Development of a virtual dashboard for event coordination between multiple groups

Trishan R. de Lanerolle
Trinity College, Hartford CT
trishan.delanerolle@trincoll.edu

William V. Anderson
Case Commons, New York, NY
williamv.anderson@gmail.com

Sam DeFabbia-Kane
Wesleyan University, Middletown CT
defabbiakan@wesleyan.edu

Eli Fox-Epstein
Wesleyan University, Middletown, CT
efoxepstein@wesleyan.edu

Dimitar Gochev
Trinity College, Hartford CT
dimitar.gochev@trincoll.edu

Ralph A. Morelli
Trinity College, Hartford CT
ralph.morelli@trincoll.edu

ABSTRACT

Collabbit is an open source web-based application that aims to increase emergency management efficiency through distributed asynchronous information sharing. The software is targeted to loosely coupled non-profit disaster relief agencies that coordinate response to and recovery from disasters. Disaster relief agencies create a common operating picture of an emergency incident through remotely posted *incident updates.* Individual users subscribe to topics of interest and receive near-instantaneous updates on those topics. Where information is lacking, users may access a topically organized contact registry. This report describes the development and deployment of the Collabbit project.

Keywords

Humanitarian Open Source, communication tool, coordination tool, Collabbit, disaster management, collaborative technologies

INTRODUCTION

Collabbit is an easy-to-use web-based collaboration system that helps cooperating agencies share and receive information during the response to emergency incidents. The system focuses on common communication protocols, including SMS and email communications, that most, if not all, agencies find accessible. It is thus well suited for loosely coupled multi-agency disaster relief efforts where centralized control is largely absent. Collabbit distributes authority to individual actors and relies upon them to organize themselves into a productive problem solving body.

Collabbit follows a nimble user-driven development process with a focus on an intuitive user experience. As we describe below, its development process was highly iterative, incorporating the constant feedback from users.

There has been a significant trend towards the use of free and open source software (FOSS) as a low cost and flexible means to support incident coordination. Recent examples of this trend include Ushahidi, developed in 2008, to mash up user-generated reports and Google Maps to gather citizen generated crisis information related to the post-election violence after Kenya's 2008 election. Ushahidi has gone on to be used to track violence in Gaza and assist in gathering reports of the recent global H1N1 viral outbreak (Aljazeera 2008).

The development of the Sahana system in response to the coordination needs following the 2004 Asian Tsunami (P. Currion 2007) was the most influential example for the Collabbit project. Sahana has been deployed to many disasters since the Tsunami, including the January 14, 2010 Haiti earthquake. Disaster relief professionals, academics, and technologists actively contribute to its continued development (Morelli, et al. 2010). Collabbit uses a similar centralized model as Sahana. However, unlike Sahana, Collabbit does not track individual victims, refugee camps, organizations, or volunteers. Instead, Collabbit coordinates the collaborative process of

Reviewing Statement: This paper represents work in progress, an issue for discussion, a case study, best practice or other matters of interest and has been reviewed for clarity, relevance and significance.

determining which agency in the coalition will act upon the shared situational awareness. In the following sections we discuss the development and deployment of Collabbit, focusing on two recent deployments in New York City.

DESIGN CONSIDERATIONS

Collabbit helps emergency managers answer two critical questions during emergency response: (1) What are partner agencies doing? (2) How can I quickly talk to the relevant person? With those questions in mind, Collabbit has two functional areas: *incidents* and *contacts.*

Users create records (*incidents*) and post situation updates to share progress, needs, and situation reports with the rest of the disaster relief network. These updates are free-form blocks of text which support file attachments and tags. They are targeted to user-created *groups*. Individuals subscribe to groups of interest and receive automated email and SMS text message updates.

Contacts aggregate all user profiles into an easily searchable and filterable display. Users find the person they would like to talk with by filtering group membership or through an integrated search feature. Once a person or group of people is found, users can download the contact into a standard format or send an email message.

Collabbit was designed and built through a collaborative effort between students from the Humanitarian Free and Open Source Software (HFOSS) Project (Morelli, Tucker, et al. 2009) and practitioners from the New York City Voluntary Organizations Active in Disaster (NYC VOAD). The goal was to create a user friendly and financially appropriate alternative to proprietary disaster relief coordination software tools.

DESIGN PHILOSOPHY AND RATIONALE

The Collabbit team aimed to develop a customizable disaster coordination tool with a focus on usability. The developers followed Keep-It-Simple-Stupid (KISS) design principles emphasizing ease of use over complex set of capabilities.

Collabbit's open source, volunteer driven, low cost development infrastructure yields software that follows the JustPartners, Inc study guidelines for successful post-disaster web portals (JustPartners, Inc. 2008). Collabbit's inherent flexibility increases the chance for community buy-in, builds trust through complete transparency, and engenders local empowerment through the ability to change the source code to meet local conditions without penalty. There are other disaster management products on the market including WebEOC, Sahana, E-team, Aid Matrix and IBM Rapid Response. All of them, with the exception of Sahana, are proprietary systems and can only be modified by the developers. Collabbit is licensed under the open source GPL Version 3 software license. The GPL ensures that end users will be free from vendor lock-in and provides the latitude to customize the code without penalty.

Collabbit recognizes that in a loosely coupled multi-agency environment each agency may have a preference for tools for data collection and analysis. It provides a unified communications platform that allows various working groups to maintain independence and custom methodologies. It simply provides a platform to share the results. For example, an animal welfare organization prefers to share their periodic update using, Microsoft Word; a disaster assessment team might upload a map generated from an unique internally developed geographic information system; while a Emergency Response Team might prefer to upload PDF reports..

During a disaster scenario Collabbit must be accessible from many different locations to accommodate many different user circumstances. The software's web-based design accommodates multiple dispersed users with a variance in technical skill and technology tools. The target audience is any collection of loosely joined organizations that want to communicate asynchronously during emergency situations. These organizations often have informal ties with no central authority.

FROM CONCEPT TO ACTUALIZATION

Collabit's first users came from the multi-agency, New York City Voluntary Organizations Active in Disaster (NYC VOAD), a local chapter of a national coalition of voluntary organizations and government agencies united to improve human services delivery during disasters and disaster recovery. VOAD agencies assist affected individuals by helping them find emergency shelter, reliable food and water sources, professional mental health assistance, disaster recovery aid, and many other relief resources and services. NYC VOAD employed the first iteration of Collabbit, at the time called Virtual EOC, during a table-top exercise in June 2009. In November 2009, the Salvation Army in New York City utilized a rebuilt and enhanced Collabbit to

assist in its effort to deliver 10,000 Thanksgiving turkey dinners at food centers throughout the New York City region. The design and deployment process for both deployments is described below.

NYC VOAD Exercise

The project, originally titled Virtual EOC, was developed based on a need identified by members of the NYC VOAD group brought to the attention of the Humanitarian FOSS Project by the 2nd author. The initial problem as stated was "reduce the amount of time that involved parties must gather together on a conference call when individuals are only interested in a small segment of the conversation, with an ability to maintain a record of outcomes and decisions made by various groups".

Collabbit was proposed as an online collaboration tool to provide a common space for a multi-agency community to share information in a targeted manner, while maintaining transparency of the information being shared. The development team proposed a custom application to meet the problem statement. The development team converted the loose outline and hand drawn concept drawings into a technical specification document through iterative discussions with NYC VOAD users.

Five undergraduate programmers began developing Collabbit on May 25, 2009. They worked full time to the targeted launch date of June 11th 2009 for the NYC VOAD tabletop exercise. A functional prototype was delivered in 21 days of the initial request. The June 11th, 2009 NYC VOAD Tabletop Exercise was designed to improve collaboration among the diverse group of agencies to delineate responsibilities during the recovery from a major hurricane in the New York Metropolitan Area. Over two dozen non-profit and government agencies participated in the exercise, including The American Red Cross in greater New York, Catholic Charities, City Harvest, Interfaith Disaster Services, the Clothing Bank, F.E.G.S., New York Cares, The Legal Aid Society, Neighborhood Housing Services, World Vision, FEMA, New York City OEM and numerous observers. Participants grouped together into multiagency human service planning committees covering common issues like immigration, sheltering, mass care, animals, housing recovery, and much more. All together there were *18* distinct committees and *over 90* individuals involved in this collaborative exercise.

Exercise participants were presented with human services issues seen in past disasters. The committees were tasked with determining which committee was responsible for the issue and creating basic steps to a successful resolution. After a short break, the groups presented their justification for accepting one of the issues and their resolution.

The Collabbit proof-of-concept prototype was utilized to track and log the various decisions and outcomes of the table top exercise (Campbell 2009). Following the exercise, volunteer members from VOAD assisted the development team with further development of the prototype. They tested the application and gave continuous expert feedback to ensure the program met their needs. Based on this feedback, the development team rebuilt the software from the ground up using Ruby on Rails (a change from PHP). The revised technical design, based on requirements generated by the users, improved security, robustness, data portability and system maintainability.

Figure 1: VOAD disaster exercise in New York City

Observations and Feedback Summary

- Collabbit's success was directly attributable to its simplified user interface. The level of computer proficiency and technical understanding varied greatly in the targeted user base.
- Collabbit's small feature set was easy to learn and was more likely to be used than more sophisticated and feature-rich alternatives.
- The rapid code-release development life cycle allowed developers and users to iteratively and quickly identify capabilities that worked and didn't work for them.

- Two-way messaging via email and SMS text messages is needed to un-tether users from the desktop. Additionally, Collabbit must support an internal and secure chat and provide personalized RSS feeds for users.

Thanksgiving Salvation Army Deployment

In a second deployment, the Salvation Army of NYC used Collabbit on Thanksgiving Day to share situational awareness and best practices at ten food distribution sites across the New York Metropolitan area to coordinate a mass feeding for approximately *10'000* people (Blog Post 2009). Salvation Army exercise participants were split into three roles: Site Coordinator, Roving Coordinator, and Command and Control Coordinators. Site Coordinators were responsible for maintaining operations at individual distribution sites. They worked directly with facility staff, the caterer, volunteers, and the dinner guests.

Roving Coordinators had responsibility for multiple distribution sites. They needed to maintain constant situational awareness of their multiple sites even while driving between sites. Command and Control Coordinators monitored all personnel and all feeding operations. They were responsible for balancing personnel and supplies between the different sites. Command and Control therefore needed to have the widest understanding of all operations.

All coordinators were given access to Collabbit. The developers assisted the Salvation Army with the initial setup. Each coordinator signed up for groups according to their role. Site Coordinators joined a group associated with their assigned facility. Roving Coordinators joined the groups for their assigned areas and also a group designated for all Roving Coordinators. Command and Control subscribed to all feeds. All together, there were *13* distinct groups consisting of *24* individuals collaborating on this project. Half an hour before the feeding started, each group was required to begin posting regular status updates representing their assigned area of responsibility. Each update was associated to their role or site.

The exercise participants in all roles depended upon Collabbit's ability to filter update messages. Users were able to restrict their view to the groups they subscribed to while having the option of seeing all updates. Participants were able to easily separate the information they needed from the extraneous updates of no interest to them. The roving coordinators found the SMS text alerts and email particularly helpful as they could stay informed even when moving to another facility.

Observations and Feedback: Salvation Army Mass Feeding Exercise

- Mobile customized incident updates were very useful to all exercise participants.
- Custom feeds allowed users to focus on their immediate responsibility. Users still had access to all updates and could gain a wider situational awareness.
- The small feature set was easy to learn and no problems were encountered.
- A two-person Collabbit support team was on hand and responded quickly to technical issues that arose during the event.

CHALLENGES

Overall feedback from the user community has been overwhelmingly positive. However we have identified possible challenges to the long term sustainability and adoption of the project. Non-profit and government organizations may not have technically proficient staff to handle backend configuration and support nor an environment permissive of open source software (Waring and Maddocks 2005). Collabbit attempts to mitigate this by providing organizations both hosted and standalone versions of the tool. An organization could take on the role of sponsor, supporting a deployment for multiple organizations. Collabbit and similar applications will require broad organizational support and a willingness to adopt these applications as many employees aren't willing to spend time learning new tools. Collabbit's intuitive user interface attempts to reduce the required training necessary to get up and running with the system.

The culture of free and open source software relies heavily on the voluntary contributions of a distributed ad-hoc community. To be successful, this community must be identified and held together. This is not an easy task. Hence, coordinating an ever changing diverse community is a challenge in creating a sustainable software project. For Collabbit to be sustainable software it will need to build and maintain an enthusiastic community of contributors and collaborators and identify funded avenues to maintain development and support on a continued basis. In a loosely-coupled multiagency environment there may be political, privacy, or legal barriers preventing the adoption and use of Collabbit.

FUTURE

The Collabbit project will maintain its tight focus on its limited scope and targeted niche of loosely bound multi-agency coordinative bodies, relying on continued feedback from the users in refining the application further. Potential avenues for improving existing functionality include: (1) Improvements with back end infrastructure with an eye to extensibility for large user bases, (2) Multi-lingual support or transliteration, (3) Real time integrated chat, and two way SMS text message and email updating. These features will be introduced into a packaged release only when users have concluded that their work flow has been improved by the addition.

CONCLUSION

Collabbit is not only a software system but a partnership between the users and the development team. Users rely upon the development team to fix problems and extend features. Users must trust the development team to act in their best interest by following coding practices and insisting on system transparency.

The Collabbit project demonstrates that the needs of VOAD and similar collections of collaborating organizations can be met adequately by a fairly simple system. Collabbit is an attractive application because: (1) it users a simple, accessible design that focuses on facilitating the communication and record keeping needs of collaborating individuals and organizations; (2) it employs a nimble, user-driven design and development approach.; and (3) it is available through a free and open software license.

ACKNOWLEDGMENTS

The work was supported in part by the National Science Foundation under grants number 0722137, 0722134, 0722199. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

1. Aljazeera. *War on Gaza.* December 2008. http://labs.aljazeera.net/warongaza/ (accessed January 14, 2009).
2. Ariyabandu, Rajindra. *Free and Open Source Software for Disaster Management: A Case Study of Sahana Disaster Management System of Sri Lanka.* UNESCAP Technical Paper, 2009.
3. Blog Post. *Collabbit used by NYC Salvation Army to help serve 10,000 Thanksgiving dinners.* November 30, 2009. http://blog.hfoss.org/?p=183 (accessed january 14, 2010).
4. Campbell, MacGregor. "Aid agencies turn to open-source software." *New Scientist magazine - 03 October 2009*, October 03 , 2009: p23.
5. Jefferson, T.I. and Harrald, J.R. "Collaborative technology: providing agility in response to extreme events." *Int. J. Electronic Governance*, 2007: Vol. 1, No. 1, pp.79–93.
6. JustPartners, Inc. *Creating an Online Lifelife: A Web Portal for Emergency Preparedness & Disaster Recovery.* Annie E. Casey Foundation, 2008.
7. Morelli, R, C de Silva, T de Lanerolle, R Curzon, and X Mao. "Helping Our Neighbors in China: An International Collaboration to Deploy Sahana in Chengdu, Sichuan." *Communications of the ACM*, forthcoming.
8. Morelli, R., Tucker, A., Danner, N., de Lanerolle, T., Ellis, H.J.C., Izmirli, O., Krizanc, D., and Parker, G. "Revitalizing Computing Education by Building Free and Open Source Software for Humanity." *Communications of the ACM,* August 2009: 67-75.
9. P. Currion, C. d. Silva, and B. Van de Walle. "Open source software for disaster management." *Communications of the ACM 50(3)* (Communications of the ACM), 2007: 61–65.
10. Samoladas, Ioannis, and Ioannis Stamelos. "Assessing Free/Open Source Software Quality." *Free Open Source Research Community.* 2003. http://opensource.mit.edu/ (accessed January 14, 2010).
11. Waring, Teresa, and Philips Maddocks. "Open Source Software implementation in the UK public sector: Evidence from the field and implications for the future." *International Journal of Information Management 25, no. 5, (October 1),* 2005: 411-428.