EMuRgency – A Basic Concept for an AI Driven Volunteer Notification System for Integrating Laypersons into Emergency Medical Services

Jesko Elsner, Philipp Meisen, Sebastian Thelen, Daniel Schilberg, and Sabina Jeschke
Institute of Information Management in Mechanical Engineering
IMA/ZLW & IfU - RWTH Aachen University
Dennewartstr. 27, 52068 Aachen, Germany
{Jesko.Elsner, Philipp.Meisen, Sebastian.Thelen, Daniel.Schilberg, Sabina.Jeschke}
@ima-zlw-ifu.rwth-aachen.de

***Abstract*—In case of Sudden Cardiac Arrest, an untreated time interval of only a few minutes usually means the victims' death. Due to a variety of parameters, e.g., the current traffic situation and the mere traveling distance, emergency medical services often arrive too late in order to procure efficient cardiopulmonary. Given this premise, it is necessary to find alternative ways for providing immediate first aid measures. One promising approach is the implementation of a Volunteer Notification System (VNS) – integrating laypersons and medically trained volunteers into the professional medical services by notifying those potential helpers who are, at the time of incident, close to the victim. By tracking the users' location, the system is able to notify those volunteers who can arrive on scene fast enough to provide the urgently needed measures. In September 2011, the European research project "EMuRgency" started the development of such a system. Whereas a running prototype has already successfully been developed, a more sophisticated solution is required to determine the relevant volunteers. While the actual distance is an important parameter, it does not necessarily determine the time of arrival at the scene. Possible obstacles might be in the way and both physical performance and the type of movement have a direct influence on the traveling speed. Furthermore, secondary criteria apply; e.g., the current situation a volunteer is in, knowledge of the area, general engagement and his or her medical expertise. This publication will give you an overview of the "EMuRgency" project and discuss the current stage of development of the VNS; furthermore, this paper will introduce the main concept for determining the relevant volunteers within an ongoing emergency scenario and the advantages of an artificial intelligence approach to enable an efficient volunteer selection will be discussed.**

Keywords*—Volunteer Notification System, First Responder, Emergency Medical Services, Sudden Cardiac Arrest, Cardiopulmonary Resuscitation, Telemedicine, eHealth, mHealth

I. INTRODUCTION

Due to the way today's professional emergency medical services (EMS) are organized, victims in need of urgent medical care are facing a lethal problem. Depending on the type of emergency, the time interval between the incoming emergency call and the arrival of the professional helpers at the scene is simply too long [1]. In Bavaria (Germany) for example, a region with good infrastructure and an advanced medical system, reoccurring studies are made every four years, analyzing the effective time interval local EMS need until arriving at the place of incident. The Institute For Emergency Medicine in Munich (INM) states in a recent study that professional EMS in the area of Bavaria require approximately nine minutes until arriving on scene [2]. The severity of this time deficit generally correlates with the infrastructure a country can provide, resulting in intensification for less advanced countries and less populated regions. While most emergencies do not involve an immediate life danger for the victim, in case of a Sudden Cardiac Arrest (SCA) the first minutes are of utter importance. Jan Bahr states that as little as three minutes is most likely enough for victims of SCA to suffer permanent brain damage and Karin Grassl states that survival without serious permanent damages are practically zero if a victim was left untreated for more than five minutes [3] [4]. Victims suffering SCA are in need of urgent medical care that professional EMS alone cannot always sufficiently provide. The implementation of a Volunteer Notification System (VNS) is a possible solution [1]; integrating laypersons and medically trained volunteers into the EMS by notifying those potential helpers who are, at the time of incident, close to the victim in order to provide early first aid measures and, therefore, gap the time until the professional helpers arrive on scene.

A. *Structure*

The first section of this paper describes the medical emergency of a Sudden Cardiac Arrest (SCA), introduces the basic concept of a Volunteer Notification System (VNS) and gives a brief introduction into the main problems regarding the determination of the relevant volunteers in an ongoing medical emergency [5]. Starting by identifying and discussing comparable systems, the second section analyzes the technological state-of-the-art of mobile phones and messaging technologies; herby determining the possibilities and restrictions for a VNS approach. Furthermore, the basic concepts of artificial intelligence (AI) that are required to enable an efficient selection of the relevant volunteers will be introduced. The third section introduces the research project "EMuRgency". One focus of the project is the actual implementation of a VNS, the current state of development is being discussed; describing the main system components and corresponding architectural details, followed by an introduction to the concept of "Prescient Profiling". The last section provides an outlook on the upcoming research within the project and shortly discusses the relevance on related topics of general location-based services.

B. Sudden Cardiac Arrest

The human heart has an electrical conduction system that controls the rate and rhythm of the heartbeat. Problems with this electrical system can cause irregular heartbeats, so called arrhythmias, which can lead to Sudden Cardiac Arrest (SCA) – a condition in which the heart suddenly and unexpectedly stops beating [6]. The hereby resulting loss of blood flow prevents the brain and any other vital organ from getting oxygen. Without immediate treatment, the victim dies within minutes [3]. It is a common misconception that SCA is the same as a heart attack, while in reality, they are quite different. SCA is an "electrical problem" that prevents the heart in its whole from functioning, whereas a heart attack occurs when part of the heart's blood supply is reduced or blocked causing the heart muscle to become injured or die [6].

C. The basic concept of a Volunteer Notification System

One possible solution for offering faster response treatment is the concept of involving volunteers into EMS by implementing a VNS, which may be defined as an IT system with the following core functionality: by tracking the location of all registered users, the system will be able to notify exactly those potential helpers who are, at the time of the incoming emergency call, close enough to the place of incident in order to provide Cardiopulmonary Resuscitation (CPR) to the victim and, therefore, gap the time until the professional EMS arrive on scene [1]. Whereas the term "close" is suitable for describing the general concept of a VNS, the actual determination of which volunteers are to be notified in an ongoing emergency situation cannot be answered without understanding further implications [5]; this topic will be addressed in part D and E of this section and a possible approach regarding an efficient volunteer selection will be discussed in Section III by introducing the concept of "Prescient Profiling".

The general concept of a VNS does not interfere with the local corresponding emergency standard procedures, but can rather be described as an optional add-on to existing EMS; the responsible dispatcher decides if to involve this optional feature. The potential volunteers are not a replacement for emergency physicians or any professional helper that is normally involved in a medical emergency workflow; their main purpose is to arrive at the victim fast enough to provide CPR and, therefore, bridge the untreated time interval until the professional helpers arrive on scene [1]. While no exact definition or specification of a VNS exists so far, it is part of this publication to discuss its architecture and introduce its main components. One possibly technical implementation of a transnational VNS, utilizing standard smartphones as client devices for receiving notifications, is the focus of the European research project "EMuRgency", which current state of development will be discussed in Section III.

D. Who are the relevant volunteers during an ongoing medical emergency ?

The decision, which volunteers are to be notified, requires a variety of information in order to notify the most promising helpers. False or unnecessary notifications – thus, notifications that will appear immediately irrelevant to the recipient – will have a negative effect on the user acceptance and the system performance, whereas not notifying potential helpers who have high chances to arrive on scene fast enough, will greatly decrease the system's value [5]. Therefore, a selection algorithm is needed in order to produce the best possible set of volunteers at any given moment.

A simple solution for selecting volunteers is the implementation of a notification radius; setting a maximum distance around the place of incident and notifying exactly those volunteers who are within this maximum radius. This approach will provide a set of helpers who are geographically close to the victim, but will they also arrive faster than those potential helpers outside the radius? To answer this question more information is needed, on both the infrastructural situation and on individual user details.

While the actual distance is an important parameter to be considered, it does not necessarily determine the time of arrival at the scene. Due to possible obstacles, the beeline calculation obviously does not offer a suitable background for estimating the arrival time; but even considering roadmap material to calculate the shortest way, does not provide sufficient information without further assumptions. Thus, the type of movement, the physical performance of a volunteer and the current traffic situation, directly influence the approximate traveling time. Furthermore, limiting the relevant decision parameters to merely distance or traveling time appears inadequate and secondary criteria apply; e.g., the potential volunteers' medical expertise, his or her individual knowledge of the area and the current situation this volunteer is involved in [5].

Fig.1 illustrates the general problem; a car driver on a highway (since no highway exit is in reach) and a pedestrian on the opposite side of a river (the next bridge is too far away) are both within the notification radius but will most likely have problems getting to the victim in time. On the other hand, assuming that the train goes in the right direction, a volunteer traveling in the train will, due to the incident being very close to the next train-station, arrive at the victim within a short amount of time even though he or she is far out of the notification radius. If the bicycle rider in the scenario will get to the victim in time (even though again out of the notification radius) depends on the physical performance of the rider and if the road is actually going downhill or uphill.

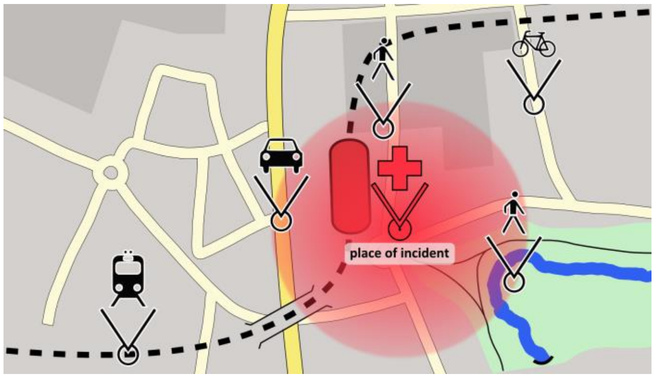

Figure 1: The general problem of an intelligent volunteer selection

At this point, it becomes apparent that a scientifically based answer to the question, "which volunteer should be notified for an individual case?" requires detailed information on the available volunteers. This approach is characterized by an extensive user profiling, therein gathering and processing data from different sources to enable an efficient decision within a limited information environment. The goal is to create a prescient system, which uses historical data and various concepts of artificial intelligence (AI) in order to calculate reliable predictions on any given aspect of the based user profile. Details on this approach and the underlying concepts will be discussed in the upcoming sections of this paper.

E. How can the selected volunteers be notified ?

The general approach within the VNS is to notify potential helpers by using available smartphone technology; a variety of messaging technologies and frameworks are existent at the moment, but no matter the usage they all have one thing in common: any location that is requested on the mobile client and any data that is sent from the client to the server consumes battery power. In recent tests, done within the project scope and with up-to-date smartphones, continuous real time connectivity from different mobile clients to the VNS server, sending any available location update and data changes instantly to the server without taking any precautions regarding battery life, drained the battery to zero within approximately five hours. Therefore, in order to provide an acceptable solution that is suitable for every day scenarios without having constant access to energy sources, a compromise between the server staying up-to-date and the mobile clients' battery load has to be found.

Another parameter to take into account is the probability of weak internet coverage in specific zones. These can result in non-frequent location updates and connectivity issues regarding the connection reliability between the mobile client and the VNS server. Thus, notifications sent might never arrive or the systems' selection algorithm might consider volunteers whose actual location differs significantly from the last successful location update sent to the server. While the actual connectivity and the quality of the technical implementation have a direct influence on the location data that is available on a user, this data is also known to have fluctuations by default, which require additional measures in order to restrain further implications on the selection algorithm. Technical details regarding "how" to notify the relevant volunteers are being discussed in Section II as part of the existing mobile and messaging technologies.

II. State Of The Art

A. Existing notification systems

While some local approaches to implement notification systems (e.g., [7] [8]) exist already, those approaches generally do not have an academic motivation or background. Therefore, publications on the aspect are still rare and the corresponding projects are neither opening their expertise nor the source codes to the public. The only publically available resources are the corresponding application download, some basic usage documentation and a reference document for the Advanced Programming Interface (API) – which merely offers functionality for providing the systems with data [7].

Based on reviews and the appearance in media all over the USA, the PulsePoint Foundation for example offers one of the most advanced software implementations in the field of emergency notifications at the moment [9]. Formerly known as the "Fire Department App" and developed for iOS only, the new version is available under the name "PulsePoint" for Android and iOS [7] [8]. Even though this application is surely great for offering everyday people a possibility to save lives, based on the available documentation, it is a US-only solution without open interfaces. From an academic point of view, it is regrettable that the achieved competences are not shared and the source codes are not publically accessible, which makes it is nearly impossible to use the project as a base for a scientific work. Furthermore, the implementation approach is rather static, allowing two types of mobile devices (Android and iOS) as recipients and no other but US specific regulations, legal circumstances and network characteristics are supported. There are a few smaller projects with less impact and publicity, but the problems stay the same and none of the available solutions actually considers an "intelligent" user selection or a bidirectional communication stream.

Beside the difficulties stated above, the available solutions are implemented as local solutions that cannot easily be adapted to other countries, regions or new legal environments. Fundamental changes are needed in order to use these systems with other than the original parameters and the underlying model itself does not provide a reasonable extension of functionality without making changes to the actual source code itself. In summary, the currently available systems lack essential interfaces, public tools for gathering and extracting information, an efficient communication flow and basic concepts for extensibility. Thus, the demand for an intelligent system, that can efficiently forecast the volunteers' location and performance, requires an entirely new approach, which is introduced in Section III as part of the "EMuRgency" project.

B. Mobile technologies

Advances in mobile technologies and the continuous growing popularity for portable digital devices with internet access in nowadays society offer a great starting point for VNS. Without supplying any special devices, a VNS is able to communicate with a huge variety of volunteers by simple using the existing hardware and infrastructure that people own and use every day. Modern smartphones for example offer a diversity of features that may be used to aid potential helpers in their mission to arrive on scene as early as possible. Some notable built-in features are real time internet connections, notification options with vibration and sound, photo and video modes, a variety of sensors to enable situation based functionality like a compass, and the fact that actually any modern mobile device is running an operation system (OS) that supports programmatic solutions for individual software [10].

Based on the basic definition of a VNS, the core functionality of any VNS is the effective localization of the volunteers. The actual localization of mobile devices within a network is a complex matter, while the reliability of the results generally depends on the corresponding network provider and its infrastructure [10]. Different companies and research groups are working on this topic, offering a variety of Advanced Programming Interfaces (API's) with base functionality to access localization data for different types of devices. One of the most advanced examples is Android's Location API, which is part of the Android software platform, developed by Google in conjunction with the Open Handset Alliance (OHA) [10].

The OHA is a consortium of various companies, working on developing and advancing open standards for mobile devices. The consortium, led by the Google Incorporation, includes some of the biggest mobile operators like Telekom and Vodafone, as well as some important manufacturers of mobile devices like Samsung and HTC. As an open-source project, the Android source code is publicly available and can be accessed freely [10]; this reflects in a high user acceptance and fast development progress due to contributions from the open source community. The comScore Incorporation frequently publishes reports on the mobile subscriber market worldwide and recently published the numbers for the first quarter of 2013, showing that Android is holding an average of more than 70% market share within the biggest countries of Europe [11]. In a press release from August 2012, the International Data Corporation (IDC) states that the Android market share is continuously increasing [12]. Both studies are based on device sales in the corresponding regions and therefore reflect the general tendency within the segment of smartphones and other mobile devices with internet access.

Even though restricting the notification recipients to exclusively smartphones and similar devices running Android is questionable, it seems to be a reasonable decision for rapid prototype development in order to provide an early running system as soon as possible. It must clearly be stated that a limitation of this kind can only be temporary and that a final model of a state-of-the-art VNS has to provide a generic communication approach in order to support a broad variety of different devices. A more detailed discussion on the topic of a possible generic approach will follow in the upcoming sections of this paper.

C. Messaging Technologies

The Hypertext Markup Language (HTML) defines the core language of the World Wide Web (WWW). With the HTML 5 specification becoming the new standard for web interactivity, a lot of features are accessible for programmers to enable client and server technologies to communicate with each other. While a detailed discussion on server push technologies and HTTP requests would clearly exceed the context of this paper, it is important to note that the HTML 5 specification includes full support for so-called WebSockets. WebSockets specify an API as well as a protocol, while the protocol defines the HTTP handshake behavior to switch from an existing HTTP connection to a lower level connection; a so-called WebSocket connection. While a common approach over the last years was to simulate a server push channel over HTTP, a WebSocket connection enables bidirectional communication natively [13].

The Communication between VNS server and volunteer's devices (the clients) is a central part of the VNS' system architecture. To support the VNS core functionality it has to transport different types of messages:

- Frequent location updates from the mobile clients to the VNS server.
- Information streaming from the server to the clients, containing information on upcoming events, general content or medical tutorials.
- Case notifications and bidirectional real-time updates between the server and the clients, which have been selected as potential volunteer in an ongoing case.

The architecture and technological choices for this communication have major implications for the system's openness regarding device platforms, the timeliness of message delivery and both client and server performance. Client performance and thus the user experience, is most notably influenced by their device's battery life time while the VNS client application is active on the device. As mentioned in Section I, not draining a device's battery is an important non-functional requirement to the system. Server performance can be characterized by the number of clients that can be handled concurrently by a defined set of physical hardware.

Whereas a manifold of approaches exist to realize this communication, a first architectural choice is to use a message queuing based communication pattern, which conforms to the current state of the art and reduces the solution space considerably. This communication pattern defines that peers communicate by means of asynchronous message exchange. Messages are sent by publishing them to one or more queues. The published messages are then received by subscribers of these queues. This communication pattern – often referred to as publish/subscribe or short pub/sub – is well established in enterprise integration and many protocols, frameworks and implementations exist, which realize this pattern. This pattern usually involves a central message broker or a cluster there of which holds the queues and routes the messages between all connecting peers [14].

Using this communication pattern with mobile devices imposes requirements onto realizations of this pattern, which are not present when communicating merely between fixed peers; like the frequently changing quality of service and bare availability of network connections between a mobile device as peer (MDP) and the broker. Also the hibernating of the MDP – in order to reduce power consumption –results in new requirements, since an MDP is (ideally) continuously connected to the broker but will not be able to receive or send messages while hibernating.

Based on the discussed requirements and corresponding with the introduced pattern, the following three frameworks have been chosen for a detailed assessment:

- Simple Text Oriented Messaging Protocol (STOMP) in combination with WebSockets as initial take, in order to use a web technology as basic transport and STOMP on top of it for the pub/sub semantics [15].
- Google Cloud Messaging for Android (GCM) is a native Android messaging framework. It delivers great reliability but implements a merely one-way communication (server to client push) [16].
- MQ Telemetry Transport (MQTT) is, according to its developers "…a publish/subscribe, extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. The design principles are to minimize network bandwidth and device resource requirements whilst also attempting to ensure reliability and some degree of assurance of delivery." [17].

The assessment's results were gathered by performing a review of literature, developer resources and the web as well as by implementing prototypes and performing preliminary tests within the project's system implementation; they are summarized in Table I. The frameworks were judged by the following four parameters:

- Connection Management refers to a framework's ability to handle the special network conditions of mobile devices. I.e., does the framework have versatile capabilities in this regard or is it necessary to cope with the challenges in the client code?
- Platform Interoperability refers to a framework's ability to be used on different mobile phone platforms such as Android and iOS.

Table I: Assessment of messaging frameworks for the VNS

	Connection Management	Platform Interoperability	Resource Efficiency	Messaging Semantics
STOMP	-	+	-	o
GCM	o	-	+	o
MQTT	+	+	+	+

- Resource Efficiency refers to a framework's utilization of network bandwidth and central processing unit as well as its ability to let a device enter lower sleep states and thus hibernate in order to save battery life time. This parameter is a very rough estimate based on a framework's architecture and not based on actual measurements. It thus has to be taken with great care.
- Messaging Semantics refers to the versatility of a framework's messaging protocol regarding message handling like Quality of Service (QoS) aspects, publication/subscription schemes and message life time.

Whereas the VNS implementation within the project was originally based on WebSockets in the early prototype versions, a variety of connectivity issues were encountered during the prototype evaluation. With HTML5 and therefore WebSockets still being a new and a not yet fully established technology, any detailed discussion on possible explanations would clearly exceed the intention of this paper. As a result to the unsatisfying performance of the WebSocket implementation, recent builds of the EMuRgency VNS are entirely based on the MQTT connectivity protocol, which will be discussed in the upcoming section.

D. *The MQ Telemetry Transport*

MQTT is designed as a protocol that transfers the enterprise integration pattern of publish/subscribe message queuing to telemetry nodes connecting to a message broker over constrained networks [17]. The current version 3.1 of the protocol specification, as provided by IBM and Eurotech, has been submitted to OASIS for general standardization [18] [19]. In detail description of the protocol can be acquired from various online resources, such as [17] [20]. As our own reviews show, scientific literature covering the protocol is sparse at the moment. The following gives a brief summary of its main distinguishing features.

The publish/subscribe scheme provided by MQTT exclusively uses topics, which can be arranged in hierarchies. In order to subscribe to multiple topics in a given hierarchy, wildcards can be used for different parts of the hierarchy when subscribing. For strict unicast messages, a topic per peer must be used. As there is no explicit notion of queues in the MQTT protocol, all peers subscribing to the same topic get their own copy of a message, thus enabling easy broadcast realization by shared topics.

Three important features of MQTT are Durable Connections, Retained Messages and Wills. If a client defines its connection as durable, the corresponding broker will use a storage location in order to persistent the relevant subscriptions and store non delivered messages; e.g., while the client is disconnected. On reconnect, these messages are then published to the client. The most recent message per topic, which was marked as retained, is stored by the broker and delivered to every newly connecting client. Upon connection, a client can define a so called "will message" on the broker. In case the broker detects an abnormal disconnection of the client (i.e., a time out), it publishes the client's will message, which can therefore inform potential subscribers of the up-to-date system status and invoke local adjustments.

E. Positioning, Prediction and Profiling

Section I.D states, that a major problem is the answer to the question "which volunteers should be notified in case of an incident?" and that the rather simple solution "Just notify everyone within a specific radius" will often lead to counterproductive notifications and is not always technologically achievable. The technological constrains introduced in Section I.E result in an increased lack of information, which therefore must be forecasted based on the historical data that is available within the system. The next three sections describe the state-of-the-art concerning the possibility to use infrastructural knowledge for positioning, generate a volunteer profile and predict the actual position and behavior of an individual volunteer.

F. Positioning using map-matching algorithms

Map-matching is a process, which was initially defined to map the inaccurate position of a person or vehicle retrieved via GPS to a valid position of a provided road network [21]. The map-matching problem is illustrated in Fig. 2.

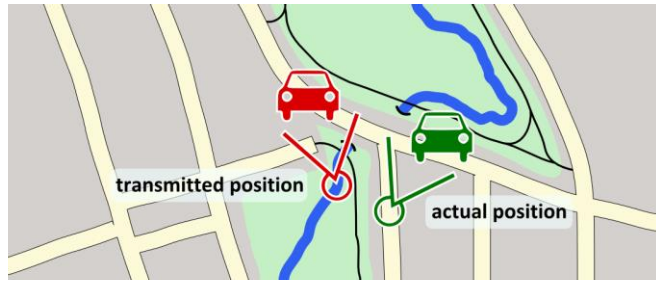

Figure 2: The map-matching problem

Different map-matching algorithms were introduced over the last years, which are generally grouped into geometric, topological and probabilistic algorithms [22]. Depending on the input-data requirement, different algorithms to determine the actual position are available. Geometric map-matching algorithms expect a transmitted position and a map, whereby topological algorithms also expect the availability of previous positions (i.e., a trajectory). Probabilistic algorithms use mostly the same input as the topological algorithms, but additionally predict several actual positions with a probability [23]. There are a few enhanced algorithms, which use further data (i.e., temporal information) and additional techniques like dead reckoning or trajectory prediction, in order to increase the quality of the map-matching [24] [25] [26]. Those algorithms are often adapted to a specific use-case and therefore difficult to apply to other cases.

Furthermore, the map-matching algorithms can be divided into so called on-line and off-line algorithms [22] [27]. On-line algorithms are mainly used to determine the actual position in real-time, whereby the latter are used to determine the trajectory (or just points of it) in a post-processing step having the complete set of previously obtained geo-referenced points available.

1) Positioning based on dead reckoning techniques

Dead reckoning techniques are used to approximate the position of an observed unit (e.g., a pedestrian, vehicle or ship) based on some stationary information (e.g., the last valid position) and collected, non-positioning information (e.g., traveling direction, destination, or average speed). Dead reckoning is widely used and applied in different research areas; e.g., autonomous driving [28], robotics, pedestrian or indoor positioning [25] [29] [30]. The techniques can be divided into the variants, which are illustrated in Fig. 3.

The techniques presented in this publication often vary in the sensors used (e.g., compass, accelerometer, motion measurement unit or odometer) and the way they utilize the available data. In general, dead reckoning techniques are good alternatives or complements when positioning data (e.g., GPS, WIFI- or cellular-positioning) is unavailable or insufficient (e.g., indoor, underwater, bad reception) but additional information like acceleration, speed, or direction are available. Furthermore, dead reckoning techniques are one approach to project a possible future position [31].

2) Trajectory prediction

A projection into the future is realized by a so called "trajectory prediction". This prediction is based on an initial location and other none-positioning sensor values. Another approach to forecast the position of a unit is based on data-mining techniques. Those techniques normally work in two phases.

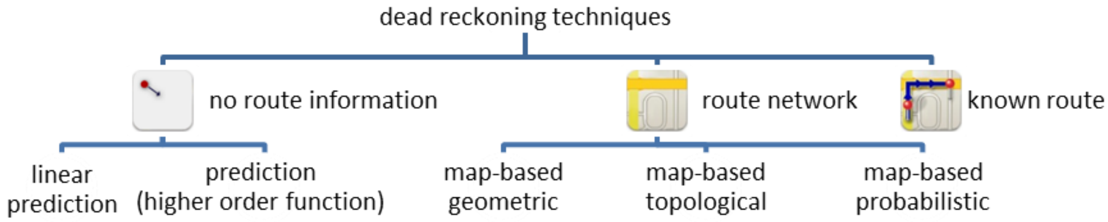

Figure 3: Variants of dead reckoning techniques

The first "learn-phase" is used to detect the trajectory patterns within the collected data. In the second "predict-phase" the current trajectory is matched against the discovered patterns. The matching patterns are finally used to project the current trajectory into the future [32]. Some techniques also combine the two phases and use the currently observed data to continuously enhance the patterns on-line [33]. The two phases of data-mining for trajectory prediction are illustrated in Fig. 4.

Generally, there are two data-mining techniques used for the "learn-phase", i.e., the pattern detection. These techniques are clustering [34] [35] and sequence mining [36] [37]. The difference between them is how they interpret (i.e., model) a trajectory within the system. Therefore, the results and usability vary concerning the kind of patterns that are looked for (e.g., region-, trajectory-, behavior-patterns). The "predict-phase" is depending on the technique applied and the usage of the discovered patterns, based on Kalman Filters, Hidden Markov Models or directly on the patterns [32] [33].

3) Profiling

The context of the historical trajectories used to determine the patterns (as described in the previous section) often depends on the use-case. For example: It might be necessary to find patterns within the collected data of a specific day and from a specific person, or it might be sensible to look at all the trajectories of a specific group of people (e.g., clustered "a priori" by age or job). Profiling does address this type of problem, i.e., the grouping or categorizing of data, so that a generalized profile is determined. A profile can thereby be generated for one entity (e.g., a person, a city) or a group of entities (e.g., all persons of a specific age, all cities with a specific population).

Profiling does use and combine different techniques from

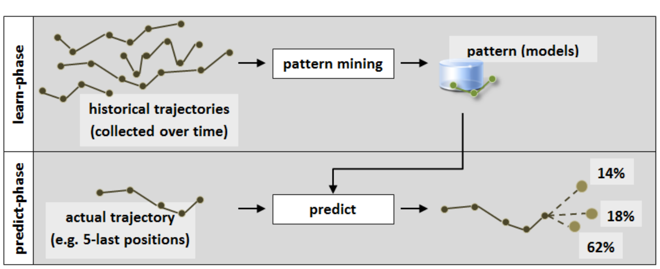

Figure 4: Two phases of data-mining techniques used for trajectory prediction

different research areas, e.g., often data mining and machine learning techniques. A state-of-the-art overview concerning profiling is very extensive and extends the intention of this paper. The goal of this publication is to highlight the necessity of an intelligent user selection and give a short introduction on the topic of volunteer-profiling in order to predict his/her current situation concerning the discussed restrictions. Especially a reliable prediction concerning the current position and situation is important, so that infrastructural constraints and situational limits can be foreseen. The introduced algorithms and techniques of the previous section cover the state-of-the-art concerning possible infrastructural constraints [38]. Some of them are also likely to be adapted in order to detect situational limitations. Thus, the type of used vehicle (e.g., in a train, afoot), possible company (e.g., with children) or individual circumstances (e.g., under time-pressure) can be implemented case by case or the system has to adapt itself over time by implementing learning adapted algorithms.

III. The EMuRgency Project

The European research project "EMuRgency" started in September 2011. Research facilities from Germany, the Netherlands and Belgium are working together on modeling and implementing an integrated Volunteer Notification System (VNS) to gap the time between an incoming emergency call and the arrival of professional helpers at the scene. The name of the project is a composition of the two words "emergency" and "urgent" and refers to urgent help that is needed in case of SCA. The three upper case letters "EMR" identify the regional base of the project; the "Euregio Maas-Rhein" (Eng. "Meuse-Rhine Euroregion").

A. Definition of the term "volunteers" within a VNS

Before describing the system, its components and the technical details, it needs to be clarified, which group of people can actually participate as volunteers within a VNS. A volunteer can be anyone with basic skills in first aid and CPR (Cardiopulmonary Resuscitation) who is willing to help in case of an emergency. It is important to differentiate this definition from the term "first responder", which was defined by US National Highway Transportation Safety Administration as "the first medically trained responder who arrives on scene of an emergency" [5]. While the definition of a first responder includes groups like police officers, firefighters and EMS, it does not include laypersons since those generally do not have medical training. Still, laypersons might be able to provide the needed measures in order to help victims of SCA and thus should be included as potential helpers within a VNS [39]. Within the EMuRgency project, the term "volunteer" is referring to any potential helper, medically trained or not, willing to aid other people in an ongoing emergency [1].

B. Integration of VNS and professional EMS

Whenever an incident is reported to an emergency dispatch center that might involve SCA, the dispatchers will do what they normally do: send professional help – but optionally also invoke the VNS. It is important to stress that the VNS, at this time of development and based on the way EMS are organized today, is a merely optional feature. This means that the responsible dispatcher may or may not involve the VNS, depending on their analysis of the case and personal motivation. In order for the optional integration to be achieved, the VNS has to provide a user-interface where the dispatcher can initiate a case by forwarding its exact location and some optional information to the system. During SCA, time is of utter importance, so this user-interface has to be as simple and efficient as possible.

Within the project, interviews were made to determine the acceptance and motivation of the dispatchers to integrate a VNS within the general workflow; even though all the interviewed dispatchers agreed on a potential benefit, it became rather clear that the general acceptance of new systems seems to directly correlate with the extra work that is involved in order to use it. Taking into account the discussed optionality and the still early stage of development within the project, a manual integration will be the starting point towards involving the VNS within the professional EMS workflow. The implementation of an integrated system that gets activated fully automated during a reported emergency is surely desirable, but requires detailed collaboration with the corresponding software providers. At the current stage of development, different regional dispatch centers expressed their goodwill to work together with the project consortium and enable a real-life test scenario within the next evaluation period upcoming in 2014. The details on this cooperation are not yet fully discussed and are also a matter of various legal restrictions regarding the different countries [40].

C. The Decision System

As soon as the system receives information on a new emergency, no matter if automated or manually initialized from the dispatcher, the VNS will determine a set of volunteers, which are to be notified of the incident. In order for this to be possible with minimum time effort, the system needs to be "aware" of all potential volunteer locations at the time of the incoming emergency call. This awareness can be achieved by making all connected clients publish their locations to the server in pre-determined time intervals or whenever a significant change occurred. It is important to understand that this location updates arriving on the server will not be interpreted as the effective actual location, but rather as a possible user location with a significant high probability.

As discussed in Section I, the location is not the only decision parameter for a volunteer to be considered. Secondary criteria apply and the actual decision process is influenced by different criteria and a variety of available data on each potential volunteer; this data is either statically available or dynamically collected within the system and then taken into consideration, using the criteria, in order to create a set of results, the decision. The following three categories represent the different types of data available within a volunteer notification system [5].

The first category of data is requested during the actual registration of a new volunteer; the medical competence level, the contact data and personal details like the birthdate are some examples. This data represents a static layer of information; the content of this information is unlikely to change and potential changes are generally not requested by the system but instead triggered by the volunteer – e.g., supplying new certificates of medical training or changing the address.

The second category of data is characterized by dynamic data and actively collected by the system. The actual location of a volunteer, represented by a combination of latitude, longitude and altitude, or an available acceleration index on a mobile client, are some examples for frequently changing data that is frequently published to the system.

The third category is represented by data that is generated by the system itself. Section II discussed various concepts and techniques of data mining in order to create new datasets. This category characterizes the concept of "Prescient Profiling", creating abstracted behavior patterns and route approximations based on the collected data of an individual volunteer.

As shown in Fig. 5, the three identified types of data are building the input base for the actual decision system, whereas the final output will be a set of volunteers to be considered. With various types of data available, a decision has to evaluate the relative importance of each of them; this evaluation must not be constant but instead occur frequently and adjust over time, assessing the quality of past decisions in correlation with the current learned knowledge base [41]. Different learning algorithms apply and need to be evaluated against each other. Whereas the basic data (static and/or dynamic) will by itself result in a plausible set of volunteers already, an enhanced set of volunteers will be selected when processing the third category of generated data. In order to produce valid assumptions on the efficiency of the selection algorithm, both sets need to be analyzed and compared.

D. Webservices

In order to enable external sources, for example mobile clients or applications from project partners, to access specific functions on the server, accessible interfaces are required. Specific APIs provide any kind of clients the necessary functions in order to communicate with the VNS without having to access a web browser. The same approach provides the different project partners with an interface to integrate alternative ways of volunteer registration into the main system. Commonly used solutions for offering a limited scope of functionality to external clients are so called

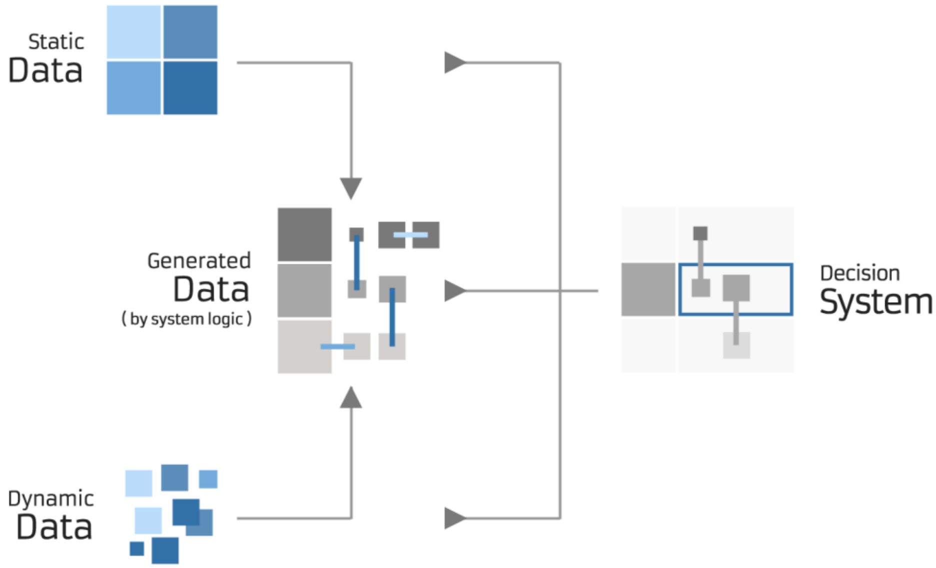

Figure 5: Different types of data utilized by the decision system

WebServices or more specific, REST based Web APIs [42]. By implementing Webservices, a predefined set of functions will become externally available; different devices will be able to communicate with each other over a given network infrastructure; like for example, the World Wide Web (WWW).

E. The messaging architecture

At the current state of development, an advanced prototype of the VNS is available and running in different test scenarios. The upcoming section will introduce the used messaging architecture and describe the essential components implemented for the different types of communication.

As long as a client is not involved in an ongoing case, the main communication occurring between the mobile clients and the server is defined by a client-to-server location publish; that automatically gets invoked in a predefined time interval or whenever the location of the mobile client significantly changed. As current implementation of the location framework, the Google Maps V2 API has been utilized, which requires Google Play Services in order to work [43]. While the first implementation approach within the early prototype versions was based on HTTP requests for posting and pushing this data [1], the current messaging architecture merely utilizes the MQTT protocol to exchange data between the client and the server [18]. HTTP requests in general produce high latencies and require a strict order of POST and GET request. Furthermore, they do not allow a server to client message push, which is required for the actual notification event. Furthermore, bidirectional real time connectivity is required to enable a variety of supportive features; for example chat-channels for notified helpers and the option to enable a graphical feedback for the dispatcher of the current situation on each volunteer.

The WWW was originally not intended to support bidirectional communication and therefore does not include the corresponding specifications or protocols. With HTML 5 introducing the WebSocket JavaScript interface, a native solution for bidirectional communication is available. Once a WebSocket connection has been established between client and server, instant data communication is supposed to be possible between both sides without explicitly having to deal with technical differences. The experiences with this technology, as stated in the sections above, were disappointing, regarding the reliability and quality of the connection. After various test runs, the decision was made to entirely switch to the MQTT messaging protocol, which has been introduced in Section II.

The MQTT implementation within the server-side applications is a more or less standard procedure. A connection to the broker is established and kept alive with reoccurring pings. In case of an unexpected disconnect, a new connection will be established. Since the server is running both the MQTT broker as well as the servlets utilizing the connection to the broker, incoming messages will be handled by the corresponding functionalities without having to care about lifetime-cycles or explicit queuing. Receiving and sending MQTT messages in this architecture are both straight forward implementations regarding the service-side.

The mobile client side of the MQTT implementation is an entirely different matter. Without going too much into detail about mobile client programming, there are a variety of architectural features that have to be taken into account. As mentioned in Section I, the battery consumption is of high importance, since a high CPU load can drain a fully

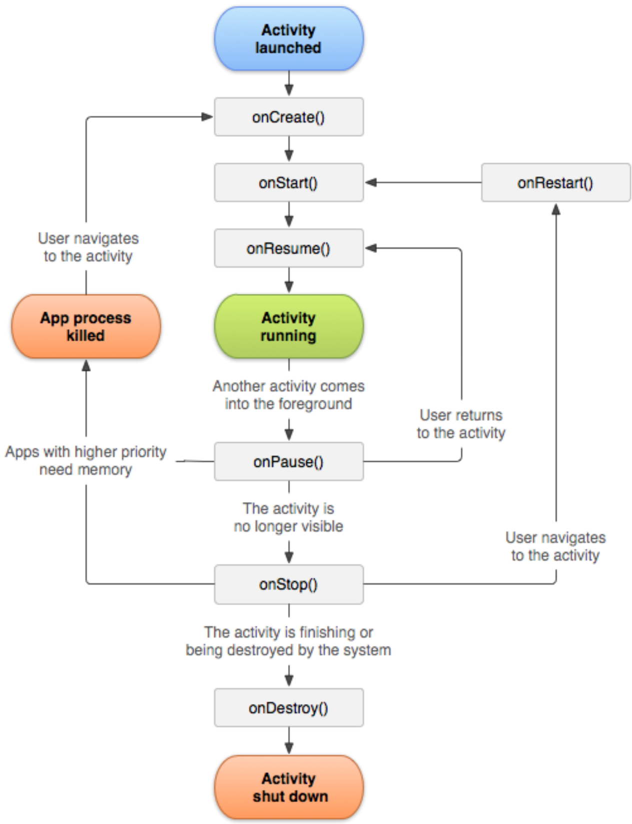

Figure 6: Lifecycle of an Android activity

charged battery within a few hours. To prevent this, smartphones usually have a so called "sleep mode", which is generally divided into a partial (only the screen turn off) and a full (the CPU goes into sleep as well). The problem is that a mobile device, which is sleeping, has very limited functionality, actually only device specific services are still running, everything else is paused; including running applications. This has various implications on the way an MQTT connection is kept alive, as well as on how incoming messages can be received or handled. The android operation system uses so called activities as "views" within an application. Their lifecycle is illustrated in Fig. 6. An application can have multiple activities running simultaneously and influencing each other's state. When the Android device goes into sleep mode, all activities will be paused (even the last displayed one) and no more functions will be called, not even if explicitly invoked by a running service [44]. Furthermore, Android might recycle any running activity during sleep mode. Since a continuous MQTT connection is required in order to implement the needed functionalities (esp. the notification), an Android background service has to handle the connection and wake the CPU (partial wakelock) every now and then to tell the broker that it is still valid or reconnect in case of a timeout. The service has to run in the same thread in order to prevent leaking. This approach delivers a steady MQTT connection on the mobile client, but does not solve the problem of paused activities. In order to handle incoming messages, the MQTT service therefore has to wake the CPU and afterwards broadcast the message to its subscribers (i.e, broadcastReceiver). Due to the lifecycle of created activities, the subscription to the broadcasted messages must not occur for succeeding instances and must be canceled in case of specific event.

Since the delegating service actually runs independently from the application and any activity, a MQTT wrapping framework has been developed within the project, which delivers an elegant way of keeping the MQTT connection alive by also providing base functionality for sending and receiving messages from a configurable MQTT broker. During tests the actual messaging implementation delivered reliable connectivity with low power consumption and enables an integrated approach of a consistent messaging architecture connecting heterogeneous devices.

To sum up the implemented messaging architecture at the current state of development, a single messaging protocol (MQTT) is implemented instead of combining various frameworks and technologies (HTTP, GCM, WebSockets), as done in the early prototype versions. While the actual implementation required the refactoring of nearly all system components, the first results are very promising and will be evaluated in detail in the upcoming weeks.

F. *An integrated VNS platform*

As a scientific project with partners from both technical and sociological research fields, the project is focusing on many more aspects than a simple technical approach for a notification system. Users of the system will be informed on ongoing events or urgent news and since a common interest level of registered people can be implied, communication channels for exchanging know-how and general information are being implemented. A real-time information flow regarding the aspects of first aid and CPR is extending the core notification functionality. While also developing concepts on raising public awareness on SCA, educational content is displayed at frequently used public places; and if in digital form, the streamed content will be enriched or synchronized with data from the VNS. Furthermore, in order to receive substantial scientific results and to determine the potential benefit of a VNS, corresponding reporting and analyzing features are being designed. Open interfaces supply options for non-project members to change or extend functionalities. An integrated VNS platform combines the different research topics with the diversity of requirements that are to fulfill [1].

G. *Regional differences for involving laypersons into professional EMS*

It is important to understand the actual role of an occurring registration and the resulting implications on the system and the user. A newly registered user for example, obviously wants to help, but comparing different countries, potential differences between the way that laypersons are legally allowed to be integrated into EMS, must be considered [40]. Some regions for example might not allow the integration of laypersons in EMS at all; and is a layperson with first aid skills but without corresponding

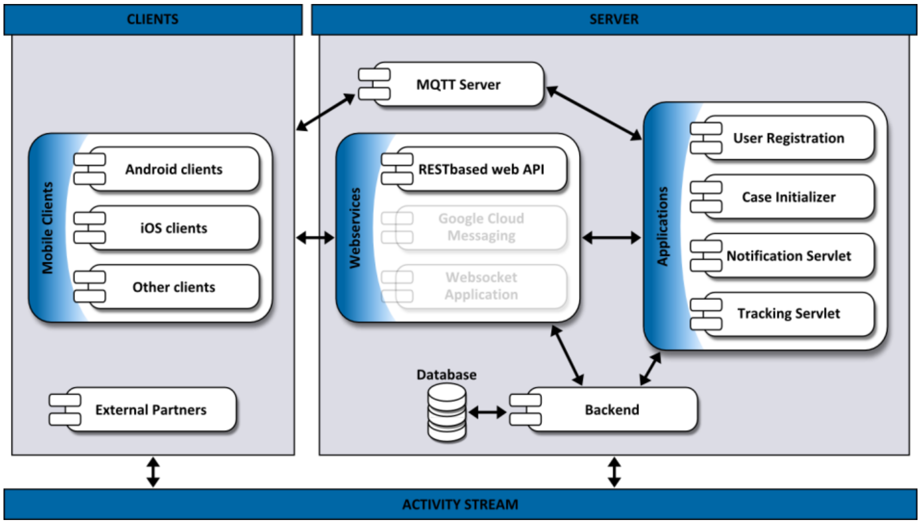

Figure 7: Core components of the EMuRgency VNS

certificates still a layperson? While those questions will not be discussed further within this paper, expert legal advices for different countries are contracted, to validate this matter. The direct implication concerning the VNS is that a new user by default will be "unconfirmed" and will not be considered a potential volunteer until "confirmed"; the confirmation process on the other hand is implemented in a separate administrative component, whereas the final details of this component are again a matter of regional differences that are, at the current stage of development, not yet fully established. With the first real-life evaluation of the system planned for 2014, many of the regional specialties will be addressed in the upcoming months.

H. An overview of the primary components within the VNS

Within the past sections of this paper, the core components of a VNS have been introduced in their corresponding context. Fig. 7 shows an overview of these components while the following paragraph will give some additional information on how they have been implemented.

User Registration: Implemented as web application, this component offers base functionality for new users to register to the system; existing profiles can be edited and specific settings can be configured by the user. Due to the implemented Webservices, alternative ways for sending a user registration are available; users can for example directly register using the mobile client.

Case Initializer: This application constitutes the actual data provider for new emergencies until the first integration into a dispatch center is available. Intended for dispatchers only, this web application provides the possibility to manually initiate a new case by providing general information on an ongoing emergency and the corresponding unique location, as pair of latitude and longitude. In order to supply a user friendly interface, the location itself is automatically calculated as an approximation for a given address. This component demonstrates a basic approach towards supplying the actual notification system with case data and is implemented as various JavaScript applications embedded within a shared Vaadin servlet [45].

Server-side applications: This component bundles the server-side functionalities of tracking, localizing and notifying the volunteers; a steady connection to the MQTT broker is established within a messaging servlet and utilized as a shared resource. The Case-Initializer is part of this component and utilizes the shared Vaadin framework as well as the MQTT functionalities. This approach provides fast interaction times when invoking new emergency events or communicating with the Backend and enables the efficient implementation of various secutiry features.

Backend: The Backend represents the interface for persisting data and provides functions to enable communication between the database and other system components. The database implements the Spring-MongoDB ORM framework [46]. Moreover, to ensure a consistent data usage within the system, all referenced data models and structures are defined within this component.

Webservices: Consisting mainly of the REST based Web API, this component actually represents an intermediate communication layer; providing predefined functions for external modules and other clients to exchange data with the server. Compared to earlier versions of the VNS approach, this component no longer utilizes Websockets or the GCM Messaging framework, but instead merely delegates external requests to the shared resources.

IV. CONCLUSION AND OUTLOOK

During the project, sociological and technical aspects are being combined. Country-specific differences in a variety of discussed parameters have been balanced against each other and are being implemented into an integrated VNS platform. Many fundamental difficulties were identified within the past sections of this paper whereas an advanced prototype of the software is available already - Fig. 8 shows screenshots of the Android client application. This prototype implements the essential components introduced in Section III and enables base notification functionality for nearby helpers. The current system has been evaluated in various test runs and is planned to be integrated in real-life scenarios in the near future. The seamless integration of the different project partners with their interdisciplinary research topics has formed into a complex VNS platform. Current discussions with different regional dispatch centers underline the necessity of such a VNS and will hopefully lead to long-term collaboration possibilities regarding the topic of volunteer integration.

There are a variety of additional features that are being discussed within the project consortium at the moment; they will be shortly introduced in the following section.

A. *Aditional features*

One promising feature is the integration of existing applications or services that provide information on nearby automated external defibrillators (AEDs). Although the time critical aspect of CPR is the main concern for the first volunteer who arrives on scene, it can prove very useful for further helpers, to have reliable information on nearby AED devices. The project is right now hosting a separate web application gathering data on AEDs within Germany, the Netherlands and Belgium.

The integration of chat channel functionality, enabling direct communication between the dispatcher and all volunteers who accepted a specific case, is another reasonable feature. By utilizing the introduced advantages of the current platform approach and subscribing each client on a case-specific MQTT topic, dispatchers and volunteers will be able to exchange information in real-time. By communicating with each other, the potential helpers will be able to share local information or arrange an AED pickup in the near vicinity of the victim. The dispatcher can thereby supply additional information that has not been included in the original notification.

There are multiple scenarios in which the use of game

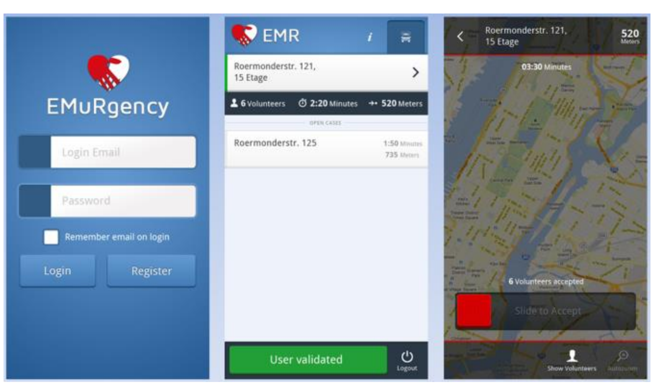

Figure 8: Screenshots of the EMR mobile client (Android version)

design elements (gamification) within a VNS can be used to influence the user behavior; the adoption of a score system for attended courses or participated cases is one example. The general idea of gamification elements is to increase the user acceptance and motivation, whereas a sensible consideration is needed in order not to distract the users' attention from the main topic.

Modern smartphones and many other portable devices with internet access offer build-in functionality for photos and videos. With internet bandwidth increasing continuously, a real-time media streaming from the place of incident is another possible feature with high benefit. The integration of telemedical concepts (e.g., the regional project TemRas) becomes possible and thereby enables an approach of professional helpers using their expertise to analyze the streamed data, supporting the volunteers at the scene with valuable information [47].

B. *Specification of a VNS and future development*

Within this paper, the basic concept of a VNS has been described and both the core functionalities and additional functionalities have been discussed. At the current stage of development, a formal specification of a VNS is not yet possible, since many parameters are not yet fully determined and legal issues are still being discussed. Diverse aspects (legal, social and technical) still need to be finalized and the system has to be evaluated in real-life, rather than within theoretical test scenarios. The upcoming integration into the professional EMS workflow for a limited period of time will prove very useful for producing valuable data on many of these aspects.

Maintaining a highly agile programming approach will assure a continuous development and a fast adaption of new functionalities, whereas the implementation of the core system reached a notable status regarding stability and reliability already. The mobile clients will be enriched with individual information channels, integrated with the server side profiling and ergonomic design will replace the so far functional driven user interfaces.

The research focus for the upcoming months will be a seamless integration of the core system with new components implementing the introduced concept of "Prescient Profiling" in order to enable an intelligent volunteer selection and produce scientific output regarding the question of "who are the relevant volunteers in case of time critical medical emergencies".

ACKNOWLEDGMENT

This paper is based on work done in the INTERREG IVa project EMuRgency (www.emurgency.eu). The project is partially financed through the European Regional Development Fund (ERDF) and co-financed by several regions of the Meuse-Rhine Euroregion and partners of the EMuRgency consortium.

REFERENCES

[1] J. Elsner, M. T. Schneiders, M. Haberstroh, D. Schilberg, and S. Jeschke, "An Introduction to a Transnational Volunteer Notification System Providing Cardiopulmonary Resuscitation for Victims Suffering a Sudden Cardiac Arrest", Proc. of the Fifth International Conference on eHealth (eTelemed2013). IARIA, April 2013, pp. 59-64.

[2] Studie des Instituts für Notfallmedizin und Medizinmanagement (INM) - Klinikum der Universität München, "Untersuchung zum Notarztdienst und arztbegleitenden Patiententransport in Bayern", April 2010.

[3] J. Bahr, "Lay resuscitation in the German rescue system. Some basic information", vol. 10, no. 3, Notfall Rett. Med 2007, pp. 197-200, doi:10.1007/s10049-007-0910-y.

[4] K. Grassl, "Herzdruckmassage im Rahmen der Laienreanimation: Einsatz eines audiovisuellen Echtzeit-Rückmeldesystems". Dissertation, LMU München: Faculty of Medicine, 2009.

[5] J. Elsner, M. T. Schneiders, D. Schilberg, and S. Jeschke, "Determination of Relevant First Aiders within a Volunteer Notification System", Proc. of the International eHealth, Telemedicine and Health ICT Forum for Educational, Networking and Business (Med@Tel 2013), Luxembourg, 2013, pp. 245-249.

[6] D. L. Kasper, E. Braunwald, S. Hauser, D. Longo, J. L. Jameson, and A. S. Fauci, "Harrison's principles of internal medicine", 16th ed, 2004.

[7] PulsePoint Foundation, www.pulsepoint.org, downloadable as "PulsePoint" App on Google Play and Apple iTunes, (accessed 08/12/2013).

[8] The San Ramon Valley Fire Protection District, www.firedepartment.org, "Fire Department iPhone App", (accessed 08/12/2013).

[9] Tracy Rosecrans , "Top 10 Heart Disease Apps of 2012", Healthline article, September 2012.

[10] K. Roebuck, "Location-Based Services (LBS): High-impact Strategies - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Vendors", Emereo Pty Limited, 2011.

[11] Electronic Publication: comScore Reports June 2013 U.S. Mobile Subscriber Market, comScore Incorporation, 2013.

[12] Electronic Publication: International Data Corporation (IDC) - Press Release, "Android and iOS Surge to New Smartphone OS Record in Second Quarter", August 8, 2012.

[13] P. Lubbers, B. Albers, and F. Salim, "Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development", Apress, 2010.

[14] G. Hohpe and B. Woolf, "Enterprise integration patterns: Designing, building, and deploying messaging solutions.", Addison-Wesley, Bosten 2004.

[15] http://stomp.github.io/ (accessed 08/10/2013).

[16] http://developer.android.com/google/gcm/index.html (accessed 08/14/2013).

[17] http://mqtt.org/ (accessed 08/02/2013).

[18] IBM and Eurotech, MQTT V3.1 Protocol Specification. Available: http://www.ibm.com/developerworks/webservices/library/ws-mqtt/index.html (accessed 08/20/2013).

[19] https://www.oasis-open.org/committees/mqtt (accessed 08/18/2013).

[20] V. Lampkin, W. T. Leong, L. Olivera, S. Rawat, N. Subrahmanyam, R. Xiang, G. Kallas, N. Krishna, S. Fassmann, M. Keen and D. Locke, "Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry", IDM RedBooks, September 2012, Available: http://www.redbooks.ibm.com/redbooks/pdfs/sg248054.pdf (accessed 08/20/2013).

[21] D. Bernstein and A. Kornhauser, "An introduction to map matching for personal navigation assistants.", Technical report, New Jersey TIDE Center Technical Report, 1996.

[22] M. A. Quddus, W. Y. Ochieng and R. B. Noland, "Current map-matching algorithms for transport applications: State-of-the art and future research directions", Transportation Research Part C: Emerging Technologies, vol. 15, issue 5, October 2007, pp. 312-328, ISSN 0968-090X, http://dx.doi.org/10.1016/j.trc.2007.05.002.

[23] W. Ochieng, M. Quddus and R. Noland, "Map-Matching in complex urban road networks", Brazilian Journal of Cartography (Revista Brasileira de Cartografia), vol. 55, pp. 1–18, 2003.

[24] Y. Lou, C. Zhang, X. Xie, Y. Zheng, W. Wang, and Y. Huang, "Map-Matching for Low-Sampling-Rate GPS Trajectories," GIS '09 Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 352–361, 2009.

[25] S. H. Shin, C. G. Park, and S. Choi, "New Map-Matching Algorithm Using Virtual Track for Pedestrian Dead Reckoning," ETRI Journal, vol. 32, no. 6, pp. 891-900, 2010.

[26] J. Eisner, S. Funke, A. Herbst, A. Spillner, and S. Storandt, „Algorithms for Matching and Predicting Trajectories", 13th Meeting on Algorithm Engineering and Experiments (ALENEX), pp. 84-95, 2011.

[27] F. C. Pereira, H. Costa and N. M. Pereira, "An off-line map-matching algorithm for incomplete map databases", European Transport Research Review, vol. 1, issue 3, Springer, pp. 107-124, ISSN 1867-0717, 2009.

[28] Z. Yan, S. Peng, J. Zhou, J. Xu, and H. Jia, "Research on an improved dead reckoning for AUV navigation", 2010 Chinese Control and Decision Conference (CCDC), pp. 1793-1797, 2010.

[29] C. D. C. Randell and H. L. Muller, "Personal position measurement using dead reckoning", in Proceedings of the Seventh International Symposium on Wearable Computers, C. Narayanaswami, Ed. IEEE Computer Society, October 2003, pp. 166–173.

[30] S. Beauregard and H. Haas, "Pedestrian Dead Reckoning: A Basis for Personal Positioning", Proceedings of the 3rd Workshop on Positioning, Navigation and communication (WPNC'06), 2006.

[31] A. Leonhardi, C. Nicu, and K. Rothermel, "A map-based dead-reckoning protocol for updating location information", Parallel and Distributed Processing Symposium., Proceedings International, IPDPS, December 2001.

[32] D. Vasquez, T. Fraichard, O. Aycard, and C. Laugier, "Intentional motion on-line learning and prediction", Machine Vision and Applications, vol.19, no. 5-6, pp.411-425, 2008.

[33] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti, "WhereNext: a location predictor on trajectory pattern mining", KDD 2009, pp. 637–646, 2009.

[34] S. Gaffney and P. Smyth, "Trajectory clustering with mixtures of regression models", in Knowledge Discovery and Data Mining, pp. 63–72, ACM Press, 1999.

[35] J. Alon, S. Sclaroff, G. Kollios, and V. Pavlovic, "Discovering clusters in motion time-series data", 2003.

[36] P. Meisen, T. Seidl, and K. Henning, "A Data-Mining Technique for the Planning and Organization of Truck Platoons", Proceedings of International Conference on Heavy Vehicles (HHVT2008), Paris, 2008.

[37] F. Giannoti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining", KDD 2007, pp. 330-339, 2007.

[38] I. H. Witten, E. Frank, and M. A. Hall, "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Amsterdam, 3rd ed, 2011.

[39] J. Bahr, H. Klingler, W. Panzer, H. Rode, and D. Kettler, "Skills of lay people in checking the carotid pulse", Resuscitation 35, August 1997.

[40] K. Fehn, Legal Advice for the project "SMS Retter", pp. 13-20, November 2009, unpublished.

[41] M. Hutter, "Universal Artificial Intelligence - Sequential Decisions based on Algorithmic Probability", Springer, 2005.

[42] L. Richardson and S. Ruby, "RESTful Web Services", O'Reilly Media, 2007.

[43] Google Maps API V2 as part of the Google Play services SDK for Android (https://developers.google.com/maps/documentation) (accessed 08/14/2013).

[44] R. Meier, "Professional Android 4 Application Development", Wrox Press, May 2012.

[45] Vaadin}> "Thinking of U and I", developer homepage: (https://vaadin.com/home) (accessed 08/10/2013).

[46] M. Pollack, T. Risberg, O. Gierke, C. Leau, J. Brisbin, and T. Darimont, "Spring Data MongoDB - Reference Documentation ", (http://docs.spring.io/spring-data/data-mongo/docs/1.3.3.RELEASE/reference/html/) (accessed 08/20/2013)

[47] S. Thelen, S. Bergrath, M.-T. Schneiders, J. Brokmann, T. Brodziak, D. Schilberg, R. Rossaint, and S. Jeschke, "Telemedizin in der präklinischen Versorgung - das Forschungsprojekt TemRas", e-Health2012, Medical Future 2011, pp. 241-244.