

A Platform for Coordinating Voluntary Helpers in Disaster Response

Matthes Rieke
52°North GmbH
Martin-Luther-King-
Weg 24
Münster, Germany
m.rieke@52north.org

Sebastian Drost
52°North GmbH
Martin-Luther-King-
Weg 24
Münster, Germany
s.drost@52north.org

Simon Jirka
52°North GmbH
Martin-Luther-King-
Weg 24
Münster, Germany
jirka@52north.org

Hans Betke
Martin Luther University
Halle-Wittenberg
Universitätsring 3
Halle (Saale), Germany
hans.betke@wiwi.uni-
halle.de

Abstract

This paper introduces a system architecture concept for coordinating voluntary helpers in disaster response activities. In recent years, many examples of natural disasters have shown how people coordinate their voluntary help via social media platforms. However, these help efforts are usually not coordinated with the activities and requirements of public authorities. This often leads to inefficient and in some cases even counterproductive activities. To solve this issue, this paper presents ongoing work on a corresponding system architecture and discusses how it can be practically applied in a disaster scenario.

Keywords: Disaster Response, GIS, Social Media.

1 Introduction

Disasters, for example large-scale flooding events, have a significant impact on public life and require comprehensive action by security authorities and disaster response organizations. In the past, such disaster response actions were organized by public authorities. However, recent events, e.g. the Elbe flooding in Germany in 2013, have shown that social media communities offer a complementary source of help and information (Kaufhold and Reuter, 2016). On the one hand, spontaneous volunteers from the local population are becoming motivated to help in disaster response activities and on the other hand, social media platforms help to collect and share information about needs for help.

However, as a result of this development, there are independent eco systems in which the help in case of a disaster situation is coordinated and delivered. This results in efficiency issues and can lead to situations in which the presence of voluntary helpers may even become counterproductive to the efforts undertaken by public authorities. For example, voluntary but inexperienced helpers may enter endangered areas so that further efforts may be required by public authorities to rescue these helpers. Thus, by ensuring a good coordination of spontaneous unaffiliated on-site volunteers (SUV) by public authorities, the impact of their help will be further increased (Lindner et al., 2017). Therefore, we developed a sophisticated technical solution that not fully automates but provides significant support in the management of SUV.

There are several related research activities which shall be mentioned in the context of the work presented in this paper. Several researchers have already proposed and analysed different approaches for coordinating the support provided by volunteers (e.g. Whittaker et al., 2015; Lorenz et al., 2017). While this work is highly relevant for the development of the underlying algorithms and workflows, the focus of this paper is directed on the underlying platform architecture. Thus, the

concept presented in this paper can be used for supporting the implementation of strategies to coordinate volunteers in a disaster situation.

Other technical solutions are focused on notifying the public in case of crisis situations. In Germany there are especially two developments which shall be mentioned: Katwarn¹ and NINA². Both applications are used by different public authorities in Germany to warn the population in case of a disaster. However, while they reach a broad audience their focus is on alerting and not on coordinating disaster response among volunteers.

Finally, geographic information systems (GIS) play an important role in disaster response activities. A good example is the collaborative collection and use of map data in regions that were hit by a disaster (Meier, 2012). Consequently, the integration of GIS technologies is an important factor in the design of the presented system architecture.

The following section 2 will further introduce the developed system architecture concepts. Subsequently, section 3 will discuss how the proposed approach was implemented and demonstrated. Finally, section 4 summarizes the findings and provides an outlook on future work that was identified.

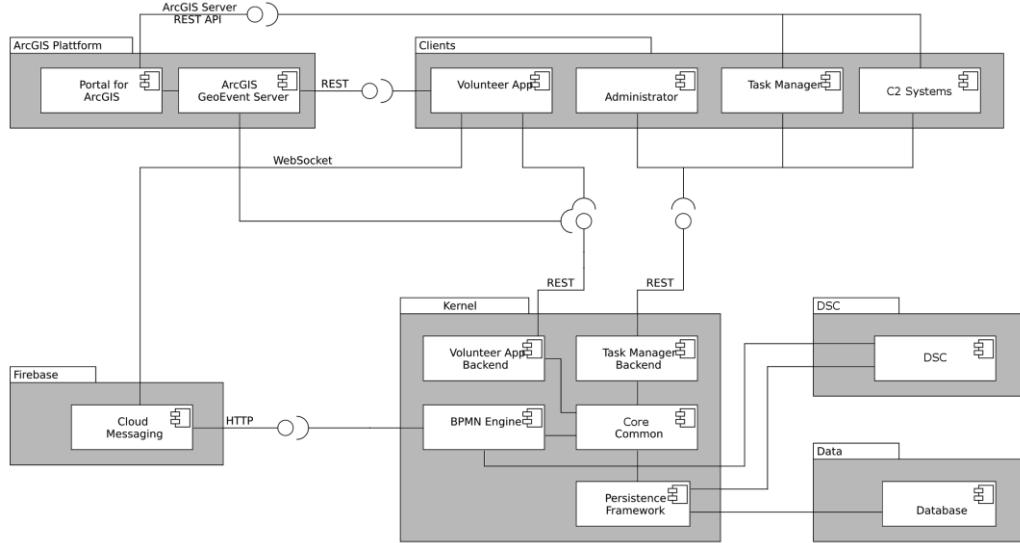
2 System Architecture

This chapter describes the technical architecture of the platform. A brief overview, introducing the different components and modules, is followed by the illustration of the domain model and a detailed description of the individual components and their responsibility.

¹ <https://www.katwarn.de>

² https://www.bbk.bund.de/DE/NINA/Warn-App_NINA.html/

Figure 1: Architecture Overview.



2.1 Overview

The platform is designed in a modular fashion and consists of several reusable components (Figure 1).

The kernel components, which are the backend of the system, contain the business logic and provide required data for different clients. They also communicate with additional third-party platforms. A lightweight REST API has been developed that enables various different clients to use the data. The kernel also contains the Business Process Model (BPM) Engine which executes the different sub processes (e.g. assigning help efforts or notifying volunteers) of the system. For implementing these business processes, we used the Open Source BPM Engine Camunda. Another central component of the platform is the Decision Support Component (DSC). It supports disaster control authorities by identifying appropriate volunteers for required tasks in consideration of their capabilities.

To store and handle relevant spatial information we utilized the ArcGIS Platform. We integrated several components of the platform to provide fundamental GIS functionalities for emergency responders that assess situational awareness.

The component for communicating with volunteers is the Cloud Messaging module. It provides platform independent functionalities to push notifications to mobile clients. This is used to notify volunteers of potential missions and updates to these.

Finally, the various clients complement the system. A mobile app for Android and iOS enables the registration of interested volunteers. Our system architecture is designed to support its integration into existing Command and Control (C2) systems. However, if absent, we alternatively provide a Task Manager app to deal with similar tasks.

2.2 Domain Model

In order to have a consistent definition across the different components of the system, an abstract domain model of the relevant real world concepts has been developed. This model features all relevant information that is required for the

automatic assignment of volunteers to certain tasks. It comprises three main areas (Figure 2):

- Situation model: the conceptual aspects that describe a certain disaster situation (e.g. a flooding, chemical hazards)
- Task model: a task entity describes specific activities that are planned to be executed by volunteers in the context of a specific situation
- Volunteer model: all information required to allow an assignment of volunteers to a specific task

Figure 2: Overview of the Domain Model.

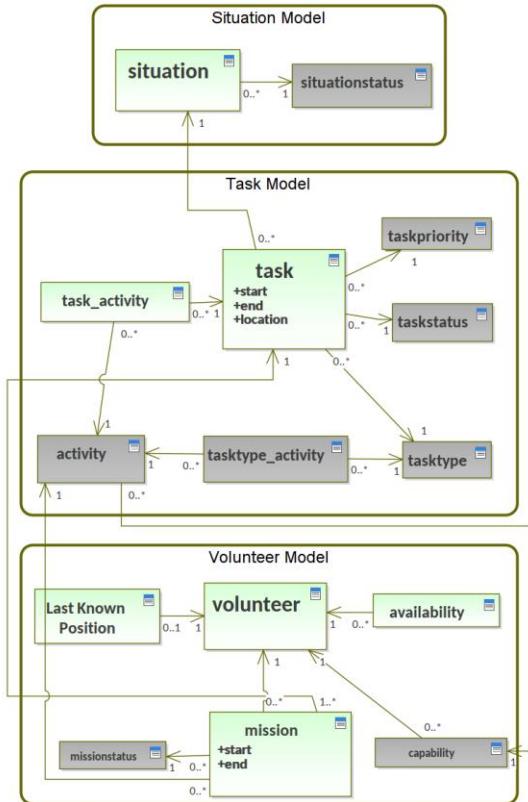


Table 1: Volunteer Capability Categories

Capability Name	Description
light_physical_work	Light physical work (e.g. messenger services)
medium_physical_work	Medium physical work (e.g. filling of sandbags)
heavy_physical_work	Heavy physical work (e.g. carrying of sandbags)
support	General supportive work (e.g. serving meals)
support_with_vehicle	Supportive work including driving vehicles
writing	Writing (e.g. documentation)

As a first step, the platform requires the instantiation of a situation, including its type (e.g. flooding, chemical hazards). A specific REST endpoint (Section 2.3.1) can be used by a C2 system for this purpose and.

Once a situation is available, the platform is ready to accept registrations of voluntary helpers. Via the Volunteer App (Section 2.3.5) interested persons provide basic information such as their home address and mobile phone number and their last known geo-location. Two additional information elements are modelled for each volunteer:

- Capabilities: a set of work categories that can be carried out by the individual (see Table 1)
- Availabilities: a set of time ranges during which the individual is available for help

If in a later state a volunteer is assigned to a specific activity of a task, a specific mission model is used to store the assignment. It allows the tracking of the acceptance of the mission by the volunteer using its status.

2.3 System Components

2.3.1 Kernel

The kernel module is the main business processing unit of the platform. It holds references to all other system modules (described in the following subsections). It provides a lightweight REST API as the interface to external components. The REST API provides a set of resources:

- Registered situations (only one can be active at a time)
- Tasks, including already finished tasks
- Registered volunteers, including their assigned missions
- All activities and those specific to a situation type (e.g. flooding)
- Capabilities and the mapping to activities

While the domain-specific resources (situation types, activities and capabilities) are read-only, the REST API

Table 2: Mapping Activities to Capabilities

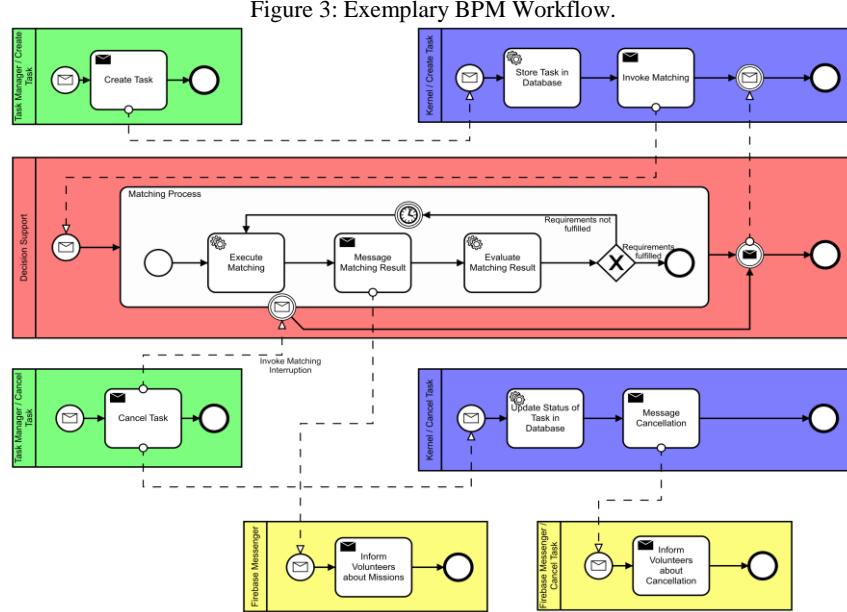
Activity Name	Description	Requires Capability
sandbags_filling	Filling of sandbags	medium_physical_work
sandbags_carrying	Carrying of sandbags	heavy_physical_work
information	Information sharing	light_physical_work
dike_monitoring	Dike monitoring	light_physical_work
driving_tasks	Driving tasks	support_with_vehicle
logistics	Local logistics	light_physical_work

In parallel to the registration of volunteers, new tasks can be created by the emergency task force. A task consists of a time window, a geo-location as well as a rendezvous point, a priority and most importantly the tasks activities. The activities define what kind of work needs to be carried out in the context of the task. Table 2 describes the set of activities applicable for a flooding scenario. Every situation type consists of a set of pre-defined activities that are relevant in the context of the situation.

Based on specific mappings the platform is able to assign volunteers to task activities based on their provided capabilities.

provides write functionality for tasks as well as volunteer missions. It therefore provides a view on and interaction functionality with the domain model as introduced in Section 2.2. The REST API acts as the integration interface for external C2 software systems. The platform is therefore easily integratable into existing software such as CommandX (Eurocommand, 2018) or DISMA (TÜV Rheinland, 2018).

Besides the REST API interface, an additional interface based on XKatastrophenhilfe (BBK - Bundesamt für Bevölkerungsschutz und Katastrophenhilfe, 2016), a standard of the German XÖV family of standards for public administration, has been designed. This XML based interface reuses many concepts already present in XKatastrophenhilfe and XÖV thus fostering the functionality of already established XÖV based systems in an interoperable way.



For every task that is created, a dedicated business process is executed within the BPM Engine (see below section). Within these processes, the other components are integrated into the workflow. In particular, the Decision Support Component is executed and its output evaluated, resulting eventually in the communication with the volunteers through the Volunteer App.

2.3.2 Decision Support Component

The Decision Support Component features the algorithmical core of the platform. It is responsible for solving the problem of assigning the best matching volunteers to the set of task activities. It is optimized in a way that respects a set of criteria (e.g. maximum work time in 24 hours for each volunteer, task priority) in order to find an optimal workload across the entire time frame of a task.

2.3.3 Business Process Model Engine

The main aspect of the platform is the support of emergency responders for coordinating voluntary help efforts. Volunteers should be integrated into crisis management processes without affecting emergency task forces immensely. Hence, we considered a process-based approach to handle the various requirements that are associated with the coordination of volunteers. The advantage of modelling a complex process as a business process is that it can be expanded, analysed, executed and monitored with supportive tools provided by Workflow Execution Engines. For modelling the different processes that have to be executed within the system we utilized the Camunda BPM Engine. First, we identified each required sub-process (e.g. running the DSC or notifying matched volunteers) and modelled the flow between these in a Business Process Model with the Camunda Modeler. An exemplary workflow model is illustrated in Figure 3.

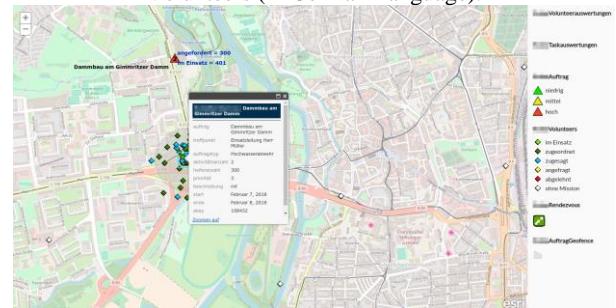
The procedure of abstract process modelling allows us, to review the whole process of our system in an abstract way. A

logical implementation of this model in Java has been integrated into the kernel so that it can be executed automatically. To monitor the running state of the processes and analyse these, the Camunda Cockpit provides appropriate functionalities. We consider to use this tool for evaluating the workflow of our system in the future.

2.3.4 Platform for Base Geodata

The system has to process different geodata such as location information for the task site or the current position of volunteers and visualize it in a way that they support emergency responders in their practical work. Therefore, we utilize several components from the ArcGIS platform to provide GIS functionalities for the platform.

Figure 4: Screenshot of a Situation Map with Visualised Volunteers (in German Language).



The above mentioned geodata are persisted in an enterprise geodatabase. We host different maps on Portal for ArcGIS that provide different views on the events at operation site (Figure 4). The portal allows a secure integration of geodata and GIS tools. In addition, the hosted situation maps are included within the Task Manager (Section 3) by the use of the ArcGIS API for JavaScript. This API also provides useful

functionalities for simple map interactions, e.g. for selecting the meeting point of volunteers.

The ArcGIS GeoEvent Server processes real-time position information which is transmitted by the clients that use the Volunteer App. Thus, the volunteers' positions will be updated continuously inside the database. Moreover, the intersection of the current volunteers' positions with operation area is calculated continuously to determine if a volunteer has arrived at operation site. So, the ArcGIS GeoEvent Server enables a real-time monitoring of different spatial events, which arise during crisis.

2.3.5 Volunteer App

We developed a dedicated mobile application with focussed functionality. It shall only provide means to register with the platform and provide all necessary information that is needed to assign the volunteer to a specific mission (Figure 5). In addition, it must feature the ability to accept or decline a mission. In order to support the volunteer in his mission, a rudimentary map has been included that displays the rendezvous point of the mission in combination with the current location of the volunteer. It is currently available on Android smartphones.

Figure 5: Volunteer App (in German Language).



2.3.6 Volunteer Push-Messaging

Once the DSC has finished its calculations and identified appropriate volunteers, these have to be notified. Since polling the REST API in small time intervals would not be very effective, we developed a push-based communication approach to provide the Volunteer App with up-to-date information. We decided to utilize Firebase Cloud Messaging (FCM) which is a well-established platform to implement push notifications for mobile clients. It is built and maintained on Google infrastructure thus providing a reliable and energy-

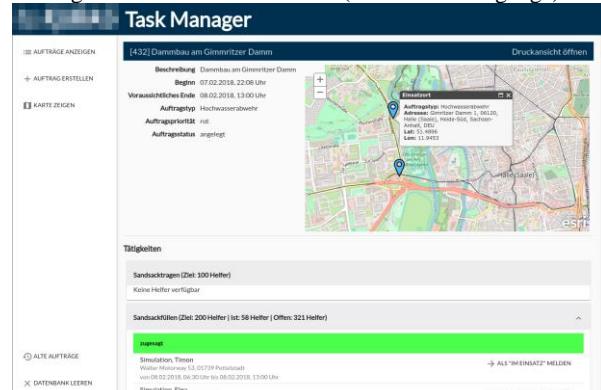
efficient (in terms of mobile phone battery consumption) connection between a server and several devices. In particular, it enables easily targeting of messages to devices in order to deliver customized notification messages. We integrated the server environment of FCM into the kernel component of the platform so that messages for volunteers can be constructed and sent based on the results of the Decision Support Component. Each matched volunteer receives a custom notification via the Volunteer App that contains the specific mission details and meeting place information. Once a volunteer has received such a notification, he can still decide if he accepts or declines the mission. This response is sent via the REST API back to the kernel and can be considered by the DSC.

3 Demonstration

The full stack of components contained in the overall platform has been demonstrated under laboratory conditions. A dedicated simulation software has been developed that allows the registration of thousands of voluntary helpers at the system. It also features the simulation of volunteers travelling to a rendezvous point by providing mocked GPS coordinates.

As the developments within the project are work in progress, no integration into existing C2 software has been achieved yet. We developed a lightweight browser-based application for demonstration purposes. The Task Manager allows to simulate all interactions required by an emergency task force. In particular, the creation of tasks and the monitoring of available volunteers are the main functionality (Figure 6).

Figure 6: Detailed Task View (in German Language).



The evaluation results are constantly analysed. Domain experts such as firefighter coordinators are regularly testing the Task Manager and the Volunteer App, providing valuable and practical feedback. The observations are used as an input to further improve the different workflows and decision algorithms.

4 Conclusion and Future Work

In this article we have introduced a platform that is able to support an emergency task force in its efforts to coordinate and task spontaneous unaffiliated on-site volunteers. The system comprises a manifold set of components all with dedicated responsibilities: a kernel module that provides a modern and lightweight REST API, the Decision Support Component as well as the messaging components and the base geodata infrastructure. By providing a dedicated mobile app, the platform allows volunteers to easily interact with the platform.

It is planned to support more interaction patterns for emergency task forces. The adjustment of existing tasks (e.g. expanding the time frame, complete cancellation due to external threats) is one of the main upcoming design efforts. It implies complex logic both internally (e.g. adjustments on the DSC) and externally (e.g. informing a volunteer about the updates).

At the current state the platform fulfils basic functionality for the interaction with the voluntary helpers. It relies on external personnel to promote its availability and benefit (e.g. through broadcasts, social media or newspapers). Sophisticated communication patterns such as automatic promotions via social media channels or the analysis of social network activity in the context of disaster situations have been identified as promising research areas.

The platform has been tested using simulated volunteers. The next step is to test the system under real-world conditions. It is planned to participate in an official disaster preparedness exercise. This allows the stress-testing of the different components both from a technical and usability perspective. The hopefully extensive feedback will be used to adjust the requirements and improve the overall system. The final results of the work in progress are planned to be reported in a dedicated follow-up publication.

Acknowledgement

This work has been funded by the Federal Ministry of Education and Research (Germany), BMBF, as part of the programme on “Erhöhung der Resilienz im Krisen- und Katastrophenfall”.

References

- BBK - Bundesamt für Bevölkerungsschutz und Katastrophenhilfe (2016) *Spezifikation XKatastrophenhilfe Version 1.1*.
- Eurocommand (2018) – CommandX. Available online: <http://www.eurocommand.com/en/comprehensive-solutions/commandx-all-inclusive.php>
- Kaufhold, M. A. and Reuter, C. (2016) The self-organization of digital volunteers across social media: The case of the 2013 European floods in Germany. *Journal of Homeland Security and Emergency Management*, 13(1), 137-166.
- Lindner, S., Betke, H. and Sackmann, S. (2017) Attributes for Simulating Spontaneous On-Site Volunteers. *Proceedings of the 14th ISCRAM Conference - Albi, France, May 2017*.
- Lorenz, D. F., Schulze, K., and Voss, M. (2017) Emerging citizen responses to disasters in Germany. Disaster myths as an impediment for a collaboration of unaffiliated responders and professional rescue forces. *Journal of Contingencies and Crisis Management*.
- Meier, P. (2012) Crisis mapping in action: How open source software and global volunteer networks are changing the world, one map at a time. *Journal of Map & Geography Libraries*, 8(2), 89-100.
- TÜV Rheinland (2018) DISMA – Disaster Management. Available online: <https://www.tuv.com/world/en/disma-%E2%80%93-disaster-management.html>
- Whittaker, J., McLennan, B. and Handmer, J. (2015) A review of informal volunteerism in emergencies and disasters: Definition, opportunities and challenges. *International journal of disaster risk reduction*, 13, 358-368.