporting pets. These efforts may be physical if users are on-site at a disaster location and are coordinating where to look for pets. They allow remote users to match or vote on pets and connect with others to build emergent online communities. We emphasize the collaborative aspects of the crowdwork being done on EPM, since the online behaviors of this digital crowd can help us to design services for other disaster-related tasks.

3.5. Promotion of EPM

EmergencyPetMatcher, pending IRB approval, will be ready to deploy at a moment's notice for a disaster event that involves the mass displacement of a population, as this is what gives rise to pet-separation issues and the anxieties around evacuation and sheltering decisions that then ensue. A social media campaign will complement the technology deployment, which can also include links to tutorials and other materials (see [31] for evidence on how this worked in our prior tech-for-disaster campaigns). Pet advocates, a subset of the intended audience for EPM, have already demonstrated intrinsic motivation to attend to the needs of missing pets. However, we know that during disaster events, more general case volunteers also seek ways to assist. EPM therefore provides a platform that meets the needs of both specialized and general interest audiences. The

platform offers ways for people to perform meaningful work in a collaborative, rapid fashion.

It is also important to note that EPM is meant to complement existing social media services, and to work in concert with those in terms of generating interest. People use multiple social media platforms, but as we have seen in our empirical work [32], seek "destinations" for collecting and collating lost and found data from across the internet. EPM as a purpose-built tool provides users with a central space to work, along with the ability to amplify that work by posting suggestions to other social networks. These sharing methods will draw more people to the platform.

4. EPM Design and Development

We now describe the methods used to design and develop the EPM system. The design process includes the use of personas, cognitive walkthroughs, and think-aloud protocol sessions. The latter were conducted with real users to shape EPM's photo-centric interface. The development process made use of an agile life cycle that focused on rapid prototyping with iterative customer feedback, scalable and dependable technologies, and developer reviews to improve communication and code clarity. The purpose here is to present the engineering rationales needed for the development of crisis informatics software.

4.1. EPM Design

The design for EPM emerged from results of our empirical research on socio-behavioral phenomena related to social media use in disaster events [2, 32]. However, the space of ICT design and development to address information needs in humanitarian crises is wide. Hughes used participatory design methods to elicit requirements for social media tools designed for emergency managers [6]. Volunteer-tech communities, which are distributed groups that help those in need through social media and open data platforms, have developed strategies for facilitating software development in humanitarian response [4, 30]. Ushahidi [19] is a well-known instance of this—it uses a crisis-mapping platform in concert with a community forum to populate it with data about crises. The community is mobilized to fill functional roles in complement with the software, such as report verification, geo-mapping, and filtering of inaccurate reports. Other design considerations emerged from products in the commercial sector, including the pet-finding service HomeAgain, which operates as a software-as-a-service and stores micro-chipped information that can be used to make accurate matches. PetFinder displays listings of all lost and found pets

with filtering capabilities to search through them, but ultimately operates much more like a “shelter board” of status reports than an interactive matching platform.

The goal of our design effort then was to envision a system that serves a large convergence of digital volunteers helping to make successful pet matches between lost and found pet reports as a collaborative and crowdwork-based activity. To facilitate a user-centered approach, our design process focused on three areas: We 1) developed detailed personas (archetypes of people we envisioned using our system and their motivations); 2) produced low-, medium-, and high-fidelity mockups of the system with iterations of cognitive walkthroughs, and 3) conducted usability tests using think aloud protocols to improve design.

4.2. Personas

We developed personas [5] to identify a representative set of users along with their backgrounds, motivations, and goals when using EPM. For example, Charlie is an EPM “matcher”: *Charlie is a 4th grade student whose teacher is interested in assisting victims of a recent earthquake in California. The teacher directs Charlie and his classmates to EPM as a volunteer activity. Charlie enjoys finding pet matches for lost dogs—the activity is like a game to him.* From the persona development, three user roles emerged: *data scout, matcher, and checker:*

Data Scout: The data scout is a volunteer on the ground helping to rescue animals in affected areas. The main behavior of a data scout is uploading pet reports to EPM via a mobile device or laptop.

Matcher: The matcher is a digital volunteer either within the affected community or outside of the area working to reunite pets with their owners.

Checker: The checker works closely with matchers to find new pet matches and votes on the likelihood of them being successful matches.

Structured models for understanding the behavior of volunteers using EPM allowed the design of system components and guided our cognitive walkthroughs.

4.3. Cognitive Walkthroughs

Fig. 3 shows a low-fidelity and a medium-fidelity mockup of the EPM interface that were used to perform cognitive walkthroughs of a user scenario and persona. All of the design choices that were made in these mockups were targets for feedback from our usability studies. The low- and medium-fidelity mockups consisted of a search and filter form at the top of the page, as well as a “Match Feed” section that

showed suggested, failed, or successful pet matches, along with the names of the users who matched them. At the center of the page are two pet-matching panels, one panel for showing lost pet statuses, and the other showing found pet statuses. Chat functionality was also available to allow for quick impromptu connections to be made among users to discuss potential matches. If a user suggests a match, a request for verification of the match can be distributed across all active chat rooms. This design allows for rapid and focused matching.

Using both the low- and medium-fidelity mockups, our team of three researchers conducted cognitive walkthroughs on the matching task. The cognitive walkthrough [13] is a self-directed progression over a sequence of steps to complete a system task. Below is a cognitive walkthrough that our previously mentioned persona “Charlie” would need to suggest a potential match for “Bob the Dog” via browsing or searching. For example, with searching, the following steps are performed:

- 1) *Charlie clicks on “Bob the Dog” in the Lost Pets column,*
- 2) *Charlie opens up the search above the pet columns by clicking on the Filters button above,*
- 3) *Charlie enters the search terms related to “Bob the Dog,”*
- 4) *Charlie clicks on the Search button,*
- 5) *EPM fades out the Found Pets column and fades in with updated results based on the search,*
- 6) *Charlie clicks on the potential match; then he clicks on the Suggest Match button below, and finally,*
- 7) *Charlie clicks submit and receives confirmation that a “Suggested Match” has been made.*

Imagining ourselves as Charlie, we documented the steps that would lead us to suggest a match for “Bob the Dog.” We then wanted to determine which of these workflows would be used most frequently when we conducted the think-aloud runs.

4.4. Think-Aloud Protocol Runs

Information from the cognitive walkthroughs helped define EPM’s workflows. With this in place, we then conducted a think-aloud session [14] with eight participants ranging from 16 to 71 years of age. The think-aloud session is a usability study for users to interact with a prototype and perform a task without any guidance. The goal in our sessions was to find “Bob the Dog” using our medium-fidelity prototype.

All of our participants completed the goal successfully. For example, the quickest participant completed the task in less than ten seconds while the slowest took five minutes. Some users reported confusion about using the searching/filtering functionality over the browsing functionality, and vice versa. Other issues included the visual landscape; some



Fig. 3. Low- and medium-fidelity mockups for the EPM interface.

users were confused about which buttons to press, how to interact with the pet columns, and how to submit a pet match. Also, before submitting their matches, some users were not sure if the match selected was the best one, stating they would continue to look for more matches. This feedback directly informed the final design shown in Figs. 1 and 2. These eight participants provided enough consistency in feedback to inform the design work as we moved from low to high-fidelity prototyping and eventually development.

4.5. From Design to Development

The design process used for developing the interface and workflows extended itself to subsequent steps of system specification. During implementation, the development team organized itself so that they could continuously represent the user: the end-users were represented by three information scientists and the engineering team was led by four software engineer researchers and developers who incorporated user feedback each iteration. The information scientists included researchers who study animal welfare issues; they participated in the design phase and then shifted to the end-user role during development. To develop EPM, the development team adapted an agile life cycle with features that directly support development of crisis informatics software. We now describe our iterative life cycle, our development environment, developer-based reviews, and the way we interfaced with our non-traditional “customers.”

4.6. EPM Development

The team made use of the core agile principles—personal communication, iterative development, customer collaboration, and being responsive to change—in our development of EPM. At the outset, the team created a roadmap with the user stories crafted from the decisions made during the design. An example user story appears as follows: *As a data scout,*

I would like to upload/submit a Pet Report with all of its information so that I can provide content to the system and allow others to see my Pet Report submission. Here it is clear that the data scout gets value from submitting pet reports, since the scout’s primary role is to upload relevant information for the pets that the scout finds after a disaster. The user stories capture feedback in a way that allows the customer to evaluate its value in the system. The developers defined a point-per-hour based ratio (1:4) that allowed them to accurately estimate the workload required for a user story. Each user story was assigned a point value, with the duration of time expected to complete that user story expressed as a function of the number of points it was assigned. Through the team’s estimation skills were not finely tuned at the outset of the project due to variation in development experience, they improved as the project progressed.

Fig. 4 illustrates how agile methods were used in the implementation of EPM. The numbers indicate the steps followed during each sprint (i.e. iteration).

1. **Customer feedback is solicited:** Customers are interviewed to identify workflows that are not usable and to discuss bugs in the prototype.
2. **Backlog input:** This feedback serves as input into the development process. Suggestions, bugs, and ideas are translated into user stories and stored in a list of development tasks for prioritization.
3. **User story selection:** User stories are selected and scoped for the current sprint. Development begins, and the team meets for a checkpoint meeting mid-week. At the end of the week, changes are finalized and the team meets for an iteration-planning meeting (IPM). The IPM allows the developers to evaluate their performance and identify user stories for the next sprint.
4. **Backlog update:** All results from the sprint, including bug fixes, finished/unfinished user stories, or new bugs are updated in the backlog (list of tasks). Incomplete user stories are split so

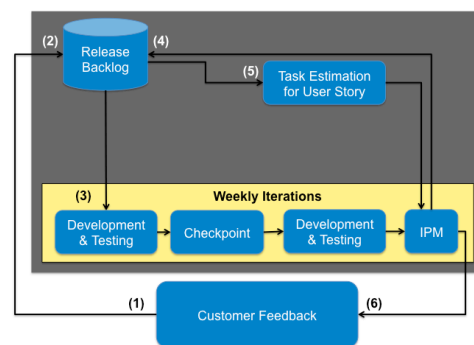


Fig. 4. The EPM development process.

that completed tasks are documented, and incomplete tasks roll over to the next sprint.

5. **Task estimation:** Based upon the performance of the current sprint, the developers update estimates for incomplete user stories, and they estimate the time required for new user stories.
6. **Iterative deployment:** Once the IPM is finished, completed user stories are incorporated into the staging environment for customer evaluation. Customers assess the changes and identify any new bugs. The prototype is critically evaluated from the perspective that it represents the version of the software ready for deployment in a disaster.

The EPM development team completed twenty-two sprints in over five and a half months. The V1 release backlog was not completely filled by the first sprint. In fact, most of the user stories were generated in subsequent sprints because it was impossible to map out every user story at the project's outset. To be adaptive to change, the development team allowed for the release backlog to fill with new stories as long as a working system was in place by the end of each sprint.

The twenty-two sprints allowed the development team to scope the features that were critical to the success of the initial deployment of V1. Although development experience in the team was limited, the developers worked closely day to day through the use of communication tools. Online documentation described the stages of the life cycle, tracked design and development notes from meetings, established coding conventions, tackled deployment issues, and tracked IRB details and usability testing reports.

4.7. Developer Reviews

Towards the end of V1, the developers engaged in a process known in commercial development as a "360 review." In this review, all members of the team convened to write a performance report of every other developer. Since there were four developers on the team, four meetings were held with three developers to review the remaining developer with anonymity and consensus. The standards for this review included: Task Estimation, Code Clarity, Code Performance, Communication, Quality Assurance, and Cooperation. Each standard was rated on a scale of 1 to 5.

Results of this review process were positive and constructive. A sample extracted from one of the review documents (used with permission and anonymized) for "Code Performance" reads: *Mark is cognizant of future changes that have to be made to the code and does not compromise on the performance of the code while completing his user story.* More constructive feedback for "Task Estimation" reads:

Mark does a good job of estimating tasks, however there is room for improvement. The user story "activity feed and storing logs" was carried over from sprint 13 to sprint 16. Maybe, this was a user story that was too much for just one sprint. A good practice for all of us could be to give room for solving unexpected conflicts/bugs/issues that might crop up during development and testing. Communication and code performance issues were among the most prevalent in the review, likely due to the variation in development experience. The reviews helped to highlight these issues in hopes that each developer would use the feedback to perform better in future software projects.

4.8. The Development Environment

The development team selected software technologies for the development environment that provided fast turnaround time for prototyping, trial and error, and feature delivery. EPM is built using Python and the Django web application framework. PostgreSQL was chosen as the relational database for scalable persistence. Nginx is an HTTP web server that serves the static content for the application, such as images of pets, user profile pictures, CSS, Javascript, and HTML files. If a request requires dynamic content, such as loading pet reports, Nginx delegates the request to the Unicorn application server. A large number of popular open source and community-led python modules were used for the project such as django-registration, python-social-auth, and Pillow and enabled large gains in code modularity, reusability, and clarity. The team also made extensive use of the Git version control system to allow developers to work in parallel while keeping up-to-date on the latest changes. To ease complexity on deployment and configuration, the virtual machine bootstrapping tool Vagrant and provisioning tool Puppet were used to enable rapid and repeatable bootstrapping of the development and deployment environments. These choices continue to provide benefits as EPM is deployed for crisis events. They make it possible to spin up additional databases and web servers to deal with spikes in user activity.

4.9. Customers and Users

The end-users of the EPM system are not customers in the traditional sense. The system is designed to support a population impacted by disaster and the community of volunteers that form around the event. EPM is a solution targeted for the spontaneously formed community of pet owners impacted by the event and the community coming to their aid. Due to the unique circumstances of how this user community forms, we relied on the work of disaster researchers for

insight into how to build a system that can be adopted easily and adapted for use in the disaster domain. We also collected data on how pet advocates used Facebook to perform the process of pet-to-family reunion and fed their behaviors into machine learning techniques to classify their activities. We then used those classifiers to simulate behaviors of potential EPM users to test EPM under realistic scenarios [25].

5. Discussion

We now discuss issues important to crisis informatics and software engineering and we examine the divide that exists between the “formal” and “informal” response and how to narrow this gap.

5.1. Software Engineering for Disaster ICT

Systems built for disaster response face challenges of scale as well as intense pressure to adapt to the changing needs of their users. It is not enough to compartmentalize requirements of all users and stakeholders into use cases, because the social environment within which the software is deployed in a disaster evolves over time, necessitating the system to evolve as well. ICT systems that presume a representation of plans that describe how a system or procedure should be used—as is the case in command-and-control models for emergency management—without gaining insight in the particularities of their use, are doomed to be brittle. From the work developing EPM, we have discovered three guiding principles that allowed us to prepare a system to be user-centered, receptive to change, and robust in the face of increased user activity. We discuss each in turn.

5.1.1. Usability. A user-centered design process promotes usability and goal-oriented workflows for users. Based on our techniques, it was clear that users who were interested in matching pets found the two-column orientation of lost and found pets to be useful and straightforward. From this, we gathered evidence that a strongly visual design must be used throughout the application to encourage matchers and checkers to stay on task. Usability must begin with analyzing the needs of the system’s users and their behaviors, the fundamental premise of a socio-technical system. Without this attention, the system risks frustrating its users and losing active users as people come to the system and are forced away by bad design.

5.1.2. Adaptability. *In situ* adaptation allows developers to respond to changing requirements generated by the evolving behaviors of a system’s

users. To achieve this, a large portion of the EPM development process is on incorporating customer feedback into user stories for iterative development. This trained the developers to seek such feedback as needed to produce working software at the end of each sprint. This development process ensures that the developers are ready to respond to new requirements as the system is being used. The challenge here is that a typical sprint of one week may need to be reduced to just a few days or hours to be truly adaptive.

This level of commitment is required of developers who build ICT for disaster response, as it is difficult to completely anticipate the features needed for the changing environment of a mass emergency. We have put in place mechanisms to enable fast adaptation of EPM functionality to user needs. This includes implementing system logging—e.g. database and web server performance plus logs of user activities—to give the developers a big picture view of EPM’s behavior and performance. We believe coadaptation between a system’s functionality and user needs is critical for ICT used for disaster response.

5.1.3. Reliability. When designing a system meant to be used in the context of a disaster event, reliability and robustness are important. Such software needs to be consistently available, responsive, and designed such that it can be updated without interrupting use. The system must also be robust enough to handle the dynamic load generated by volunteers that can result in bursts of high activity. It should allow communities of users to converge on it so that important work can be carried out throughout a disaster. Techniques to support this include implementing redundant storage mechanisms for backing up crowd data, caching solutions for decreasing latency of data retrieval, and incremental and continuous stress-testing [1, 28].

These engineering principles are by no means exhaustive, but they help support designing for the intended audience, adapting the software based on actual (not planned) use, and building with reliable technologies. They provide a strong foundation to build ICT for formal and informal disaster response.

5.2. Closing the Information Divide

The roles of both emergency managers and members of the public in responding to disaster events are rapidly changing. Emergency events before the ubiquity of ICT saw the need for formal response to provide situational awareness for the event, including damage assessment reports, casualties, missing persons, and status updates on the disaster. To properly relay such official updates, a public-information officer provides information to mainstream media outlets that

then convey that information to the public. As was evident in research that examined varying perspectives between public volunteers and emergency management personnel, victims of the 2010 Madeira floods reported having no sense of situational awareness and feeling “left out” by emergency response [27]. However, the true “first responders” were the disaster victims themselves [6, 20], collecting knowledge about their surroundings, maintaining ties with family, and assisting others. With the wide-scale use of ICT, we have observed that the ability to volunteer in these situations has been amplified via technologies that support peer-to-peer communication [7, 21, 30, 32]. These “true first responders” do many of the tasks that trained personnel perform: they route/verify information, find and report on pets, and assist in transporting, feeding, and providing support for victims [23]. With the “power of the crowd” close at hand via ICT, formal emergency management decisions can be made with the public in mind.

There are challenges that must be addressed when introducing ICT for use by both the public and the formal response. The trustworthiness of data published and distributed by digital volunteers is variable; misinformation and disinformation do occur, and there are no observable safeguards in place in social media services to filter out such reports, except by the actions of their users. The mistrust of social media data by the formal response is understandable given their need to relay correct information. However, most of the time, data published by the public are consistent, and activities to verify information have been observed in previous events [7, 21]. It is also important to see that a “gold standard” of accurate and verified information can never be achieved due to the varying context around which people create and distribute information [23]. To leverage public data on a large scale to help inform disaster management decisions, questions of accuracy should migrate from “How accurate is this?” to questions such as “What kind of information would be useful now?” This shift in perspective can help bridge the gap between the informal and formal response and allow emergency managers to take advantage of the work of digital volunteers.

With respect to EPM, these issues were considered from the very start. The principles followed for this work were instantiated not purely for fast development but also to respond to the changing interactions of the informal and formal response with regards to pet displacement. The user role of “data scout” exists in the world today and is occupied by members of the public who volunteer at shelters alongside emergency management personnel. Though the formal response may not have time to submit pet reports to EPM, data scouts can deploy to all of the shelters within a region

and take on that task, freeing the official workers to focus on receiving animals, transferring animals and other resources between shelters, etc. Given that the formal response will have some insight into the fact that animals are being reported in this way, they can then come to appreciate the value of digital volunteers who might identify matches of these pets with lost pet reports, allowing the animal to be removed from the shelter and reunited with its original owners.

6. Conclusion

Disaster events create opportunities for volunteers to help victims in need. With pervasive ICT in the hands of the public, the roles that formal and informal response play in the distribution of information are changing, bringing new challenges for system design to foster digital volunteer communities that collectively solve real-world problems. Pet displacement is one such problem. In this work, we have presented EmergencyPetMatcher, a system that allows the digital crowd to report, match, and verify lost and found pets collaboratively. We presented EPM features, explained its design and development processes, and discussed engineering rationales that address the information divide between emergency responders and members of the public. We also identified principles most relevant to the design and implementation of systems that are meant to support disaster-based crowd work in ways that are safe, reliable, realistic, and helpful for all.

Acknowledgments

This material is based upon work sponsored by the NSF under Grant IIS-0910586.

References

- [1] Anderson, K. M., Schram, A.: Design and implementation of a data analytics infrastructure in support of crisis informatics research (NIER track). In 33rd International Conference on Software Engineering. Honolulu, HI, 844–847 (2011).
- [2] Barrenechea, M. Barron, J. White, J.: No place like home: pet-to-family reunification after disaster. CHI 2012 Extended Abstracts On Human Factors In Computing Systems. Austin, TX, 1237–1242 (2012).
- [3] Bernstein, M., Little, G., & Miller, R. (2010). Soylent: a word processor with a crowd inside. Proc. of the 23rd Annual ACM Symposium on User Interface Software and Technology (UIST). (2010).
- [4] Blanchard, H., Chapman, K. Volunteer Technology Communities: Open Development. In Global Facility for Disaster Reduction and Recovery. (2012).

- [5] Cooper, A.: *The Inmates are running the asylum*. Indianapolis, IN: Macmillan Publishing, 288 pages (1999).
- [6] Dynes, R. R.: *Organized behavior in disaster*. University of Michigan: Heath, 235 pages (1970).
- [7] Hagar, C., Haythornthwaite, C. Crisis, farming and community. *The Journal Of Comm. Inform. North America*, vol. 1, 41–52 (2005).
- [8] Hughes, A. Participatory design for the social media needs of emergency public information officers. *Proc. of Information Systems for Crisis Response and Management (ISCRAM)*. 727-736 (2014).
- [9] Hughes, A., Palen, L., Sutton, J., Liu, S., Vieweg, S. “Site-seeing” in disaster: an examination of online social convergence. *Proc. Of the Information Systems for Crisis Response and Management (ISCRAM)*. 2008.
- [10] Hunt, M. G., Bogue, K., Rohrbaugh, N.: Pet ownership and evacuation prior to Hurricane Irene. *Animals*. Vol. 2, 529–539 (2012).
- [11] Kendra, J. M., Wachtendorf, T. Reconsidering convergence and converger legitimacy in response to the world trade center disaster. In *Terrorism and Disaster: New Threats, New Ideas (Research in Social Problems and Public Policy)*. 11, 97-122 (2003).
- [12] Kittur, A., Nickerson, J. V., Bernstein, M., Gerber, E., Shaw, A., Zimmerman, J., Horton, J. The future of crowd work. In *CSCW*. 1301, New York, New York, USA (2013).
- [13] Lewis, C., Polson, P., Wharton, C., Rieman, J.: Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. In *Proc. SIGCHI Conference on Human Factors In Computing Systems*. 235–242 (1990).
- [14] Lewis, C., Rieman, J.: *Task-centered user interface design: a practical introduction*. Boulder, CO: University of Colorado, Boulder, Department of Computer Science, 138 pages (1993).
- [15] Lowe, S., Rhodes, R., Zwiebach, J. E. Chan, C. S.: The impact of pet loss on the perceived social support and psychological distress of hurricane survivors. In *Journal of Trauma Stress*. Vol. 22, 244–247 (2009).
- [16] Martin, D., Hanrahan, B. V., O’Neill, J., & Gupta, N. Being a turker. *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*. 224–235 (2014).
- [17] Meier, P. Next Generation Humanitarian Computing. Keynote talk at ISCRAM 2014.
- [18] Mike, M., Mike, R., Lee, C.: Katrinas animal legacy: the PETS act. *Journ. Animal L. and Ethics*. 13, 133–160 (2011).
- [19] Okolloh, O. Ushahidi, or “testimony”: Web 2.0 tools for crowdsourcing crisis information. In *Participatory Learning and Action*. 65-70 (2009).
- [20] Palen, L., Liu, S. B.: Citizen communications in crisis: anticipating a future of ICT-supported public participation. In *Proc. Of SIGCHI Conf. on Hum. Fact. In Comp. Syst.* San Jose, CA, 727–736 (2007).
- [21] Palen, L., Vieweg, S., Liu, S. B.: Crisis in a networked world: features of computer- mediated communication in the April 16, 2007, Virginia Tech event. *Soc. Sci. Comp. Rev.* Vol. 27, 467–480 (2009).
- [22] Palen, L., Anderson, K.M., Mark, G., Martin, J., Sicker, D., Palmer, M., Grunwald, D.: A vision for technology-mediated support for public participation and assistance in mass emergencies and disasters. In *Proc. Of ACM-BCS Vis. Of Comp. Sci. Conf.*, art. 8, 12 pages (2010).
- [23] Palen, L., Vieweg, S., Anderson, K. M.: Supporting “Everyday Analysts” in Safety- and Time-Critical Situations. *The Info. Soc.* 27(1): 52–62 (2010).
- [24] Pets by the Numbers. Retrieved from http://www.humanesociety.org/issues/pet_overpopulation/facts/pet_ownership_statistics.html (2013).
- [25] Rajiv, A.: Analyzing user behavior on Facebook’s “hurricane sandy lost and found pets” page to improve support for pet matching in crisis informatics applications. MS Thesis, CU Boulder, 85 pages (2013).
- [26] Robson, E. S. Responding to liability: evaluating and reducing tort liability for digital volunteers. Woodrow Wilson Center. 2013.
- [27] Rogstadius, J., Teixeira, C., Karapanos, E., Kostakos, V.: An introduction for system developers to volunteer roles in crisis response and recovery. In *10th Int. Conf. on Info. Syst. For Crisis Resp. and Mgmt.* 1–10 (2012).
- [28] Schram, A., Anderson, K. M.: MySQL to NoSQL: data modeling challenges in supporting scalability. In *Proc. Of the 3rd Ann. Conf. on Syst. Prog. And App.: Soft. For Hum.* 191–202 (2012).
- [29] Starbird, K.: What “crowdsourcing” obscures: exposing the dynamics of connected crowd work during disaster. *Coll. Int.* (2012).
- [30] Starbird, K., Palen, L.: “Voluntweeters”: self-organizing by digital volunteers in times of crisis. In *Proc. Of SIGCHI Conf. on Hum. Fact. In Comp. Syst.* Vancouver, BC, Canada, 1071–1080 (2011).
- [31] Starbird, K., L. Palen, S. Liu, S. Vieweg, A. Hughes, A. Schram, K. Anderson, M. Bagdouri, J. White, C. McTaggart, C. Schenk. Promoting structured data in citizen communications during disaster response: an account of strategies for diffusion of the “tweak the tweet” syntax. In Christine Hagar (Ed) *Crisis Information Management: Communication and Technologies*, Chandos Pub., (2012).
- [32] White, J. I., Palen, L., Anderson, K. M.: Digital mobilization in disaster response: the work and self-organization of online pet advocates in response to hurricane sandy. In *Proc. of the 17th ACM Conference on Computer Supported Cooperative Work and Social Computing*. (2014).