Towards Automated Individual Communication for Coordination of Spontaneous Volunteers

Sophie Gerstmann
Chair of Information Management
Martin Luther University, Halle, Germany
sophie.gerstmann@student.uni-halle.de

Hans Betke
Chair of Information Management
Martin Luther University, Halle, Germany
hans.betke@wiwi.uni-halle.de

Stefan Sackmann
Chair of Information Management
Martin Luther University, Halle, Germany
stefan.sackmann@wiwi.uni-halle.de

ABSTRACT

In recent years, spontaneous volunteers often turned out to be a critical factor to overcome disaster situations and avoid further damages to life and assets. These Volunteers coordinate their activities using social media and mobile devices but are not integrated in usual command and control structures of disaster responders. The lack of professional disaster response knowledge leads to a waste of potential workforce or even dangerous situations for the volunteers. In this paper, a novel approach for a centralized coordination of spontaneous volunteers through disaster response professionals while using popular communication channels esp. messaging services (e.g. Facebook Messenger, WhatsApp) is presented. The architecture of a volunteer coordination system focusing on automated multi-channel communication is shown and the possibilities of a universal chatbot for individual assignment and scheduling of volunteers are discussed. The paper also provides first insights in a demonstrator system as a practical solution.

Keywords

Spontaneous volunteers, chatbot, social media, system architecture

INTRODUCTION

The growing number of disaster incidents in the recent past represents a new challenge for many societies and reveals new dangers for life and assets (e.g. Munich Re, 2018). Another problematic phenomenon for disaster response is the decreasing number of honorary volunteers in many security authorities and organisations (SAO) (e.g. McLennan et al. 2016). On the other hand, new technologies such as social media and mobile devices allow an even faster and more direct communication from SAOs to the people than ever before. This reveals new possibilities to use the potentials of unaffiliated citizens for provision of workforce and material in critical situations (Sauer, 2014). These spontaneous unaffiliated volunteers (SUV) do already play an important role and were crucial to overcome disasters, for example the Middle European Flood in 2013 (Thieken et al., 2016).

However, it is currently challenging for SAOs to access this new potential due to differences in the organizational structure and especially a missing direct, individual communication channel (Whittaker et al., 2015). Whereas SUVs use popular tools such as Facebook or Twitter to organise themselves, official responders have separate command and control systems and communication channels. The usual communication between these two groups is mostly conducted with mass media calls from officials or deployment of emergency call centres. Although sharing of information between civilians and SAOs in general makes great progress with tools such as Ushahidi or several warning apps, a direct communication for a goal oriented coordination and exploitation of the spontaneous willingness to help in a disaster situation has not yet been established.

In recent disaster events, the self-organisation of SUVs often led to situations where official responders were overwhelmed by spontaneous converging volunteers. Coordination of these unexpected large crowds of people can be very challenging especially during critical response activities or under dangerous circumstances. Deficits in volunteer coordination can not only lead to a waste of workforce but also decrease motivation to offer help in future events or even cause improper and counterproductive actions of unguided, enthusiastic non-professionals, necessitating further response efforts or even cause new damages (Barsky et al., 2007).

To overcome this situation approaches such as Betke (2018) propose so called volunteer coordination systems (VCS) that provide a direct communication between official responders and SUVs for a goal-oriented management of volunteering offers and demands. A crucial feature of this kind of systems is the possibility to take the individual circumstances of each volunteer into account. This refers to the individual communication behaviour by supporting a variety of popular communication systems as well as to the individual volunteering offer in terms of e.g. time and capabilities. Since the necessary information would exceed the capabilities of official responders, we suggest a (semi-) automated solution.

In this paper, we take up the idea of the VCS and try to answer the question: how can chatbots be used to improve the coordination of spontaneous unaffiliated volunteers? Therefore, we discuss the idea and architecture of an automated multi-channel chatbot as communication interface for SUVs and VCS. This work is part of the research project KUBAS in which different aspects of a VCS are investigated to develop a suitable holistic system. The project follows the design science research process of Peffers et al. (2006) whereas papers such as Betke (2018) address the first phases, this paper addresses the design and development phase (phase 3). In the second chapter, we derive general requirements for a suitable coordination interface for SUVs in a volunteer coordination system. Then we take a short look at chatbots in general and their current application in disaster response in search of a possible existing solution. Chapter three comprises the discussion of new volunteer chatbot architecture with respect to the former derived requirements followed by an insight in an early demonstrator. The paper closes with a summary and outlook for the next project steps.

REQUIREMENTS FOR INDIVDUAL VOLUNTEER COORDINATION

To design a suitable communication interface for SUVs we first need to define the specific requirements. Betke (2018) presented user stories of volunteers and official responders to describe the functional demands of both user groups of a VCS. However, we excluded official responders' side since they did not address the volunteer communication. Instead, we take a closer look on the volunteer user stories to derive the specific requirements for a volunteer-coordination-interface (VCI) between SUVs and SAOs respectively the intermediate VCS. As a typical human-computer-interface, the VCI is a frontend module with the purpose to present, receive and transfer information.

Besides the functional requirements discussed below there are two universal demands on the VCI, whereby the first one is to include as many popular communication systems as possible to reach the maximum number of possible volunteers and the second is an automation of the communication as far as possible. The following table (Table 1) comprises the original user stories and a short explanation of the derivation of each functional requirement (R1-R13). The requirement have been discussed in a workshop with a complete crisis committee staff consisting of six staff members.

Table 1. Volunteer Communication Requirements

	User Story	VCI Requirement
R1	*As a volunteer, I want to be informed about relevant help requests so that I do not have to search for them.*	This user story implicates the need for an automated and individual notification about a relevant mission for the volunteer.
R2	*As a volunteer, I want to get detailed information about help request so that I can make a decision to help.*	The information about help requests must be presented in an appropriate way. The VCI needs access to all data important for the decision making of the volunteer. For instance, these may include mission site, period and type of mission.
R3	*As a volunteer, I want to see my volunteer mission details at any time so that I do not lose any information.*	Information received via the VCI needs to be persistent. It must be accessible and provided at any time so that there are no interruptions in the information flow.
R4	*As a volunteer, I want to provide sufficient information about my*	The VCI must record and retrieve completely and properly all the input data of the volunteer. A processing of different data

	volunteering offer so that I can get the best help requests.	types must be possible.
R5	*As a volunteer, I want to group with my friends so that we can work at the same sites.*	An opportunity to find and interact with friends must be given by the VCI. This can be realized by either using information from existing networks (such as Facebook or WhatsApp) or an underlying VCS. The feasibility and implementation of this requirement depends on the specific data protection laws of each country in which the system is used.
R6	*As a volunteer, I want to change my volunteer offers so that it matches my current circumstances.*	The VCI must provide a function to re-query specific data sets at any time and induce the overwriting process of obsolete data in underlying coordination system. This requirement aims at repeatability of R4. The processing and replacement of data however is a not a feature of the VCI but of the underlying VCS.
R7	*As a volunteer, I want to get information if my volunteering offer was recognized to be sure it will be taken into account.*	The VCI shall give a confirmation to the volunteer once his data was successfully send to the VCS.
R8	*As a volunteer, I want to be able to get offers from more than one request so that I can help at different sites.*	The VCI must be able to present information of a number of requests or accepted missions of a volunteer. This requirement refers to R2 and R3 and implies some kind of navigation through the available information. However, the background processing of data and matching of suitable requests /missions is not part of the VCI.
R9	*As a volunteer, I want to use mobile devices so that I am not bound to a single location.*	The VCI must be designed for deployment on mobile devices of as many different types as possible.
R10	*As a volunteer, I want to use different devices so that I can be sure that I get help request information.*	For the communication between volunteer and the coordination system a multitude of devices (e.g. smartphone, mobile phone, computer) and channels must be addressed. A great variety of devices and channels allows a great participation of potential volunteers. This also concerns universal design aspects as people can communicate and offer help according to their abilities.
R11	*As a volunteer, I want to get information about disaster incidents in my home area so that I can prepare myself.*	Besides features regarding the communication regarding volunteering, the VCI must also serve for information purposes, for instance to inform about critical incidents at the volunteer´s location.
R12	*As a volunteer, I want to share on-site information with disaster authorities so that I can report potentially dangerous incidents.*	The VCI must allow submission of disaster response relevant information. The transfer and presentation of information to disaster authorities is realised by VCS. In connection with a variety of supported communication channels, this requirement may also serve as a connection between physical and digital volunteers whereby the former send information from their mission site for analysis by the latter.
R13	*As a volunteer, I want to get information about the traveling options so that I arrive on site as fast as possible.*	The VCI must provide some kind of routing feature to the volunteer, e.g. via external applications and display this information appropriately.

The derived requirements may not be exhaustive but constitute a good basis with cornerstones for the development of a useful communication interface for SUVs. As stated in Betke (2018), most of the presented requirements, such as local independence or a persistent information flow, can be realised by a mobile app. However, the sole application of a mobile app restricts the communication with volunteers in many ways. On the one hand, the potential user group is limited to owners of devices such as smartphones and tablets, excluding people with other devices such as simple cell phones or landline telephones. On the other hand, a stable broadband connection or wireless network must be accessible to guarantee the functioning of the app. Since this cannot be guaranteed in every disaster situation the usage of alternative communication channels should be taken into account. Apart from that, it is very important to include popular communication platforms especially social media, which recently became crucial for self-organisation of volunteers (Kaufhold and Reuter, 2016). To reach as many volunteers as possible and to utilise the full potential volunteer offers, we propose the implementation of a universal chatbot as VCI and discuss this in the following.

CHATBOTS IN DISASTER RESPONSE

Chatbots, also often referred to as conversational agents or virtual assistants, are software agents that enable an intelligent and individualised text-based communication between human and computer-based systems. As an extension to simple text-based dialogue systems, chatbots are able to understand, process and respond in natural language. Therefore, they are able to store the user input and respond to the request in the current context (Androutsopoulou et al., 2018; Ciechanowski et al., 2019). Present chatbot frameworks and platforms often offer a multi-channel integration (e.g. SMS, social messenger services), which allows the developer to design a single chatbot and let it interact on various channels. A further benefit of chatbots based on these frameworks is the lower effort for maintenance and adjustment. Another general advantage of chatbots is the high level of user-friendliness. Whereas app-users have to download and install the volunteer app to register for missions, chatbot-users can use a pre-installed messenger service, for example Facebook or WhatsApp, or simply communicate with the chatbot via SMS. Furthermore, the menu prompting of an app sometimes can be confusing. A chatbot however, leads the user through a dialogue requiring less technical comprehension and so also opens the system for further user groups, for example illiterate persons via text-to-speech technology. Chatbots can instantly respond if the user’s entry was incorrect or unclear. Thus, users can always be aware that their input has been accepted. In contrast to an app, the communication between user and system can be designed personally and human-like. Due to the human-like behaviour, possible fears of using computers can be reduced (Zamora, 2017; Muir, 1987). The direct interaction allows monitoring of the users´ emotion, e.g. by analysing the choice of words to get insight into their satisfaction with the chatbot. Several studies have been conducted to analyse the user’s perception of a chatbot. According to Zamora (2017), users consider the application of their own natural language and short interaction times as further advantages of chatbot communication. To what extent a chatbot is perceived by users as beneficial depends on its ability to lead human-like conversations and process and respond in natural language (Shawar and Atwell, 2007; Wuenderlich and Paluch, 2017).

Because of the previously mentioned benefits, chatbots take over the communication with customers in a rising number of business applications. In banking, finance and insurance sectors they take over tasks such as checking the balance or block the credit card in order to prevent a fraudulent use (Dole et al., 2015; Eling and Lehmann, 2018; Koetter et al., 2018). In the tourism sector chatbots inform about the cheapest flight available or book a table in a restaurant (Ivanov and Webster, 2017; Nica et al., 2018). They assist customers in finding the right product on e-commerce websites or answer product related questions (Johannsen et al., 2018; Cui et al., 2017; Thomas, 2016). In social networks social bots automatically communicate with users and are criticised for manipulating their opinion on political issues (Ferrara et al., 2016; Wagner et al., 2012).

In order to provide an overview on the current state of research devoted to the use of chatbots in disaster management, we conducted a literature analysis. For this purpose, the several scientific online databases, of ScienceDirect, ISCRAM and AIS as well as Google Scholar have been searched for the keywords “chatbot“, “bot“, “conversational agent“ and “virtual assistant” in combination with “disaster“, “crisis“, “catastrophe“ and “volunteer”. The identified relevant literature gives evidence that chatbots have the potential to improve the communication between public authorities and the civilian population. They are especially qualified to provide clarification on complex issues and for information purposes in civil protection. (Androutsopoulou et al., 2018). Currently, chatbots are mainly used for decision support and information retrieval in the phases of disaster preparedness and response. Thus, Augello et al. (2016) propose a model where a chatbot is used to warn decision-makers and civilian population of impending environmental catastrophes and support them in decision-making. For this purpose, the chatbot collects information from various sources. Sensors at measuring stations provide data to predict the future development and describe the current status of a disaster. In addition, the chatbot uses input information about users, for example their current location. This information is combined to provide an individual situation aware risk assessment of the user situation and offers advisory function. Sermet and Demir (2018) follow a similar approach. They developed an intelligent system using an ontology based knowledge engine to improve societal preparedness for flooding with help of voice recognition, artificial intelligence and natural language processing. The chatbot uses information from an official flood information system

Besides these scientific approaches, chatbots are already used in practice. For instance, in the U.S. the Richter chatbot assists users in preparation for an impending earthquake or tsunami (BotList, 2018). It provides educational videos about how to behave in a critical incident, provides the users with documents to create their personal emergency plan and informs about the nearest public shelter and emergency services. In Japan, Bebot, a multi-language chatbot, sends alerts and information about nearby evacuation areas to visitors (Japan Today, 2018).

The chatbots presented already meet some of the requirements from chapter 2. For example, the chatbot of Sermet and Demir is able to process natural language, which ensures that all user input is retrieved properly (R4). Almost any of the presented chatbots is able to inform the user on disaster incidents, which meets requirement R11. Nonetheless, the main intention of chatbots presently used in disaster management is merely information retrieval for improving personal safety. Therefore, existing approaches have a different target audience and none of them takes coordination of volunteers into account. Since the literature analysis did not reveal a suitable chatbot solution as communication interface for SUVs, we will discuss the design of an appropriate software artefact serving this purpose in the following chapter.

VOLUNTEER COORDINATION SYSTEM WITH MULTI-CHANNEL COMMUNICATION

As stated before, the purpose of the chatbot proposed in this paper is to serve as a communication interface between SUVs and a VCS, which processes data from volunteers as well as SAOs to support an optimal usage of the volunteering potential. We therefore designed the chatbot as one module of a holistic VCS. The respective VCS architecture extract depicted in Figure 1, consists of a communication subsystem comprising the chatbot, a management engine and interfaces for communication devices of the volunteers on the one hand and official on-site responders on the other. On site responders or crisis committees, send their requests to the coordination system, which are then handled by the management engine. SUVs send their offers through the channel of their choice, which are attached to the chatbot via web interface. The chatbot takes over the communication with the volunteer. Based on natural language processing (NLP), the chatbot detects the user intents and induces the management engine to process their request. Hereinafter the single modules of our proposed architecture are described.

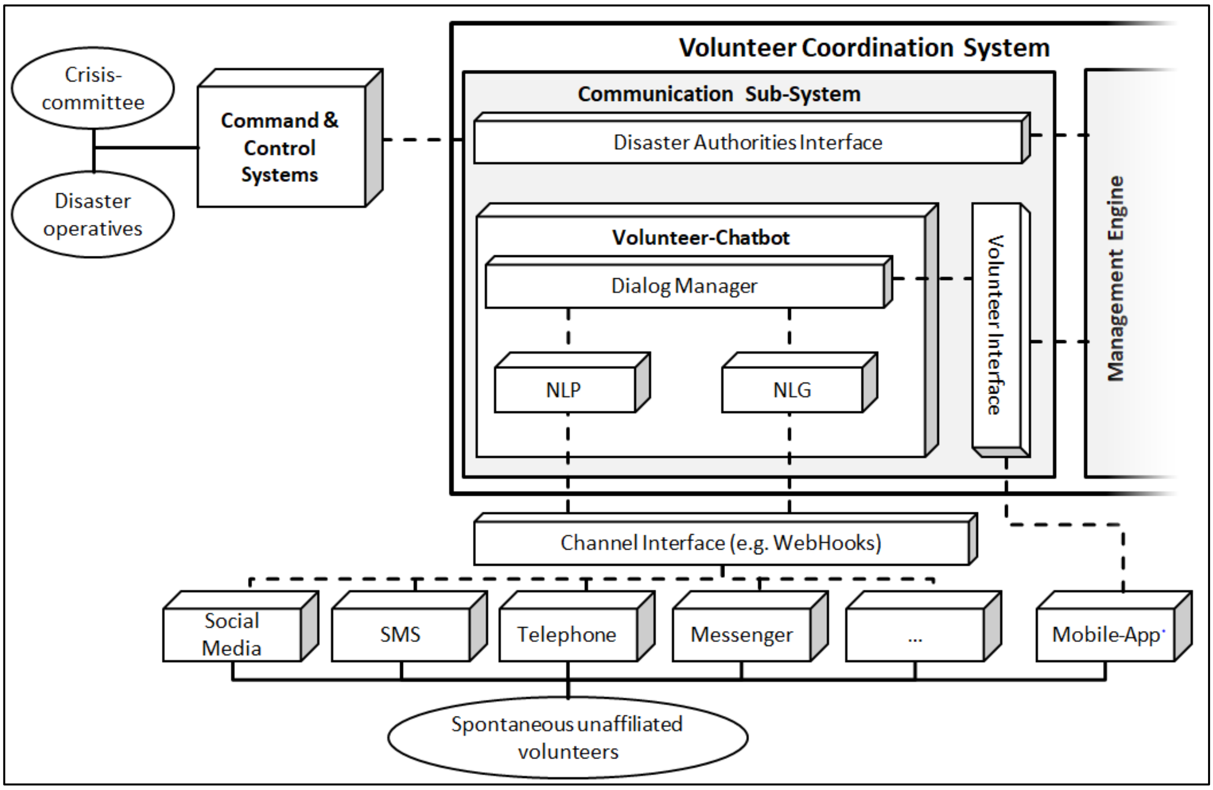

Figure 1. Volunteer Chatbot Architecture

The management engine in the proposed architecture consists of a workflow engine, which manages all in - and outgoing information as well as the interaction of all further system components. It comprises the database where all information about volunteer offers, help requests and volunteer user data is stored (R6). The engine monitors the individual status of each volunteer (e.g. offers, mission status, location, communication channel). Another major component connected to the management engine but not displayed in figure 1 is the decision support module whose task is to constantly analyse the situation on volunteer offers and help requests to determine matches and provide information about which SUV should be alarmed for open help requests. For a detailed description of the concrete architecture of the coordination system and its modules see Betke (2018).

SUVs communicate their information, for instance registration information, volunteer offers etc. through the chatbot. The chatbot can be called via multiple channels, for instance SMS, Skype or messenger services such as Facebook messenger or WhatsApp (R9, R10). In our proposed architecture these channels are connected to the chatbot via webhooks. If the user's input is matched with a defined event, a callback function will be evoked and a HTTP – POST is sent to the web service, which returns the requested result (R3). To match a request or offer to the correspondent user, the assignment of a specific user id is required (R1). The user id can be extracted from the channel that is used by the user, e.g. account number of social networks, mobile number of messenger services or SMS or is generated and validated by the VCS.

To process user inputs and manage the dialogue between user and system, two solutions exist. The first is to use a simple, rule-based dialog system. To respond to the user input, these dialog systems refer back to pre-defined question-answer patterns. These patterns can be created by using AIML (Artificial Intelligence Markup Language), a derivative of XML. AIML patterns are grouped in intentional categories and contain templates with pre-defined answers. User input is matched against each pattern. Because of that, the chatbot is inflexible when it receives input, which has not been defined in a pattern. A chatbot using AIML is stateless and will not be able to store the conversation state and therefore is not aware of changing context. It can be assumed that volunteers would be unsatisfied, if the chatbot fails to handle their request properly. We therefore suggest the application of artificial intelligence instead of pattern oriented chatbots. Based on machine learning, these chatbots are self-learning and able to answer unexpected questions. In our proposed architecture, the dialog system consists of a natural language understanding (NLU) component, the dialog manager and a natural language generator (NLG). The NLU component is responsible for analysing the user input and triggering an action according to the user intent. For this purpose, intents have to be defined by the developer and the NLP component has to be trained on a variety of possible user input (R4) and chatbot output sentences. For example, to register for a mission a volunteer could ask "How can I register for a mission?" or simply "I want to help". Although the queries have a completely different syntax, both have the same intent (register for a mission). Based on machine learning the NLP component checks the query against every defined intent and matches it to the most suitable. To extract the specific parameters of an intent, for instance preferred date and time of a mission, entities need to be defined. Entities specify one or many actions of a single intent and represent a class of objects with values representing possible objects of this class. Another opportunity to train the chatbot for intent detection is the use of ontologies, which connect intents, phrases, nouns, adjectives and the relationships among them to a specific domain. For instance, if the user asks "How can I register for a mission?" the chatbot will parse the query and locate the register entity in the ontology. If the register entity is related to the time and date entity, the chatbot will ask the user to input their name to register for a mission.

After understanding the user intent, the chatbot needs to respond correctly and with the data that have been requested by the user. This is managed by the dialog manager. For this purpose, the dialog manager needs to maintain the context of the information and control the dialog or rather information flow. To execute the registration process, the chatbot needs to know all the relevant information, for instance user name, preferred mission site etc. If the input data of the user is unspecific and further information is needed, the chatbot needs to realize the missing data and request the user to fill in these slots. To avoid user input that cannot be processed by the chatbot or potentially leads to a misunderstanding and perhaps to cancellation of the registration process, we suggest designing a dialog, which is strongly guided by the chatbot itself. This can be done by providing the user with a set of predefined options. Thus, the user is led through the conversation by the chatbot and is always aware of what they say next. If there is nevertheless invalid user input, the chatbot will send a default message, implying that the input could not have been processed (R7). In order to fulfil the registration process, the dialog manager needs to pass the relevant data to the management engine. For this purpose, depending on the intents and entities extracted by the NLG component, a specific JSON code is generated and transferred to the management engine that can be called via REST API. To find an appropriate mission, the dialog manager sends the relevant data to the management engine and posts a request on available missions. The management engine will check the received data with the data sent by official responders and send these data back to the dialog manager (R8). Natural Language Generation is applied on constructing the answer containing the requested information on proposed missions. For this, the dialog manager uses templates and replaces the slots with the data retrieved from the management engine (R1, R2). If the user accepts the mission, this information needs to be stored in the database of the management engine and the dialog manager sends a confirmation to the user (R7). Besides registering and assigning for missions, users also required the opportunity to send on-site information to disaster authorities (R12). To realise this requirement, the chatbot needs to retrieve the needed information from the user by working through a series of questions determining the dimensions of the incident (e.g. What happened? Who was involved? When did it happen?). Again, the management engine manages the passing of this information to disaster authorities.

Furthermore, it was required for the communication system to have access to external applications. To provide general information on critical incidents at the SUV'S location, the communication system can be connected to a public warning system or retrieve information direct from official responders via the VCS (R11). The data is processed and preselected by the management engine, which triggers the chatbot to start a warning dialogue. To group with friends (R5), access to social networks is required, whereby the interface of social networks usually can be called via webhooks. Alternatively, the user can group with friends within the communication system by providing identifying information such as mobile number, sur- and last name. For on-site routing (R13), access to travel apps is needed, for instance map services or train connections. For this purpose, the chatbot needs a template, which can integrate and display the routing graphically or work as announcer for a navigation system.

SIMPLE CHATBOT DEMONSTRATOR

So far, we have implemented a first simple prototype chatbot on Facebook messenger via Chatfuel. Chatfuel is a bot platform for creating simple dialogs on Facebook messenger. We decided to use the Facebook messenger for a first exemplary implementation since it offers a mature API, large user base and is an important platform for volunteers. This chatbot is used as demonstrator to evaluate the technical feasibility and give a first impression on the user acceptance for this communication method.

The chatbot's ability to respond to user input depends on the amount of training examples connected to the specific intent. We generated training examples by searching a thesaurus for synonyms, for example "help", "register" and "volunteering". The training examples were supplemented in the course of time with the growing experience that we made with users testing the chatbot. To minimalize the risk of wrong user input, we designed the chatbot to suggest a set of options from which users can choose. In addition, it is possible to interact with the chatbot by putting own questions. Every dialog starts with the chatbot introducing itself and offering the user to either register for a mission, to get information on the VCS-system (called KUBAS in our example) or the possibilities of voluntary work in general. If the user wants to register for a mission, the chatbot then asks about their preferred work. If there is a suitable mission for the volunteer, the user is then asked to confirm their offer and will be sent a confirmation (Figure 2).

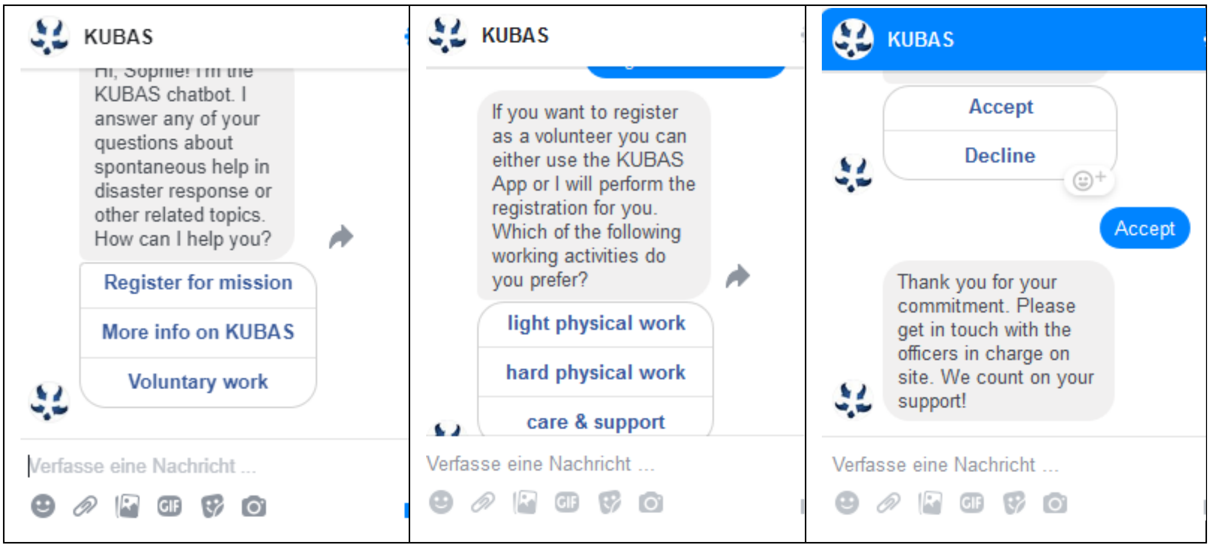

Figure 2. Chatbot Functions (Welcome message, Offering working activities, Confirmation of mission)

On an invalid input, the chatbot will send a default answer, informing that the user input has not been understood and referring to our KUBAS website for further information and assistance.

Our chatbot demonstrator has been presented to a specialist audience of disaster authorities and organisations at several workshops and got thoroughly positive feedback. Testers especially highlighted the integration in existing social networks, uncomplicated usage via natural language, direct feedback and reduced workload though automatisation. They mentioned a great potential for the application of chatbots in disaster response and volunteer coordination. However, at the present time the chatbot is not able to connect with the coordination platform, and therefore, volunteers are not able to register for a mission. In addition, we learned from user testing, that a more human-like conversation is desired. For this purpose, more training examples need to be provided. As the use of emoticons, such as the like-button on Facebook is very common, especially on messenger services, the processing of emoticons needs further exploration. Requirements such as grouping with friends or integrating information from external applications have also not been implemented yet. Our next steps will be to switch to another platform that will allow multi-channel integration, such as Dialogflow, and to further develop the dialog and connect the chatbot to a VCS platform.

CONCLUSION

To reach a wide range of volunteers and to increase resilience by using a large variety of communication channels, we proposed an architecture that comprises a chatbot to handle the communication flow between VCS and SUVs. We used user stories, that have been developed in earlier work (see Betke 2018), to retrieve requirements for a communication system. We consider chatbots to be a suitable tool to meet these requirements, since just one chatbot is able to operate on many different channels, such as SMS and messenger services. Furthermore, chatbots are considered to be user-friendly and have the potential to simplify the registration process for the volunteer. So far, a simple chatbot has been implemented in our KUBAS Facebook messenger.

However, this chatbot does not meet all the requirements set out in this paper, as it is only able to conduct a basic conversation with the user, but not connected to a VCS. Future work focuses on further development of the demonstrator chatbot including the necessary VCS integration, multi-channel support and redefinition of the dialog modelling based on natural language processing. We plan to evaluate the usability of the chatbot in a large-scale exercise with a large number of SUVs and disaster authorities. Evaluation will include a comparison of SUV coordination with volunteer coordination app, chatbot and social media.

ACKNOWLEDGMENTS

The paper is part of the KUBAS project (www.kubas.uni-halle.de) conducted by Martin-Luther University Halle-Wittenberg, University of Regensburg, City of Halle (Saale), ESRI Germany and sponsored by the German Federal Ministry of Education and Research.

REFERENCES

Androutsopoulou, A., Karacapilidis, N., Loukis, E. and Charalabidis, Y. (2018) Transforming the communication between citizens and government through AI-guided chatbots, *Government Information Quarterly* [online] https://doi.org/10.1016/j.giq.2018.10.001.

Augello, A., Maniscalco, U., Pilato, G. and Vella, F. (2016) Disaster Prevention Virtual Advisors Through Soft Sensor Paradigm, in Pietro, G.D. et al. (Eds.), *Intelligent Interactive Multimedia Systems and Services 2016*, Springer, New York, 619–627.

Barsky, L. E., Trainor, J. E., Torres, M. R. and Aguirre, B. (2007) Managing volunteers: FEMA's Urban Search and Rescue programme and interactions with unaffiliated responders in disaster response, *Disasters*, 31, 4, 495-507.

Betke, H. (2018) A Volunteer Coordination System Approach for Crisis Committees, *Proceedings of the 15th International Conference on Information Systems for Crisis Response and Management*, Rochester, NY.

BotList (2018) Meet Richter - Earthquake preparation made simpler. [online] https://botlist.co/bots/meet-richter.

Ciechanowski, L., Przegalinska, A., Magnuski, M. and Gloor, P. (2019) In the shades of the uncanny valley: An experimental study of human–chatbot interaction, *Future Generation Computer Systems*, 92, 539–548.

Cui, L., Huang, S., Wei, F., Tan, C., Duan, C. and Zhou, M. (2017) SuperAgent: A Customer Service Chatbot for E-commerce Websites, *Proceedings of Association for Computational Linguistics 2017*, Vancouver, CA.

Dole, A., Sansare, H., Harekar, R. and Athalye, S. (2015) Intelligent Chat Bot for Banking System, *International Journal of Application or Innovation in Engineering & Management*, 4, 5, 49–51.

Eling, M. and Lehmann, M. (2018) The Impact of Digitalization on the Insurance Value Chain and the Insurability of Risks, *The Geneva Papers on Risk and Insurance - Issues and Practice*, 43, 3, 359-396.

Ferrara, E., Varol, O., Davis, C., Menczer, F. and Flammini, A. (2016) The rise of social bots, *Communications of the ACM*, 59, 7, 96-104.

Ivanov, S.H. and Webster, C. (2017) Adoption of Robots, Artificial Intelligence and Service Automation by Travel, Tourism and Hospitality Companies – A Cost-Benefit Analysis, in Marinov, V., Vodenska, M., Assenova, M. & Dogramadjieva E. (Eds), *Traditions and Innovations in Contemporary Tourism,* Cambridge Scholars Publishing, Cambridge, 190-203.

Japan Today (2018) AI chatbot offers new natural disaster support feature. https://japantoday.com/category/features/new-products/ai-chatbot-offers-new-natural-disaster-suort-feature.

Johannsen, F., Leist, S., Konadl, D. and Basche, M. (2018) Comparison of commercial chatbot solutions for supporting customer interaction, *26th European Conference on Information Systems*, Portsmouth,UK.

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.

Koetter, F., Blohm, M., Kochanowski, M., Goetzer, J., Graziotin, D. and Wagner, S. (2018) Motivations, Classification and Model Trial of Conversational Agents for Insurance Companies, *11th International Conference on Agents and Artificial Intelligence 2019*, Prague, CR.

McLennan, B., Whittaker, J. and Handmer, J. (2016) The changing landscape of disaster volunteering: opportunities, responses and gaps in Australia, *Natural Hazards*, 84, 3, 2031-2048.

Muir, B.M. (1987) Trust between humans and machines, and the design of decision aids, *International Journal of Man-Machine Studies*, 27, 5, 527-539.

Munich RE (2018) Number of relevant natural loss events worldwide 1980-2017. [online] https://natcatservice.munichre.com/events/1?filter=eyJ5ZWFyRnJvbSI6MTk4MCwieWVhclRvIjoyMDE3fQ%3D%3D&type=1.

Nica, I., Tazl, O.A. and Wotawa, F. (2018) Chatbot-based Tourist Recommendations Using Model-based Reasoning, *20th International Workshop on Configuration*, Graz, A.

Peffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V. and Bragge, J. (2006) The design science research process: a model for producing and presenting information systems research*, Proceedings of the first international DESRIST 2006*, 83-106.

Sauer, L. M., Catlett, C., Tosatto, R. and Kirsch, T. D. (2014) The utility of and risks associated with the use of spontaneous volunteers in disaster response: a survey, *Disaster medicine and public health preparedness*, 8, 1, 65-69.

Sermet, Y. and Demir, I. (2018) An intelligent system on knowledge generation and communication about flooding, *Environmental Modelling & Software*, 108, 51–60.

Shawar, B.A. and Atwell, E. (2007) Chatbots: are they really useful?, *LDV Forum: Journal for Language Technology and Computational Linguistics*, 22, 29-49.

Thieken, A.H., Kienzler, S., Kreibich, H., Kuhlicke, C., Kunz, M., Mühr, B., Müller, M., Otto, A., Petrow, T., Pisi, S. and Schröter, K. (2016) Review of the flood risk management system in Germany after the major flood in 2013, *Ecology and society*, 21, 2.

Thomas, N.T. (2016) An e-business chatbot using AIML and LSA, *International Conference on Advances in Computing, Communications and Informatics*, Piscataway, NJ.

Wagner, C., Mitter, S., Körner, C. and Strohmaier, M. (2012) When social bots attack: Modeling susceptibility of users in online social networks, *Proceedings of the WWW'12 Workshop on 'Making Sense of Microposts'*, Lyon, FR.

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, pp.358-368.

Wuenderlich, N.V. and Paluch, S. (2017) A Nice and Friendly Chat with a Bot: User Perceptions of AI-Based Service Agents, *International Conference on Information Systems 2017*, Seoul, ROK.

Zamora, J. (2017) I'm Sorry, Dave, I'm Afraid I Can't Do That, *The 5th International Conference on Human-Agent Interaction*, Bielefeld, DE.