Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2020) 28: 2507 – 2522
© TÜBİTAK
doi:10.3906/elk-1911-5

Prominent quality attributes of crisis software systems: a literature review

Ahmet Arif AYDIN[*]
Department of Computer Science, Faculty of Engineering, İnönü University, Malatya, Turkey

Received: 01.11.2019 • **Accepted/Published Online:** 28.01.2020 • **Final Version:** 25.09.2020

Abstract: Developing software systems to meet user-demanded functionality is critical. Achieving the design goals by providing the needed functionality is a necessary task, and it is about figuring out a proper set of quality attributes and implementing each one by reflecting a complete set of quality attributes. This study presents popular quality attributes of crisis software systems by conducting a literature review. Each crisis software system has been studied by concentrating on crisis management phases where the system is used, design purposes, and the data processing style. The findings of this research shed light on the crisis software development process by presenting a quality attribute-oriented perspective, addressing design challenges, and recommending to developers remedies to handle challenges.

Key words: Software systems, crisis software, quality attributes, crisis informatics, crisis management

1. Introduction

We are surrounded by a digital world enabling the public community to use various portable devices such as smartphones, tablets, and laptops with fast and ubiquitous Internet access. This digital environment has significantly changed most of the services provided by government offices, corporations, healthcare, and humanitarian organizations [1]. Furthermore, emergencies such as earthquakes, floods, storms, or fires constitute an inevitable part of our lives and the public community cannot avoid the existence of these natural disasters. Thus, the government and public society must be ready to deal with them. Conventional one-way crisis management has also been changed to benefit from the contemporary digital environment. In contemporary crisis management, crisis informatics researchers play a role to support emergency management by creating collaboration channels among government offices, police departments, humanitarian organizations, fire and rescue teams, first-aid crews, and public community members such as volunteers, witnesses, and people in need [2]. The major requirement to conduct crisis informatics research is having crisis software [3]. In addition, it is essential to develop crisis software systems for the purpose of supporting crisis management during and after emergencies. For this reason, developing the right crisis software system is important, and it requires defining the right set of quality attributes, which can be achieved by carefully identifying quality attributes in accordance with user demands. Thus, developing the right crisis software systems can be achieved by having knowledge of existing crisis software systems and their design rationales, understanding crisis management phases, and identifying relations among software quality attributes. To connect these ideas, the main contributions of this paper are explained next.

[*]Correspondence: arif.aydin@inonu.edu.tr

1.1. Main contributions

The main contribution of this study is a software quality attribute-oriented literature review of crisis software systems. To the best of the author's knowledge, such a literature review especially for crisis software systems has not been conducted so far and this paper presents popular quality attributes of crisis software systems by focusing on crisis management phases, discussing design purposes, and providing the data processing performed in crisis phases by the software. Moreover, developing the right software by achieving a set of requested functionalities is a challenging process, and it requires having domain knowledge, using the right combination of technologies, identifying a set of quality attributes, and choosing the appropriate software architectural style. To support the software development process, other contribution of this paper are providing crisis software development-related challenges, presenting potential remedies to overcome the identified challenges, and providing the prominently used quality attributes to provide insights for developers to carry out the required functionality.

1.2. Review method and limitations

This literature review started by searching for the terms "software quality attributes", "crisis software and applications", and "crisis management" in IEEE Explore and Google Scholar and performing a cross-reference check. The search results for each term are respectively 2014, 808, and 109. Table 1 presents the number of hits per each term and a list of the cited papers for each one. After checking the content manually, the contribution, and the references of these papers, 8 for "crisis management", 17 for "software quality attributes", and 20 for "crisis software and applications" were studied and cited. These papers were studied by focusing on the following research questions:

- RQ1: What is the design purpose of the examined crisis software system?
- RQ2: Which quality attributes were implemented in the examined crisis software system?
- RQ3: What types of data processing and analysis were performed by the examined crisis software system?

There are two potential limitations pertaining to this paper. First, there may be crisis software systems not included in this study because of the lack of adequate information (i.e. commercial products), or not mentioning the quality attributes, or not answering the addressed research questions. Second, the studied papers are only publicly reachable papers written in English or Turkish, and there may exist relevant papers not publicly available or not written in English or Turkish.

Table 1. Search terms and numbers of papers.

Search terms	IEEE Explore	Google Scholar	References
Crisis management	109	84	2, 13–19
Software quality attributes	2014	3780	4–11, 20–28
Crisis software and applications	808	22	29–48

Software quality attributes research has been gaining the attention of researchers on various subjects such as software systems development [4–6], software maintenance and testing [7], software quality models [8], software architecture [9], selection and prioritization of quality attributes [10], and trade-offs and relations among quality attributes [11]. In particular, this paper aims to present a software quality attribute-oriented literature review for crisis software systems and applications. In addition, this paper is a significantly restructured and extended version of [12]. The rest of the paper is organized as follows: Section 2 provides background information for crisis informatics research, crisis management, and software quality attributes. Section 3 presents features and explanations of the reviewed crisis software systems. Section 4 provides the findings of this literature review, design challenges, and potential remedies. The last section concludes the paper.

2. Background

This section aims to establish grounds for this research by presenting an overview of the following topics: crisis informatics, crisis management, and software quality attributes. Each topic is explained next.

2.1. Crisis informatics and management

Crisis informatics is an emerging research field that aims to respond to emergencies by addressing sociotechnical concerns with consideration to include not only official responders but also members of the public in support of emergency response [2]. Crisis informatics researchers study how members of the public use cutting-edge information and communications technologies and social media during emergencies, explore how publicly available social media data can be beneficial for supporting crisis management and response, and examine how the public can be encouraged to act collaboratively during mass emergencies to increase situational awareness that indicates the desired manner of individuals and the public to comprehend difficulties of an emergency and act consciously [13, 14]. In the crisis informatics literature, time-oriented crisis management is presented in four phases: preparation (mitigation), warning (preparedness), response, and recovery. Contemporary crisis informatics research plays a role between officials and the public to respond to emergencies with the purpose of creating a social bridge between officials and the public [2]. Figure 1 shows that the time-oriented four phases under high-level stages are before (precrisis), during (crisis), and after (postcrisis).

2.1.1. Preparation

The preparation phase is a long-term precrisis phase considered as a precautionary phase before any natural disaster occurs [15]. The vital activities of the preparation phase include arranging risk awareness, improving the existing conditions, enhancing readiness, and strengthening the level of preparedness of government agencies and the public to handle the consequences of emergencies. Government agencies such as AFAD[1] or FEMA[2] and humanitarian organizations are responsible for acting during this phase.

2.1.2. Warning

The warning phase is a precrisis phase that requires increasing situational awareness in the public community by training and teaching the public community how to act during an emergency [16]. This phase also requires preparing strategies to reinforce readiness in emergency situations [17].

2.1.3. Response

The response phase is a crisis phase that requires putting emergency strategies into action to support victims with fast, accurate, and immediate solutions. First aid and rescue teams perform immediate operations to satisfy urgent needs and to keep down human and property loss. During this phase evacuation plans are put

[1] AFAD (2020). Disaster and Emergency Management Presidency [online]. Website https://en.afad.gov.tr/ [accessed 12 Feb 2020].

[2] FEMA (2020). Federal Emergency Management Agency [online]. Website https://www.fema.gov/ [accessed 12 Feb 2020].

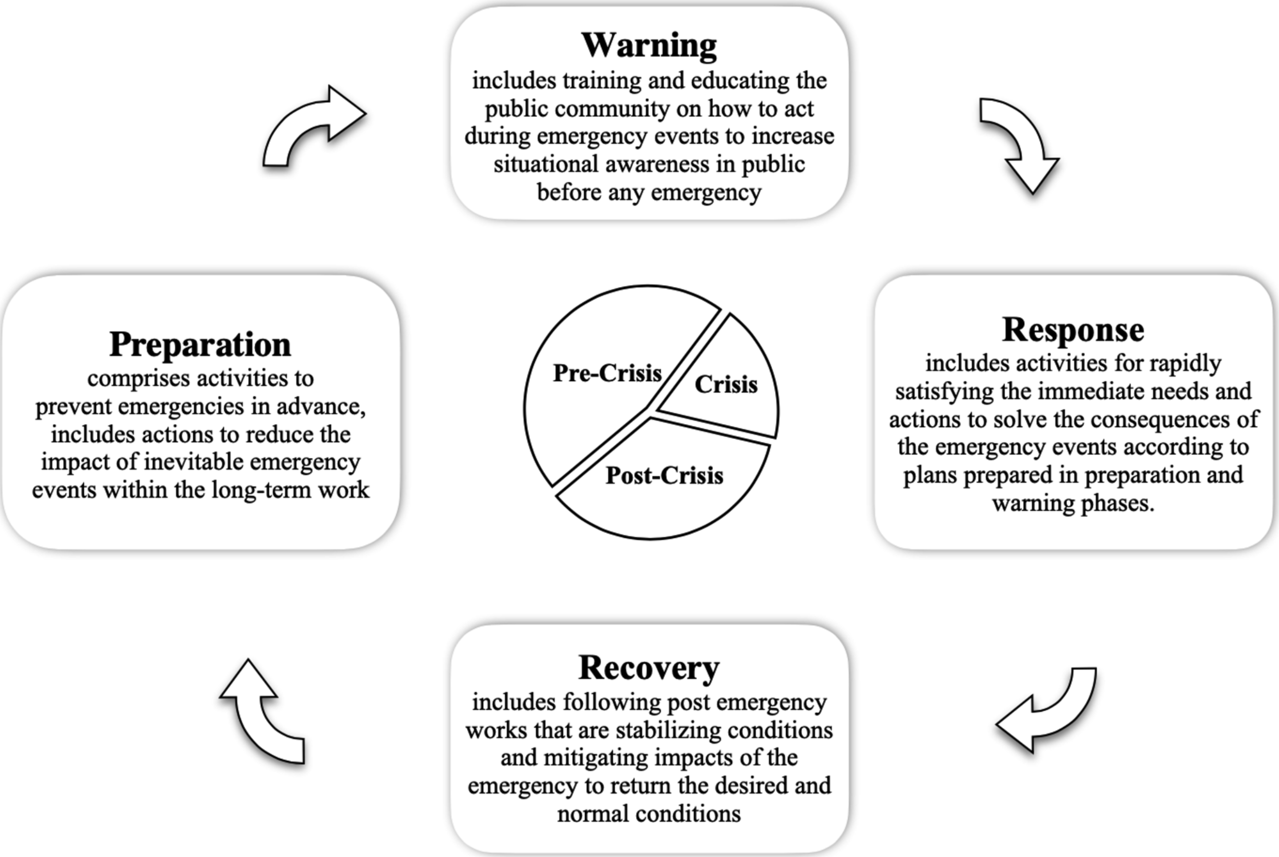
Warning
includes training and educating the public community on how to act during emergency events to increase situational awareness in public before any emergency
Preparation
comprises activities to prevent emergencies in advance, includes actions to reduce the impact of inevitable emergency events within the long-term work
Pre-Crisis
Crisis
Post-Crisis
Response
includes activities for rapidly satisfying the immediate needs and actions to solve the consequences of the emergency events according to plans prepared in preparation and warning phases.
Recovery
includes following post emergency works that are stabilizing conditions and mitigating impacts of the emergency to return the desired and normal conditions

Figure 1. Crisis management life cycle.

into action to transfer victims to safe places in order to provide shelter. Search and rescue teams find people in need to save lives and provide medical treatments [18].

2.1.4. Recovery

Recovery is a long-term postcrisis phase requiring collaboration among government agencies, humanitarian organizations, and the public to protect the community from the physical and mental effects of the emergency. During recovery the following actions are performed: defining safe and unsafe zones, relocating people in unsafe zones, assessing and rebuilding damaged homes and structures, and checking the safety of transportation options and roads. The main purpose of recovery is to stabilize conditions after emergencies and to support the public to return to normal conditions [19].

2.2. Software quality attributes

In software quality attributes research, a set of user-demanded software system functionalities has been described by the following terms: software quality, quality properties, quality features, quality attributes, nonfunctional requirements, customer value, and defect level [20]. In this paper, "quality attributes" is used to describe "a set of features and characteristics of a product or a service continuing its capability to meet the desired needs" [21]. Furthermore, developing software systems to achieve a set of quality attributes to meet needs is nearly impossible without acknowledging the impact of software architecture [22] since software architecture affects both operational and evolutionary quality attributes [23]. Therefore, choosing the proper architectural style is a fundamental process in accomplishing the quality attribute-oriented design purposes of crisis software.

In [12], software quality attributes were presented within two categories that are executional and evolutionary for detailed quality attribute definitions [24]. Table 2 shows executional (operational) quality attributes and the impact of these quality attributes observed in action, during which users actively perform their tasks [25]. Table 3 presents evolutionary (developmental) quality attributes defined as "capability of a set of static attributes of a software product to satisfy the stated and implied needs when the software product is used under specified conditions" [26, 27]. Hence, evolutionary quality attributes are important for developers because they support them in maintaining software systems, implementing new functionality requests, evolving the software in use, and enhancing the existing features [20, 28]. In addition, compliance is considered a subquality under all main quality attributes as it is related to the entire software development life cycle to conform to the standards, regulations, and conventions.

Table 2. Executional quality attributes.

Quality attribute	Related quality attributes
Efficiency	Performance, time behavior, resource behavior
Reliability	Availability, fault tolerance, recoverability, robustness, maturity
Usability	Understandability, learnability, operability
Functionality	Suitability, accuracy, interoperability, security
Scalability	Efficiency, performance

Table 3. Evolutionary quality attributes.

Quality attribute	Related quality attributes
Maintainability	Reusability, changeability, testability, simplicity, flexibility, analyzability, stability, modularity
Portability	Adaptability, installability, replaceability, coexistence

3. Reviewed crisis software systems and applications

This section presents information about the reported crisis software systems and applications in an order according to publication dates. Table 4 gives an outline of the examined papers, design purposes, and implemented quality attributes. The deNIS [29] was developed for the German Ministry of International Affairs to create a central platform for collecting distinct types of emergency-related data such as images and location information during a fire. The primary purpose is to centralize the decision support mechanism to promote crisis management by facilitating the collaboration of governmental agencies and the public by using portable devices and sensors. The deNIS was developed by concentrating on performance, scalability, portability, and flexibility quality attributes and it provides real-time knowledge presentation, situation reports, and visualization capabilities. BlueArrow [30] is a web-based decision support system used in emergency evacuation planning that generates evacuation routes for victims to move them as soon as possible to a safe place according to the latest situations of traffic congestion, road networks, and weather conditions by providing real-time data processing capabilities. The authors' focus was performance and scalability. The FireGrid [31] infrastructure provides a sophisticated computational environment to support the decision-making process in large-scale emergencies. The initial focus of this system was to support incident commanders responsible for carrying out dispatch and offense/defense decisions for fire-fighting scenarios. FireGrid supports taking control of fire in a building by getting data in almost real time by using live sensors. The system provides high-performance computing resources and predictive models to infer incident conditions. Availability, robustness, performance, scalability, and security quality attributes were implemented to achieve the design purposes. TwitInfo [32] was developed to summarize events, find peaks of high tweet activity (scalability) with a novel streaming algorithm, and visualize sentiments of Twitter data in real time (performance). Furthermore, the usability quality achieved by the user interface makes it easy for users to browse an enormous number of tweets making use of a timeline-based display, to reach subevents, and to explore tweets by geolocation information, sentiment, and popular URLs. EPIC Collect [33], Project EPIC's Twitter data collection system, provides the opportunity of collecting tweets with specific contents based on a well-defined list of keywords in a 24/7 available, reliable, flexible, scalable, and concurrent fashion to support crisis management by providing real-time data processing and analytic capabilities. Senseplace2 [34], a web-based visual analytics application, serves to extract geographically grounded situational awareness information by using geolocation information. Senseplace2 enables collecting crisis-related tweets via Twitter and provides an interactive page comprising a query panel, timeline display/control, map, list of tweets, and history view to perform an interactive analysis by achieving usability and readability, and supports extensibility and interoperability quality attributes to handle diverse data formats. In [35], a web application for Twitter data analytics was developed. This application allows dealing with substantial amounts of Twitter data to achieve performance at scale in a batch-oriented fashion. The primary focus of this work was scalability and performance. The application can support long-term statistical data processing and analytics in the response, recovery, and preparation phases of crisis management. ESA-AWTM [36] provides emergency situational awareness information for the Australian Government's Crisis Coordination Centre. The primary purpose is to identify situational awareness information for crisis coordinators by tweets generated during the response phase of emergency situations. The interface of ESA-AWTM allows users to watch and refresh alerts to queries of interest in real time and the Burst Detector/Alert Monitor provides visualization capability for incident status by using stylized words based on statistical models. ESA-AWTM was developed by focusing on availability, accessibility, and performance.

In [37], a software architecture was constructed for automated tweet gathering in a defined territory and semantic analysis. This architecture collects tweets from Twitter by a Twitter4j application that makes use of coordinate information and transfers tweets into a PostgreSQL database using the PostGIS spatial extension by addressing scalability, performance, reliability, and fault-tolerance quality attributes. Additionally, a Django web application was developed to facilitate exporting the results of queries via a map or csv. Wordnet and Solr are used to find tweets sharing the same meaning. Twitris [38] is a citizen sensing platform for gathering and analyzing the data generated by diverse sources (blogs, news). To glean deeper insights and useful information, natural language processing techniques are applied to millions of tweets about a variety of events ranging from entertainment to disasters. Unfortunately, the developers of Twitris do not reveal the technologies and systems used to make analysis at this level of scale. An optimized version of the Virginia Tech alert system was presented in [39] to deal with performance issues of the previous version. This crisis system informs students during emergency conditions by a smart phone application. The system performs functionality according to user geolocation information and reliably sends information by SMS or makes phone calls for subscribed students.

The main design focus of the system is achieving performance and reliability quality attributes. SmartRescue [40] is a smartphone-based communication framework that enables gathering geolocation information via smartphone sensors and sharing data with emergency responders. This framework provides emergency-specific machine learning techniques to process sensor data in a batch-oriented fashion for extracting useful information to support people in need during emergencies. Besides, it provides mapping and visualization capabilities. The authors concentrated on flexibility and robustness quality attributes, and they will focus on usability to provide a user-friendly platform. EARS [41], developed for the Italian National Institute of Geophysics and Volcanology (INGV), is a decision support system making use of Twitter to facilitate detection and damage assessment of earthquakes in Italy by using data mining and natural language processing techniques. The system alerts government authorities and earthquake-affected people via emails or direct messages about the consequences of the most recent earthquakes. Robustness, reliability, and performance are critical to fulfill real-time messaging and detection purposes. The EPM [42] is a web-based crowd work platform that aims to report, match, and find lost-and-found pets during and after emergencies to unite them with their families. The rationale behind the EPM is to collaboratively manage the lost-and-found task via digital crowds by using web technology. The authors developed the EPM by concentrating on usability, adaptability, and reliability quality attributes for systematically handling the lost-and-found task through digital crowds.

The EPIC Analyze [43] application provides 24/7 data collection and enables batch processing and real-time analysis capabilities on tweet datasets for crisis informatics researchers. The application allows answering all deep queries applied to the entire JSON object of tweets stored in Cassandra[3] with a comprehensive search functionality using semantic keywords and geoqueries. Researchers can export the results of queries into a map or csv format. EPIC Analyze provides scalability, performance, fault-tolerance, and functionality quality attributes to support crisis management by handling collection and analysis of large volumes of tweet datasets. The authors of DisasterBox [44] aimed to create an integrated communication platform to enable a two-sided communication channel among officials and the public for distributing the most up-to-date information about a natural disaster. The platform provides performance, flexibility, and fault-tolerance quality attributes. Officials can also use this tool to guide victims of a disaster for information transmission. The CyberGIS [45] framework provides an integrated platform for handling and analyzing social media and socioeconomic data generated during emergency conditions. The platform enables geovisualization, spatial and text mining, real-time event tracking, map generation, and statistical analysis to support crisis management. The developers of CyberGIS dealt with scalability and performance challenges by using contemporary data processing and storage technologies such as Hadoop, Hive, Mahout, and Lucene. The EPIC-OSM [46] framework enables collaboration among public individuals to create a crisis map after an emergency and allows people to contribute to the quality of the map in an emergency-affected zone for emergency response endeavors. The authors presented the challenges of the analysis in OpenStreetMap (OSM) and they explained how the challenges were handled to achieve efficiency, reusability, extensibility, scalability, interoperability, and availability quality attributes. IDCAP [3, 47] provides incremental collection and indexing on Twitter data, real-time analytics at interactive speeds, and highly concurrent batch-oriented data processing. The design of IDCAP makes possible scalable, reliable, 24/7 available, efficient real-time data collection and analytics at interactive speeds by carefully putting together Redis, Cassandra, Apache Spark, and sophisticated algorithms. The system also provides a user-friendly web interface to simplify data analysis tasks. The Indoor-Outdoor Viewer application [48] was developed to visualize and get information about the present conditions of an emergency area before emergency responders

[3] Cassandra (2020). Apache Cassandra [online]. Website http://cassandra.apache.org/ [accessed 12 Feb 2020].

Table 4. Crisis software systems and applications.

Crisis software	Design purpose, data processing style	Quality attributes
deNIS [29]	Crisis data collection and management, real-time	Performance, scalability, portability, flexibility
BlueArrow [30]	A web-based decision support system for emergency evacuation, real-time	Performance, scalability
FireGrid [31]	An infrastructure for crisis response, real-time	Availability, robustness, performance, scalability, security
TwitInfo [32]	Visualizing and summarizing events on Twitter, real-time	Scalability, performance, usability
EPIC Collect [33]	Crisis data collection and analytics, system, real-time	Scalability, availability, robustness, flexibility, reusability
SensePlace2 [34]	A web-based geovisual analytics application, real-time	Extensibility, interoperability, usability, readability
McTaggart's Web application [35]	Twitter data analytics application, batch-oriented	Scalability, performance
ESA-AWTM [36]	Supplying situational awareness information, real-time	Availability, accessibility, performance
Twitter4j [37]	An architecture for spatial and semantic analysis of Twitter data, batch-oriented	Scalability, performance, reliability, fault-tolerance
Twitris [38]	Citizen sensing platform, real-time	Scalability, flexibility
Virginia Tech alert system [39]	User location-oriented optimized alert system for emergencies, batch-oriented	Performance, reliability
SmartRescue [40]	Smartphone-based communication framework, batch-oriented	Flexibility, robustness
EARS [41]	Earthquake alert and report system, real-time	Robustness, reliability, performance
EPM [42]	A web-based crowd work environment, batch-oriented	Usability, adaptability, reliability, modularity, reusability
EPIC Analyze [43]	Twitter data collection and analytics platform, real-time	Scalability, performance, fault-tolerance
DisasterBox [44]	Emergency social media channel for communication, real-time	Performance, flexible, fault-tolerance
CyberGIS [45]	Event tracing and mapping, real-time	Scalability, performance, reliability, flexibility
EPIC-OSM [46]	Collaborative crisis mapping, real-time	Efficiency, extensibility, scalability, interoperability, availability, reusability
IDCAP [47]	Twitter data collection and analytics platform, real-time	Scalability, reliability, availability
Indoor-Outdoor Viewer [48]	A tool for emergency responders using Twitter, real-time	Usability, extensibility, interoperability

arrive in the area. This software facilitates viewing indoors and outdoors together and permits marking the disaster area map to organize responders' actions on a desktop or mobile device. The authors highlighted usability, extensibility, and interoperability quality attributes.

4. Findings and discussion

In this section, a high-level view of the quality attributes in the big picture of crisis management phases is provided to support crisis management and software development. Crisis software development-related challenges and potential remedies for challenges are then presented.

4.1. Quality attributes-oriented challenges and remedies

The previous section presented the design purpose, achieved quality attributes, types of data processing and analysis (batch or real-time), and primary design phase of the crisis software systems and applications studied. This research reveals that an individual crisis system cannot provide a complete perspective on all quality attributes since each system has a particular design purpose with a partial set of quality attributes. Nevertheless, each crisis software system contributes to the construction of the big-picture view of quality attributes in crisis management phases. Based on the gained information and insights, a high-level view of quality attribute-oriented perspectives for crisis management phases is presented within three categories in Figure 2.

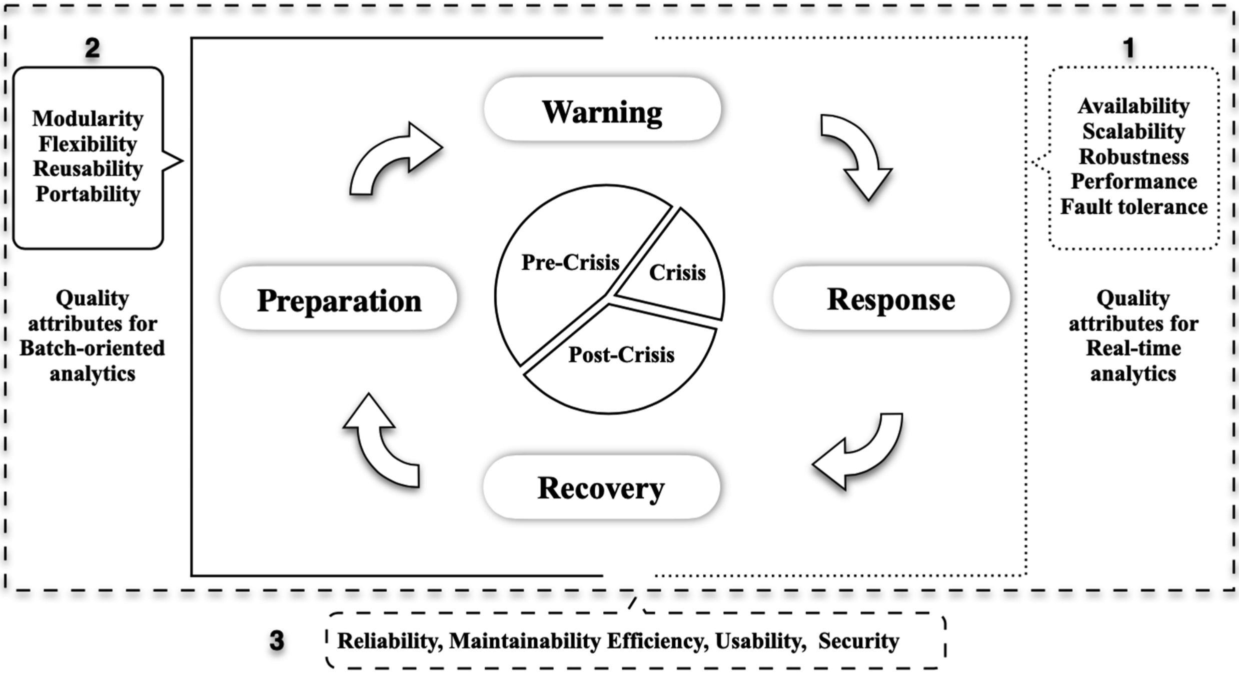

Figure 2. Crisis management phases.

The first group of quality attributes includes availability, scalability, robustness, performance, and fault tolerance, illustrated by dotted lines in Figure 2. These quality attributes are particularly important in the response phase to get the right information at the right time because even seconds matter to save lives, to perform immediate actions, and to dispatch emergency teams at the right time. Thus, crisis software systems developed to support the response phase must have real-time processing and analytic capabilities to get valuable information instantly. The findings of this paper reveal that 80% of the crisis software systems studied were developed to support emergency response with real-time data gathering and analytical capabilities. Furthermore, the crisis software systems constructed for the response phase require real-time data handling capabilities and lack of availability, scalability, robustness, performance, or fault tolerance is not tolerable. The availability ability of a software system allows being a reachable service whenever needed since disasters can strike at any time. For this reason, crisis software systems must be available 24/7 to perform their designated functionality. For example, the FireGrid system [31] aims to detect fires in buildings at the right time. If the system is not available during a sudden fire, this situation demolishes the design purpose. Hence, availability is essential to realize the design purpose of a crisis software system developed for crisis response. The scalability capability makes a software able to perform its functionality without degrading services when the input dramatically increases. For example, before Hurricane Sandy struck, tweets containing the keyword "sandy" were in the hundreds; this number jumped to millions during the disaster. EPIC Collect [33] provided scalable data collection for Hurricane Sandy including about 16M tweets including the keyword "sandy". Thus, to handle such dramatic tweet explosions, crisis software must provide scalable collection to stream substantial amounts of tweets and store the collected tweets into proper data stores without losing [3]. The robustness quality enables software to work accurately under stressful conditions or invalid inputs. Crisis times are troublesome, and crisis software must be robust to deal with various issues. For instance, Twitter allows gathering tweets by Streaming API with rate limits (e.g., delivering 1% of the total tweets). For example, to carry out the rate limit and API connection issues, EPIC Collect [33], EARS [41], and IDCAP [47] make use of multiple accounts to provide robust tweet collections. The performance capability enables software to perform its functions within time and speed constraints. For instance, ESA-AWTM [36] supplies a Burst Detector/Alert Monitor interface that allows retrieving, clustering, and displaying tweets directly from Twitter on the fly and returning results in under a second. Fault tolerance allows software to pursue its routine work through the existence of hardware and software failures. For example, in [37], a fault-tolerant Twitter data collection application was built by setting up separate machines to prevent software or hardware failures.

The second group includes modularity, flexibility, reusability, and portability, which are the quality attributes prominent for software developers to maintain crisis software systems in the desired state within long-term usage and these are illustrated by straight lines in Figure 2. Recovery, preparation, and warning are long-term phases that require handling massive amounts of complete datasets to derive hidden characteristics of past emergencies and get insights to become ready for future emergency conditions. Also, crisis software must have modularity, flexibility, reusability, and portability quality attributes to provide long-term support. The modularity and flexibility quality attributes are significant since both have a direct impact on the evolution of software systems [49]. Modularity enables developers to work in parallel to enhance the existing functionality, carry out new functionality requests, and handle hardware modifications in the long term. Flexibility enables software to adapt to new modification requests without resistance. For example, EPIC Collect achieves flexibility by using Spring MVC[4] to run the software on a vast variety of servers while permitting integration of new service requests without disrupting the functionality of the working services [33]. Reusability is a key quality attribute for saving time and effort because it enables developers to make use of the existing, accurately functioning and error-free components of the software instead of implementing new ones from scratch [49]. Consequently, reusability supports incremental development to fulfill diverse user needs. To exemplify, in [33], the Spring framework provides a testable and reusable code base by making use of dependency injection to satisfy various user demands. Besides, reusability enables the creation of a common code base for the existing functionality needed by users. For instance, in EPIC-OSM [46], all users follow the same rules for querying data with the aid of reusability. The portability quality allows software to become platform-free and enables the software

[4] Spring (2020). Spring Framework [online]. Website https://spring.io/ [accessed 12 Feb 2020].

to perform functionality under diverse environments. For example, moving large volumes of data to perform batch-oriented processing is not an efficient and desired way because of network constraints, security reasons, and transferring time and costs. Instead, the demanded software can process enormous amounts of data after having portable software in the needed environment. Thus, portability is one of the highly demanded qualities of crisis software [29].

The third group of quality attributes includes reliability, maintainability, efficiency, usability, and security, which are essential for all software systems, not only for crisis software systems, and these are illustrated by dashed lines in Figure 2. The reliability quality enables software to perform the designated functionality whenever required. Reliable software provides the same answer for the same question on a completed dataset in different circumstances. For example, in a complete dataset, the resulting set of tweets belongs to a specific user, or the results of the top ten most retweeted tweets must be the same for search queries at various times [3, 35]. The maintainability quality impacts the responsibilities of developers and makes developers' job easy in implementing new features, improving the existing functionality, and adapting to a changing environment [35]. For example, relational database technologies such as MsSQL, MySQL, and PostgreSQL cannot support storing data under multiple nodes, eventual consistency, or horizontal scalability. To get benefits from these features in crisis software, migrating from relational databases to NoSQL technologies can be a solution [50]. Thus, maintainable software can promote migration and adapting tasks. Efficiency is another highly demanded quality attribute because it allows performing the required functionality in terms of performance, resource, and time behavior goals. Table 4 displays many crisis software systems developed to support performance for batch and real-time processing. Hence, efficiency is necessary for both real-time and batch-oriented data processing software systems. Usability is a notable quality for users since it provides remarkable ease in using and controlling software. The interface design for data analysis environments is critical to facilitate tasks of analysis for users without a good programming background [51, 52]. For instance, the usability quality of TwitInfo [32] was achieved by designing a user-friendly interface based on gathered user feedback. Security is a vital quality attribute for all software systems, not only for crisis software. Designing software systems to protect users' personal information is necessary [53]. Unfortunately, there always exist vicious people who intend to intercept software systems to benefit from others' valuable information even under crisis conditions. Crisis researchers use public data that are also reachable by vicious people who can propagate misinformation to mislead officials and delay emergency response. Thus, security is a critical quality attribute for crisis software [54].

4.2. Other challenges and future directions

There are various challenges about the development, maintenance, and evolution processes of crisis software systems such as understanding domain needs, identifying design purposes, determining the right architectural style, defining a proper set of quality attributes, and using the right combinations of contemporary technologies. The overall quality of a software system is assessed by considering whether it addresses a set of demanded functionalities [23] or not. Thus, understanding the domain needs and figuring out the design purposes is a paramount and challenging task for developing the desired high-quality software. To handle the challenge of conceptual gaps between users, crisis managers, and developers of software systems, the foremost remedy is enabling collaboration between users, crisis managers, and developers to identify the requests about demanded software systems and implement the desired set of functionalities. An essential remedy to overcome conceptual gaps between users and developers is to organize regular meetings that allow developers and users to be on the same page by understanding user-requested functionality and prioritizing delivering urgent features first. For example, in Project EPIC, crisis analysts' requests were determined by collecting user feedback. The most requested statistics were the volume of datasets, tweet count for the different date ranges, number of unique tweets in a dataset, top ten most retweeted tweets, and top ten URLs [35]. For instance, user's feedbacks are received for improving collaborative crisis mapping in EPIC-OSM [46].

Moreover, prototyping is another remedy to close the theoretical and practical gaps between developers and users. Since a prototype is a minimal working version of the required software, users can find a chance to observe their conceptual demands in action and the prototype allows users to figure out the misinterpretations between them and the developers. Thus, especially identifying the conflicts in the initial stages of the development significantly influences the success of building the right functionality. For example, figuring out usability challenges in EPM [42], checking implemented functionality in SmartRescue [40], and collecting user feedback for future needs in Indoor-Outdoor Viewer [48] were done by allowing users to use prototypes in their tasks. Software architecture plays a key role in achieving user-demanded quality attributes as it provides a big-picture view of software systems, which allows developers to internalize the responsibilities of each component and relationships among software components [55]. For this reason, developing a software architecture in isolation without considering quality attributes is impossible since software architecture of a system does not merely affect achieving the desired quality attributes but also the future development and maintenance of the software [21]. Choosing an appropriate architectural style is an important topic and beyond the focus of this literature review; however, the method proposed in [28] helps to manage the challenge of selecting a proper architectural style. Furthermore, identifying quality attributes in the initial stages of software development is a challenging process. Figure 2 provides a big picture of quality attributes regarding the data processing styles in crisis management phases to overcome this challenge. To overcome challenges of prioritizing quality attributes, understanding relations between qualities attributes, and making trade-offs, the graph representation of quality attributes presented in [11] can be one potential remedy. This graph represents quality attributes and relations presented by seven types of arcs that include antagonistic, trade-off, impact-on, means-to, and complementary and functional dependency to show the effect on other quality attributes. In addition, concentrating on only one quality attribute is not beneficial during software development as it causes losing the big-picture view of the desired quality attributes and affects the functionality of other quality attributes in a software system. Thus, Figure 2 enables developers to keep a comprehensive view of prominent lists of quality attributes implemented in crisis software systems.

Using the right combination of contemporary technologies, engaging with sophisticated approaches, using parallel processing techniques, and making the right technology trade-offs are important for developing the demanded software. The studied crisis software systems allow collecting, storing, and analyzing streaming data in real time or batch data processing capabilities. A few important challenges related to streaming the data are handling the generation of fast and large volumes, dealing with diverse formats, and providing 24/7 available, reliable, scalable, and fault-tolerant collections. For example, in IDCAP [47], this includes handling the fast data achieved by using a queuing technology (RabbitMQ) for not losing data and in-memory data store Redis[5] to enable instantly reaching the data. The data generated by various sources and various formats introduced data modeling, scalable storage, and interoperability challenges. Relational databases were used in the following crisis software systems: FireGrid [31], TwitInfo [32], and Twitter4j [37]. Also, NoSQL technologies were used for managing flexible data modeling, achieving horizontal scalable storage, gaining no single point of failure

[5] Redis (2020). Redislabs [online]. Website https://redis.io/[accessed 12 Feb 2020].

feature, and providing tools for efficient analysis. For example, DataStax Enterprise (DSE)[6] is used in EPIC Collect [33], EPIC Analyze[43], and IDCAP [47]; MongoDB is used in SmartRescue [40] and Disasterbox [44]. To handle the interoperability challenge, making a proper database technology trade-off such as using the proper database (NoSQL, RDBMS, or in-memory), making use of hybrid databases, designing proper data models, and engaging with sophisticated querying techniques are highly important [56].

5. Conclusions

Developing the right software systems to achieve a set of requested functionalities is a challenging process. This literature review on crisis software systems and applications aims to support crisis management and to provide guidance for crisis researchers and the developers of crisis software systems by the following contributions. First, a comprehensive explanation of the crisis software systems and applications is provided. Each crisis software system is explained regarding prominent quality attributes, crisis management phases, design rationales, and the data processing performed in crisis phases. Second, this study allows developers to gain insights on which quality attributes are mostly implemented and which data processing style is suitable for crisis management phases. Answers to these questions help developers to identify a proper set of quality attributes for their new crisis software development endeavors. Third, relationships between data processing styles (batch or real-time) and quality attributes are represented to suggest a set of quality attributes for future data processing and analytics demands. In addition, crisis software development-related challenges are presented and potential remedies to deal with various challenges are discussed. To conclude, the contributions of this literature review can guide crisis researchers about existing crisis software systems and provide insights for the developers to identify the required quality attributes to carry out the required functionality in their future development efforts.

References

[1] Han H, Yonggang W, Tat-Seng C, Xuelong L. Toward scalable systems for big bata analytics: a technology tutorial. IEEE Access 2014; 2: 652-687. doi: 0.1109/ACCESS.2014.2332453

[2] Palen L, Anderson KM, Mark G, Martin J, Sicker D et al. A vision for technology-mediated support for public participation & assistance in mass emergencies & disasters. In: ACM-BCS Visions of Computer Science Conference; Edinburgh, UK; 2010. pp. 1-12.

[3] Aydin AA. Incremental data collection & analytics the design of next generation crisis informatics software. PhD, University of Colorado, Boulder, CO, USA, 2016.

[4] Gu T, Lu M, Li L. Runtime models for analysing and evaluating quality attributes of self-adaptive software: a survey. In: 12th International Conference on Reliability, Maintainability, and Safety; China; 2018. pp. 52-61.

[5] Jaffar RN, Hussain AAAM, Chiad W. A new model for study of quality attributes to components based development approach. Periodicals of Engineering and Natural Sciences 2019; 7 (3): 1177-1185. doi: 10.21533/pen.v7i3

[6] Arcos-Medina G, Mauricio D. Aspects of software quality applied to the process of agile software development: a systematic literature review. International Journal of Systems Assurance Engineering and Management. Springer India 2019; 10 (5): 867-897. doi: 10.1007/s13198-019-00840-7

[7] Alakus TB, Das R, Turkoglu I. An overview of quality metrics used in estimating software faults.In: IEEE 2019 International Conference on Artificial Intelligence and Data Processing Symposium; Malatya, Turkey; 2019. pp. 1-6.

[8] Nistala P, Nori KV, Reddy R. Software quality models: a systematic mapping study. In: IEEE/ACM International Conference on Software and System Processes; Montreal, Canada; 2019. pp. 125-134.

[6] DSE (2020). DataStax [online]. Website https://www.datastax.com/products/datastax-enterprise [accessed 12 Feb 2020].

[9] Lytra I, Carrillo C, Capilla R, Zdun U. Quality attributes use in architecture design decision methods: research and practice. Computing Springer Vienna 2019; 102 (2): 551-572. doi: 10.1007/s00607-019-00758-9

[10] Senthilkumar R, Arunkumar T. A survey on prioritization of software quality attributes. Indian Journal of Science and Technology 2016; 9 (44): 1-7. doi: 10.17485/ijst/2016/v9i44/86988

[11] Becker S, Happe L, Mirandola R, Trubiani C. Towards a methodology driven by relationships of quality attributes for QoS-based analysis. In: ACM/SPEC International Conference on Performance Engineering; Prague, Czech Republic; 2013. pp. 311-314.

[12] Aydin AA. An overview of quality attributes for data intensive systems in crisis informatics. In: International Artificial Intelligence and Data Processing Symposium; Malatya, Turkey; 2017. pp. 1-5.

[13] Salfinger A, Girtelschmid S, Proll B, Retschitzegger W, Schwinger W. Crowd-sensing meets situation awareness: a research roadmap for crisis management. In: Hawaii International Conference on System Sciences; Kauai, HI, USA; 2015. pp. 153-162.

[14] Lee N, Hirschmeier S, Müller S, Luz L J. Enablers in crisis information management: a literature review. In: 50th Hawaii International Conference on System Sciences; Waikoloa Village, HI, USA; 2017. pp. 274-283.

[15] Fischer D, Poseggga O, Fischbach K. Communication barriers in crisis management: a literature review. In: 24th European Conference on Information Systems; İstanbul, Turkey; 2016. pp. 1-18.

[16] Verma S, Vieweg S, Corvey WJ, Palen L, Martin J et al. Natural language processing to the rescue ?: extracting 'situational awareness' tweets during mass emergency. In: Fifth International AAAI Conference on Weblogs and Social Media; Barcelona, Spain; 2011. pp. 385-392.

[17] Mukkamala A, Beck R. Disaster management and social media use for decision making by humanitarian organizations. In: 49th Hawaii International Conference on System Sciences; Kauai, HI, USA; 2016. pp. 1379-1385.

[18] Pavković B, Berbakov L, Vraneš S, Milenković M. Situation awareness and decision support tools for response phase of emergency management: a short survey. In: 25th International Workshop on Database and Expert Systems Applications; Munich, Germany; 2014. pp. 154-159.

[19] Lindsay BR. Federal Emergency Management: A Brief Introduction. Washington, DC, USA: Congressional Research Service, 2012.

[20] Chandrasekar A, Sudharajesh M, Rajesh P. A research study on software quality attributes. International Journal of Scientific and Research Publications 2014; 4 (1): 2250-3153. doi: 10.29322

[21] Losavio F, Chirinos L, Lévy N, Ramdane-Cherif A. Quality characteristics for software architecture. The Journal of Object Technology 2003; 2 (2): 133-150. doi: 10.5381/jot.2003.2.2.a2

[22] Chen L, Babar MA, Nuseibeh B. Characterizing architecturally significant requirements. IEEE Software 2013; 30 (2): 38-45. doi: 10.1109/MS.2012.174

[23] Mattsson M, Grahn H, Mårtensson F. Software architecture evaluation methods for performance, maintainability, testability, and portability. In: 2nd International Conference on the Quality of Software Architectures; Västerås, Sweden; 2006. pp. 1-20.

[24] ISO. Information Technology-Software Product Quality. Geneva, Switzerland: ISO, 2000.

[25] Buschmann F, Ameller D, Ayala CP, Cabot J, Franch X. Architecture quality revisited. IEEE Software 2012; 29 (4): 22-24. doi: 10.1109/MS.2012.77

[26] IEEE. Standard Glossary of Software Engineering Terminology. New York, NY, USA: IEEE, 1990.

[27] ISO/IEC. Systems and software engineering - systems and software quality requirements and evaluation (SQuaRE)– System and software quality models. Geneva, Switzerland: ISO, 2011.

[28] Yang Q, Zhao W. A method of selecting appropriate software architecture styles: quality attributes and analytic hierarchy process. BSc, University of Göteborg, Göteborg, Sweden, 2012.

[29] Wietfeld C, Wolff A, Bieker U. MobileEmerGIS: a wireless-enabled technology platform to support emergency response management. In: IEEE Conference on Technologies for Homeland Security; Woburn, MA, USA; 2007. pp. 51-56.

[30] Ling A, Li X, Fan W, An N, Zhan J et al. Blue Arrow: a web-based spatially-enabled decision support system for emergency evacuation planning. In: International Conference on Business Intelligence and Financial Engineering; Beijing, China; 2009. pp. 575-578.

[31] Han L, Potter S, Beckett G, Pringle G, Welch S et al. FireGrid: an e-infrastructure for next-generation emergency response support. Journal of Parallel and Distributed Computing 2010; 70 (11): 1128-1141. doi: 10.1016/j.jpdc.2010.06.005

[32] Marcus A, Bernstein MS, Badar O, Krger DR, Madden S et al. Twitinfo: aggregating and visualizing microblogs for event exploration. In: ACM CHI Conference on Human Factors in Computing Systems; Vancouver, Canada; 2011. pp. 227-236.

[33] Anderson KM, Schram A. Design and implementation of a data analytics infrastructure in support of crisis informatics research (nier track). In: International Conference on Software Engineering; Honolulu, HI, USA; 2011. pp. 844-847.

[34] Maceachren AM, Jaiswal A, Robinson AC, Pezanowski S, Savelyev A. SensePlace2: GeoTwitter analytics support for situational awareness. In: IEEE Conference on Visual Analytics Science and Technology; Providence, RI, USA; 2011. pp. 181-190.

[35] McTaggart C. Analysis and implementation of software tools to support research in crisis informatics. MSc, University of Colorado, Boulder, CO, USA, 2012.

[36] Cameron MA, Power R, Robinson B, Yin J. Emergency situation awareness from twitter for crisis management. In: 21st International Conference Companion on World Wide Web; Lyon, France; 2012. pp. 695-698.

[37] Oussalah M, Bhat F, Challis K, Schnier T. A software architecture for Twitter collection, search and geolocation services. Knowledge-Based Systems 2013; 37: 105-120. doi: 10.1016/j.knosys.2012.07.017

[38] Purohit H, Sheth A. Twitris v3: from citizen sensing to analysis, coordination and action. In: Seventh International Association for the Advancement of Artificial Intelligence Conference on Weblogs and Social Media; Boston, MA, USA; 2013. pp. 3-4.

[39] Zeitz K, Marchany R, Tront J. Designing an optimized alert system based on geospatial location data. In: 47th Hawaii International Conference on System Sciences; Washington, DC, USA; 2014. pp. 4159-4168.

[40] Radianti J, Gonzalez JJ, Granmo O. Publish-subscribe smartphone sensing platform for the acute phase of a disaster: a framework for emergency management support. In: Fourth International Workshop on Pervasive Networks for Emergency Management; Budapest, Hungary; 2014. pp. 285-290.

[41] Avvenuti M, Cresci S, Marchetti A, Meletti C, Tesconi M. EARS (earthquake alert and report system): a real time decision support system for earthquake crisis management. In: 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; New York, NY, USA; 2014. pp. 1749-1758.

[42] Barrenechea M, Anderson KM, Palen L, White J. Engineering crowdwork for disaster events: the human-centered development of a lost-and-found tasking environment. In: 48th Hawaii International Conference on System Sciences; Kauai, HI, USA; 2015. pp. 182-191.

[43] Anderson KM, Aydin AA, Barrenechea M, Cardenas A, Hakeem M et al. Design challenges/solutions for environments supporting the analysis of social media data in crisis informatics research. In: 48th Hawaii International Conference on System Sciences; Kauai, HI, USA; 2015. pp. 163-172.

[44] Parham K, Shash Y, Wong J, Frazier TZ, Cang L et al. DisasterBox: designing social media for disaster relief. In: 78th ASIS&T Annual Meeting; St. Louis, MO, USA; 2015. pp. 10-13.

[45] Huang Q, Cervone G, Jing D, Chang C. DisasterMapper: a CyberGIS framework for disaster management using social media data. In: ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data; Seattle, WA, USA; 2015. pp. 1-6.

[46] Anderson J, Soden R, Anderson KM, Kogan M, Palen L. EPIC-OSM: a software framework for OpenStreetMap data analytics. In: 49th Hawaii International Conference on System Sciences; Kauai, HI, USA; 2016. pp. 5468-5477.

[47] Aydin AA, Anderson KM. Batch to real-time: incremental data collection & analytics platform. In: 50th Hawaii International Conference on System Sciences; Waikoloa Village, HI, USA; 2017. pp. 5911-5920.

[48] Gelernter J, Maheshwari N, Sussman A. Visualization and communication tool for emergency response. In: IEEE International Symposium on Technologies for Homeland Security; Woburn, MA, USA; 2018. pp. 1-5.

[49] Parnas DL. On the criteria to be used in decomposing systems into modules. Communications of the ACM 1972; 15 (12): 1053-1058. doi: 10.1145/361598.361623

[50] Schram A, Anderson KM. MySQL to NoSQL data modeling challenges in supporting scalability. In: 3rd Annual Conference on Systems, Programming, and Applications: Software for Humanity; Tucson, AZ, USA; 2012. pp. 191-202.

[51] Barrenechea M, Jambi S, Aydin AA, Hakeem M, Anderson KM. Getting the query right for crisis informatics design issues for web-based analysis environments. Journal of Web Engineering 2017; 16 (5): 399-432.

[52] Barrenechea M, Anderson KM, Aydin AA, Hakeem M, Jambi S. Getting the query right: user interface design of analysis platforms for crisis research. In: 15th International Conference on Web Engineering; Rotterdam, the Netherlands; 2015. pp. 663-666.

[53] Baykara M, Daş R. Softswitch: A centralized honeypot-based security approach using software-defined switching for secure management of VLAN networks. Turkish Journal of Electrical Engineering and Computer Sciences 2019; 27 (5): 3309–3325. doi: 10.3906/elk-1812-86

[54] Reuter C, Kaufhold MA. Fifteen years of social media in emergencies: a retrospective review and future directions for crisis informatics. Journal of Contingencies and Crisis Management 2018; 26 (1): 41-57. doi: 10.1111/1468-5973.12196

[55] Garlan D, Shaw M. An Introduction to Software Architecture. Advances in Software and Knowledge Engineering. Pittsburgh, PA, USA: World Scientific Publishing Company, 1993.

[56] Kadadi A, Agrawal R, Nyamful C, Atiq R. Challenges of data integration and interoperability in big data. In: IEEE International Conference on Big Data; Washington, DC, USA; 2014. pp. 38-40.