Intelligent Utilization of Dashboards in Emergency Management

Yaniv Mordecai
Motorola Solutions Israel
yaniv.mordecai@motorolasolutions.com

Boris Kantsepolsky
Motorola Solutions Israel
boris.kantsepolsky@motorolasolutions.com

ABSTRACT

Effective decision-supporting visualization is critical for strategic, tactic, and operational management before and during a large-scale climate or extreme weather emergency. Most emergency management applications traditionally consist of map-based event and object visualization and management, which is necessary for operations, but has small contribution to decision makers. At the same time, analytical models and simulations that usually enable prediction and situation evaluation are often analyst-oriented and detached from the operational command and control system. Nevertheless, emergencies tend to generate unpredictable effects, which may require new decision-support tools in real-time, based on alternative data sources or data streams. In this paper, we advocate the use of dashboards for emergency management, but more importantly, we propose an intelligent mechanism to support effective and efficient utilization of data and information for decision-making via flexible deployment and visualization of data streams and metric displays. We employ this framework in the H2020 beAWARE project that aims to develop and demonstrate an innovative framework for enhanced decision support and management services in extreme weather climate events.

Keywords

Emergency Management, Natural Disasters, Dashboards, Business Intelligence, Decision Support Systems.

INTRODUCTION

2017 saw an onslaught of natural disasters in North America, including a series of devastating Hurricanes Harvey, Irma, and Maria, a severe earthquake in Mexico, wildfires in California, and a massive-scale blizzard that struck the entire central-eastern part of the USA. This trail of devastation caused thousands of casualties, hundreds of billions of dollars in damages, and massive disruptions of normal life – and has shown the difficulty in coping with natural disasters in developing countries like Mexico and Puerto-Rico as well as super-developed countries like the USA. The difficulties and tremendous efforts needed for preparation, absorption, and recovery, have further raised the global awareness of the need for holistic national emergency management agencies, that would be equipped with state-of-the-art technology to run the operation – before, during, and after the emergency, in order to reinstate normal routine as fast as possible, and to prevent the emergency from escalating towards a disaster or a crisis.

beAWARE is an EU-funded project (#700475) to deliver a prototype disaster management system for extreme weather conditions. beAWARE focuses on flood, forest fire, and heatwave scenarios. The beAWARE platform is an end-to-end solution for collecting information from multiple data sources – such as end users, social networks, sensors, and data providers – analyzing it, predicting and assessing emergencies, alerting the public, and managing first responders' activities. (beAWARE 2017).

As part of beAWARE, we were required to provide a comprehensive solution to visualize decision-supporting information to senior authority officials, decision-makers, and stakeholders (e.g., mayor, chief inspector, or head of emergency services). This resulted in the need to consider various options for advanced visualization that would cover the various requirements made by the operational stakeholders who participate in the project.

We decided to tackle this problem as a management problem similar to the one faced by decision-makers and managers in everyday business contexts, through the use of Business Intelligence & Analytics (BI&A) –

techniques, technologies, systems, practices, methodologies, and applications that analyze critical business data to help an enterprise better understand its business and market and make timely business decisions (Chen, Chiang, and Storey 2012).

BI&A is mostly used in business-related applications, such as financial analysis, sales predictions, supply chain management, operational performance tracking. This is mostly due to the compelling need for quantitative metrics in the decision-making processes associated with these business practices, and the availability of the underlying data in enterprise information systems, including Transaction Processing Systems (TPS), Management Information Systems (MIS) Enterprise Resource Planning (ERP), Customer Relationship Management (CRM) and Supply Chain Management (SCM) solutions.

Applying BI&A to emergency management is not as trivial and straightforward as one would assume at a first glance. The challenges include real-time data and data-source availability, decision-making and analysis dynamics, and visualization challenges. All of these may be present in business contexts as well, but are not supposed to be resolved under pressure and time-critical life-saving decisions do not depend on them. This is also probably one of the reasons for the scarce literature on the application of business intelligence dashboards to emergency management in general and in the context of natural disasters specifically.

With a platform like beAWARE in place, the problem of data and data-source availability is significantly reduced, due to the interconnectivity that beAWARE provides among multiple sensor and sources of information, analytic services, and presentation services – all of which are interoperating as a system-of-systems to deliver value to decision makers, control center officers and analysts, rescue teams, and the public.

The main turning point in our approach is the facilitation of a generic architecture, which allows for the reception of any data stream into a consolidated database, flexible definition of data visualization and display settings for each interesting data stream, and an extendable set of services to conduct processing and analysis of the incoming data streams in order to create new, enhanced information streams that provide greater value for the decision-makers.

The rest of this paper is organized as follows. First, we provide an overview of related work in the area of emergency management systems and business intelligence dashboard applications in this domain. Next, we review the publicly-available operational requirements for decision-supported visualization derived from the associated beAWARE Project deliverable. Third, we describe a **framework for effectively delivering and visualizing decision-supporting information to the decision-makers in real-time, before and during an emergency,** which utilizes the end-to-end collaboration facilitated by the beAWARE technology platform. Finally, we conclude the paper and outline further research plans.

RELATED WORK

Emergency Management Systems (EMS)

Emergency management (EM) is the iterative and comprehensive handling of emergency-related tasks, including pre-emergency mitigation, near-emergency preparedness, in-emergency response, and post-emergency recovery. A Command & Control Center (C3) is typically in charge of coordinating the activities of various workforces – police, firefighters, medical teams, and crisis response teams (e.g., hazardous material squads, collapsed building rescue and evacuation forces, etc.) (Dusse et al. 2016).

A study of the command and control (C2) architecture in the wake of the Kobe earthquake in Japan, 1995, has argued that the main goal of the C3 (a.k.a. Emergency Room) is to maximize the efficiency of the disaster response field teams (Kuwata, Ishikawa, and Ohtani 2000). This can be done by: a) real-time map and map-placed object sharing; b) informative image sharing for enhancing situational awareness; c) supporting multi-modal communication including voice (live and recorded), text (typed and handwritten), and map cues; d) monitoring the safety and security of the field team members.

The proposed monitoring, collaboration, and control mechanisms call for the implementation of complementary mechanisms on the side of the C3, including map-based functionality and data visualization and interaction. This has been the dominant approach to emergency management command & control over the past two decades. A similar approach could be found in a command and control system for emergency management, called Command Post, proposed for integrating and visualizing data from different information sources on a single visualization interface, as well as providing a communication and coordination medium for different First Responder Team members (Bakopoulos et al. 2011). The architecture relied on a central Real-time Information Merging and Visualization C2 application. The C2 supports the command team in interactions with First Responders, and controlling the reaction to events during an emergency.

On the other end of emergency response system functionality, the strategic purposes and goals of emergency response decision support systems (ERDSS) are in assisting the authorities to enhance their emergency response capabilities mainly through early warning, contingency planning and plan evaluation, coordinating and commanding emergency response activities, and managing critical resources, and provide *knowledge* (Shan et al. 2012). Considering these roles, the following processing centers (with corresponding modules) and their capabilities were proposed, on top of the tactical management functionality:

- **Emergency Early Warning Service**: collect safety, risk, and asset information; analyze data and predict trends; determine security thresholds; provide continuous risk estimation.
- **Emergency Plan Management Module**: analyze emergency risks and requirements; classify emergency, disaster, and crisis conditions; prepare, model, and simulate emergency plans; test, validate, and analyze emergency plan effectiveness; supervise emergency plan execution; gather improvement requests.
- **Command and Coordination Center**: identify and confirm the severity of the emergency; monitor the emergency response effort; make emergency response decisions; coordinate multiple organizations to conduct rescue work; assess the level of victims' satisfaction; determine the necessary rescue measures.
- **Emergency Relief Supplies Management Module**: determine relief supplies categories and quantities; provide routing algorithms, modeling and simulation; provide logistics operation and coordination; manage relief supplies distributions; collect relief supplies and victims' satisfaction feedback; provide instructions on how to execute emergency rescues.
- **Emergency Knowledge Bank**: store and retrieve emergency management knowledge: procedures, protocols, plans, statistics, historical data and reports, lessons learnt from similar cases, specialist and specialty directory, emergency services directory, population directory.
- **Emergency Finance Budget Management**: provide financial planning, budget allocation, costing, accounting, and overall cost estimates.

Information Visualization (InfoVis) is a research area that focuses on design and development of new presentation approaches, visual layouts, visual interaction methods, data manipulation and transformation, and insight generation for information search, information exploration, and knowledge acquisition, for the purpose of performing various heterogeneous analysis tasks (Nazemi et al. 2015). A study of InfoVis applications in EM classified and ranked the sources of information, visual paradigms, visualization techniques, and interaction techniques used in studies on EM systems for various tasks and scenarios (Dusse et al. 2016). Not surprisingly, the interactive 2d-map was found to be the most common interactive visualization technique. Another interesting finding is that the common techniques of information visualization can be applied in response to any emergency scenario, as most of the studies were generic and only a small portion of them focused on specific scenarios. A significant portion of the applications is around pre-emergency mitigation and preparation, and not only around the response to an ongoing emergency situation. However, the applications for the post-emergency recovery phase are the least-addressed.

The authors did not include clear definitions of the classified techniques; hence it is difficult to understand what exactly is meant by some of the terms they used. The classification of cognitive tasks, such as information searching, event management, task assignment, decision making, data analysis, or interaction with other users, is clearly missing in this study. Hence, the findings cannot be attributed to cognitive tasks that are performed during each phase.

Studies have shown that bombarding the user with information is not necessarily helpful in closing knowledge gaps, and may even generate stress and discomfort among the users due to their inability to cope with the flow of information (Saaty 2008) . Strategies for information visualization to improve clarity and reduce uncertainty include clustering, partitioning, 2D-binning, and abstraction – usually in 3-4 layers (Novotny 2004).

Risk Management

One of the primary purposes of EMS is to reduce risk, which explains why "emergency management" and "risk management" are often used interchangeably. EMS can assist in risk management by promoting decision making for risk probability and impact reduction. Therefore, each object or concept that generates significant risk or could be severely impacted by risk should be clearly visualized to the user, as well as the risk magnitude, in order to promote prioritized treatment. Sources of risk should be clearly distinguished from objects at risk, including persons, assets, and processes.

The Risk Management (RM) literature comes mostly from domains in which RM is a primary and critical activity, e.g., project management, systems engineering, and information technology. Various associations and large-scale organizations have developed RM frameworks, both as part of general management frameworks and as dedicated risk-centered approaches. Leading standard providers are the Project Management Institute (PMI), International Organization for Standardization (ISO), National Association of Space and Aviation (NASA), National Institute of Standards and Technology (NIST) and International Council on Systems Engineering (INCOSE). The ISO 31000 Risk Management Standard defines a framework for risk management, which includes an iterative risk analysis process, and concurrently, ongoing communication and monitoring (Conforti et al. 2011; SRA 2015; Purdy 2010). An illustration of the ISO 31000 framework is shown in Figure 1.

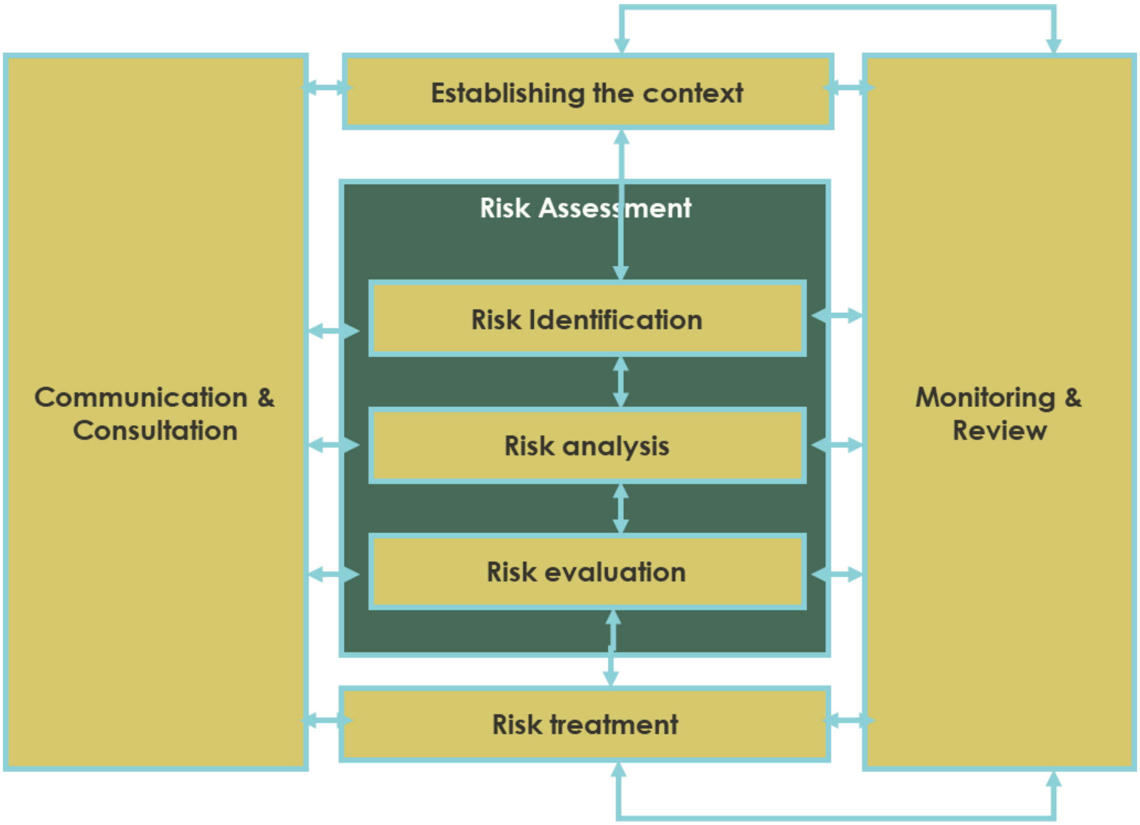

Figure 1. The ISO-31000 Risk Management Process

Business Intelligence and Dashboards in Emergency Management

Business intelligence & Analytics (BI&A) technology, often called "dashboards" by professionals and users alike – due to the resemblance of some of the common metric displays to analog gauges or light panels – has been a critical decision making tool in organizations for the past two decades (Chen, Chiang, and Storey 2012), and has gone through various evolutionary phases, especially reaching new peaks of applicability due to the "big-data" revolution of the past several years.

However, BI&A applications in emergency management remain scarce. This is due to several reasons, according to (Schlegelmilch and Albanese 2014): First, Emergency Management Agencies (EMAs) overemphasize the aggregation and consolidation of information for static presentation, and at the same time underemphasize the requirements for data management and insight creation. Second, available analysis or data presentation available in EMS platforms is typically limited to two-dimensional static data analysis based on end-user uploaded or manually-entered data. This does not allow real information-reliant decision making or insight generation.

A study of interactive dashboards to assist foodbanks in countries struggling with food insecurity, found that users need not only static, raw information to be presented to them, but also the ability to explore, connect, and filter data and data artifacts using the dashboard user interface (Desai, Jiang, and Davis 2016). While these conclusions are probably applicable to dashboards in general, the foodbank case is an interesting combination of economy and emergency that considers the unique characteristics of the food insecurity domain. Food banks receive donations of food and distribute the food through various channels to provide for hungry populations. Making decisions is difficult due to uncertainty on both the supply and demand, and lack of access to structured information at different organization levels.

A study on the communication and visualization of emergency information in emergency management systems in Australia (Surakitbanharn and Ebert 2017) found that the authorities used an on-line dashboard to provide information to the public about the current state-of-affairs, including some counters (number of emergency news reports, road closures, weather warnings, power outages), an on-map display of events, and useful information like contact details and advice to the public on how to react to various incidents. Reportedly, this application has led to the overall situational awareness of the public. However, the study does not indicate whether dashboards were used for enhancing situational awareness among emergency managers and supporting their decision-making processes at various levels.

Finally, trying to balance the amount of information delivered to decision-makers in emergency control centers, to account for the dynamics of presentation and visualization needs, an architecture for emergency management dashboards has been proposed in (Nascimento, Vivacqua, and Borges 2016). The authors observed emergency control centers and learned that: a) emergency managers are overwhelmed with reports and data from multiple sources, b) they often ignore information that may be relevant to the decisions they make, c) information is stored without filtering or processing, d) visualization is static by design, with no possibility of customization, and e) new information requirements are generated and in the absence of supporting functionality the gap is closed by other means such as auxiliary applications, spreadsheets, and workarounds or manipulations on the available functionality. Considering these observations, the authors searched for an architecture that would allow users to do the following functions: 1) Access information from multiple data sources, 2) Analyze information, 3) Visually explore information, 4) Change or customize views according to context, and 5) Share visualizations and interact with teammates.

The architecture proposed by (Nascimento, Vivacqua, and Borges 2016) is illustrated in Figure 2 below. It consists of the following modules: "Extractor Module", "Collaboration Module", "Storage Module", "Presentation Module", "Selection Module", and "Interface Module". Excluding the "Interface Module", which runs on the Client, all other modules run on a Server. This architecture is a good basis but has some limitations that we address in the next section.

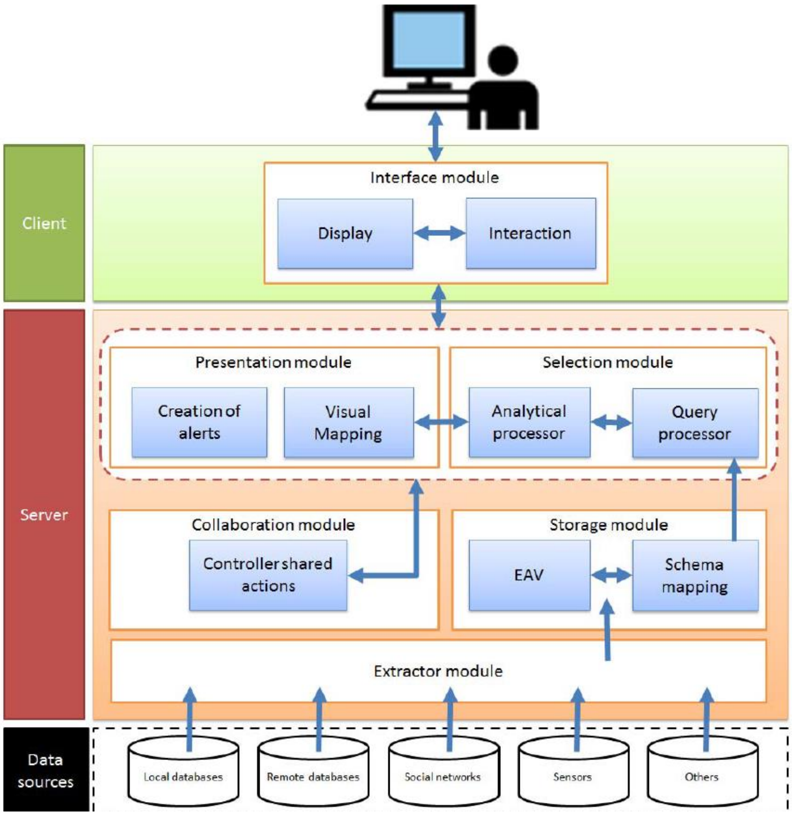

Figure 2. Emergency Dashboard Architecture, as proposed in (Nascimento, Vivacqua, and Borges 2016)

EMERGENCY DASHBOARD OPERATIONAL REQUIREMENTS

In this section we review the requirements for decision-supporting information visualization, which culminate in the decision to provide an emergency dashboard as the component of the solution that covers these requirements. The requirements were filtered out of a complete list of User Requirements, defined in beAWARE Deliverable D2.1 (beAWARE 2017). In addition to the original text of each requirement, we have drawn a visualization approach to match each requirement, comprising one or more ways to visualize and display the information to the decision-maker.

Table 1. Initial User Requirements for Decision-Supporting Visualization (beAWARE 2017)

ID	UR#	Requirement name	Requirement description
1.	UR_103	Flood warnings	Provide authorities with warnings on river levels overtopping some predefined alert thresholds, based on forecast results
2.	UR_112	Detect element at risk from reports	Provide authorities with the ability to detect the number of element at risk and the degree of emergency from mobile app and social media reports
3.	UR_118	River overtopping	Provide authorities/citizens with the ability to know if the river level is overtopping predefined alert thresholds
4.	UR_128	Risk Level Evaluation	Provide authorities with the ability to evaluate the forecasted level of risks (based on all the available dataset)
5.	UR_303	Forest-Fire Risk Assessment	Provide the authorities with a risk assessment regarding the probability of a forest fire to occur during or in the upcoming period after a heatwave. The relevant authorities will have an assessment of a fire risk based on the weather forecast during a heatwave and especially during the following days
6.	UR_306	Number of people affected	Provide the authorities an estimation of the people that might be affected from the heatwave and in which areas
7.	UR_313	First responders status	Provide to the authorities the current status and location of all first responders when they are performing their tasks
8.	UR_316	Capacity of relief places	Provide to the authorities the current state of the available capacity of all relief places provided to the public
9.	UR_320	Hospital availability	Show to the authorities the current availability of the hospitals.

EMERGENCY DASHBOARD ARCHITECTURE

Based on the end-user requirements and their analysis, and along the main insights drawn from the review of related literature, we determined the set of the functional requirements for the Emergency Dashboard system. The guidelines of a model-based requirements-driven architecting framework proposed in (Mordecai and Dori 2017), which used Object-Process Methodology, OPM (Dori 2016) as its underlying modeling language. This cyclic process consists of four phases:

- Recording stakeholder requirements
- Eliciting system requirements from stakeholder requirements
- Defining the system architecture
- Reviewing the outcome with the stakeholder and improving it

Stakeholder Requirements

Shown in the previous section, our end-user requirements constitute an initial set of stakeholder requirements. The next step is to form a list of well-defined system requirements based on the stakeholder requirements, when *the system* is defined as the Emergency Dashboard System (EDS).

System Requirements

A set of system requirements for the EDS has been constructed based on the Stakeholder Requirements. This set is shown in Table 2. Transforming stakeholder requirements into system requirements includes several aspects. First, we try to infer which system-level capabilities the EDS should exhibit to satisfy the stakeholder requirements. Second, system requirements should be as decoupled from each other as possible and lead to a modular solution. Third, system requirements should be reused and refined to cover as many user requirements as possible, so that, in general, the system requirements set would be as minimal and concise as possible. Finally, system requirements should be well-defined, well-phrased, and easy to understand by both the stakeholder, who reviews and approves them, and the system architect, who transforms them into features, functionalities, and functional requirements, and later allocates those to system components.

Table 2. Initial System Requirements for Emergency Dashboard System

UR#	SysReq ID	System Requirement
UR_103	EDS_01	EDS shall provide a User Interface for displaying metrics based on external measurement reports
UR_103	EDS_02	EDS shall provide an interface for external metric data providers to send their data to
UR_103	EDS_03	EDS shall provide a metric display showing a time-series within allowed warning thresholds
UR_103	EDS_04	EDS shall provide a way to bind a data stream to a metric display for presentation to a specific authority, role, or user
UR_103	EDS_05	EDS shall provide a way to associate a metric with a position on a map
UR_112	EDS_06	EDS shall provide a metric showing a "traffic light" with a number for each color category.
UR_118	EDS_07	EDS shall provide a metric display showing a warning or alert when predefined warning or alert thresholds are violated
UR_118	EDS_08	EDS shall provide a metric display showing a time-series within allowed warning thresholds
UR_128	EDS_09	EDS shall provide a metric display showing a given number relative to minimum and maximum values (gauge)
UR_303	EDS_10	EDS shall serve multiple authorities, in multiple geographies, with interests in multiple scenarios
UR_306	EDS_11	EDS shall provide means to display aggregate data based on available datasets (pivot charts)
UR_313	EDS_12	EDS shall provide means to breakdown aggregate metric displays into multiple category-divided metric displays
UR_316	EDS_13	EDS shall provide means to display the latest update available in a time-series as a separate metric display (e.g. progress bar)
UR_320	EDS_14	EDS shall provide access to metrics according to user pertinence to an authority, user role, and/or user identity.

System Functionality

Based on the system requirements we derived the main functionalities of the EDS. The top-level functionality of the system is **Displaying Decision-Supporting Functionality**. The main beneficiary of this functionality is the **Decision-Maker**, who uses the outcome of this functionality – the **Dashboard** – in the **Decision-Making** process. Additional agents are external data providers, who feed data stream messages into the system. Other external processes include **Managing System Definitions** (by an administrator or administrative superior system); external **data analytics services** that may be utilized for data analysis by the system (such as web-based format convertors, service algorithms, etc); and external applications' **map-based information display**, which complements the dashboard application and provides a detailed map with events and tactical layered information.

The top-level functionality consists of four *feature functionalities* and two *support functionalities*. Each functionality can be further decomposed into several sub-functionalities, functions, or objectives. These help clarify the roles of each functionality and ensure its relevance and necessity.

Feature functionalities are the functionalities that provide direct value:

- Receiving data streams
  - Receiving data stream messages through an API
  - Storing data stream messages' content in database
  - Informing the system on updates to the data stream
- Analyzing data sets
  - Running selected common analysis techniques on raw data streams according to data analysis definitions and storing the results as a new data stream
  - Running external analytic services on existing data streams, as defined in the data analysis definitions, obtaining the results
  - Sending analyzed data streams for storing
- Staging data sets for presentation
  - Marking data streams for presentation according to data display definitions
  - Checking data stream availability for presentation
  - Fetching latest available information for data stream to display
- Displaying the dashboard
  - Fetching staged data streams
  - Drawing the user interface
  - Drawing the dashboard frame
  - Deploying the metric displays according to the tenant, role, and user connected and according to the metric display definitions

Supporting functionalities are the functionalities that provide indirect and usually internal value:

- Routing data sets
  - Transferring notifications and pointers to data from originators to consumers
  - Tracking information flow in the system
- Managing data access
  - Providing user registration and access permissions for users
  - Responding to data access requests for display to specific users
  - Tracking user activity in the system

System Architecture

In order to facilitate a viable solution, the next step is to define a robust architecture that covers the system's functionalities. For each functionality we allocated a dedicated conceptual component. A conceptual component contains functions that can be implemented in various forms. The feature components are: Receiver, Analyzer, Stager, and UI. The support components are: Router and Authorizer.

All the components in our EDS are intended as software components. As a good practice, they should employ reusable technologies, software libraries, on-line services, micro-services, and open interfaces. In addition, the system is designed, developed, and incorporated using techniques that cover the relevant requirements and integrate them all into a working system, while reducing risk and assuring performance. However, this elaboration is beyond the scope of the present paper.

An Object-Process Diagram (OPD) of the architecture is shown in Figure 3. This OPD shows the following elements and aspects:

- the top-level functionality of the system, decomposed into the main functionalities, as listed above;
- external agents and processes that feed input into the EDS or consume/handle its output; and
- data objects that are provided as inputs to the system or one of its subsystems or as outputs of the system or its subsystems.

The ellipses represent the processes in the model – functionalities and services, while the rectangles represent the objects – users (human or machine), software components, or information objects. Processes and objects that are defined as external to the system are indicated by dashed contours. Physical or actual objects, such as humans, systems, or agents, are shaded, while informatical objects – data and information – are not. These are typical OPM language notations that help distinguish objects – things that exist or might exist, from processes – things that occur or might occur; internal things from external things, and physical things from informatical things. In addition, we have used white background for feature functionalities and components, and grey background for support functionalities and components. External beneficiaries are colored in deep purple while external functionalities and processes are colored in light purple. Data objects are colored in yellow.

The diagram also shows potentially-needed computation resources, such as databases, application servers, authentication server/service, and routing server/service, for each functionality. For example, the "**Receiving Data Streams**" service of the **Receiver** component requires **Database** and **Application Server**. These computation resources may eventually become sub-components of the conceptual components that own each functionality. These resources appear at the lower layer of the OPD. While some resources play a potential role in more than one functionality, in practice we may decide to deploy multiple, separate, and independent instances of these resources in order to optimize performance, cost, robustness, availability, or a combination thereof. However, such considerations are beyond the scope of the present paper.

The diagram was created with OPCAT 4.2 – OPM's free modeling and simulation software. As a major part of the OPM language, which is simultaneously graphical and textual, OPCAT also provides a machine-generated text in Object-Process Language (OPL) – OPM's textual modality. The text provides simple, readable explanation of each element and construct in the diagram, including the objects, processes, and links among them. The OPL textual specification of the diagram is available as Appendix A.

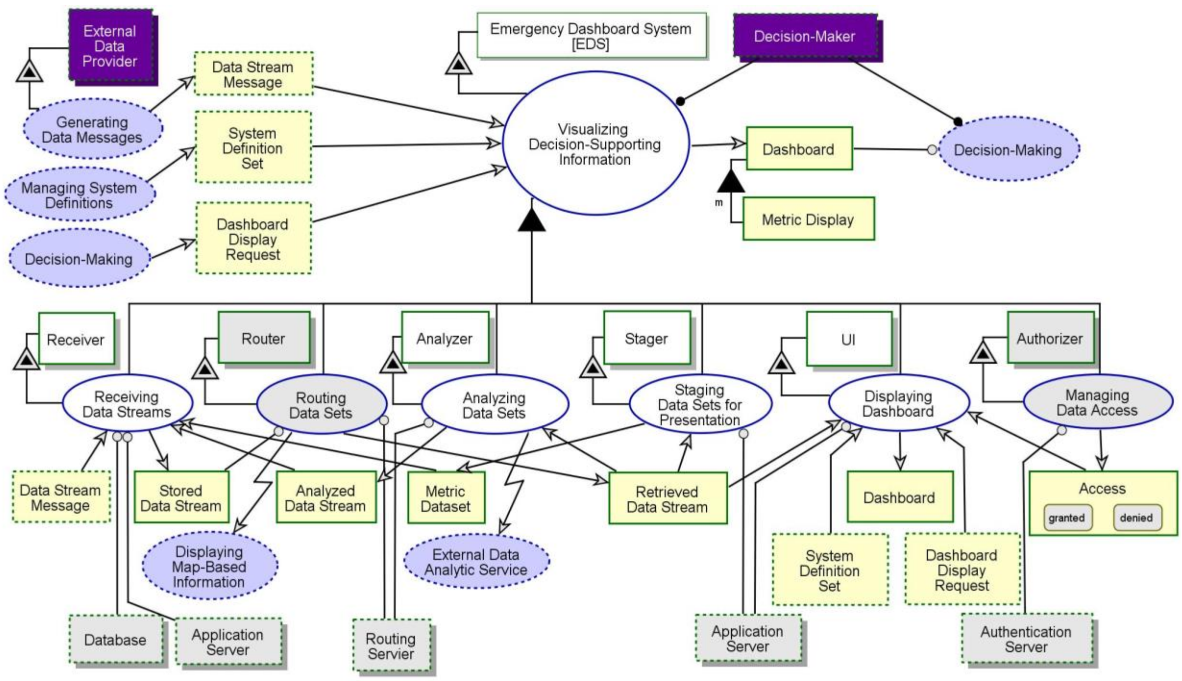

Figure 3. Proposed Emergency Dashboard System Functionality – an Object-Process Diagram

CONCLUSION

In this paper, we have proposed an Intelligent Utilization of Dashboards in Emergency Management, which carefully derives a robust functionality-supporting structure from a set of well-defined generically-oriented system requirements, based on but not bounded by or directly implementing an authentic set of end-user requirements. Rather than try to implement each end-user requirement for decision-supporting visualization directly and in a tailor-made fashion, our approach is to understand the underlying mechanisms needed to support not only the present requests and expectations of the end-users, but also their future needs and challenges with processing and using data for decision-making during emergency scenarios.

Our proposed functional decomposition and preliminary conceptual structure extends the functionality and scalability offered by previous frameworks. Our system requirements and functionality definition approach attempted to facilitate, apart from direct compliance with the immediate end-user requirements, that our solution would be robust, flexible, and applicable to the problem at hand – i.e. enhancing decision support in emergency management via information visualization, while considering the dynamics of the problem and the need to allow flexibility across the board – in data reception, in data processing and information extraction, and in data and information visualization to support evolving decision making processes.

In future research, we intend to elaborate this framework into a feasible architecture, implement and demonstrate a prototype solution, and study the solution's usability when coping with the genuine dynamics of emergency management. We intend to study the growth in operational needs on the one hand and the growth in delivered capability on the other hand, and propose ways to refine and improve the described solution to better support the effort to optimize the dashboard as a critical decision-support tool.

ACKNOWLEDGMENTS

This research was funded by the European Union Horizon 2020 Program under grant #700475 - beAWARE.

REFERENCES

Bakopoulos, Menelaos, Sofia Tsekeridou, Eri Giannaka, Zheng-Hua Tan, and Ramjee Prasad. 2011. "Command & Control: Information Merging, Selective Visualization and Decision Support for Emergency Handling." In *8th International ISCRAM Conference*, 1–10.

beAWARE. 2017. "D2.1 Use Cases and Initial User Requirements."

beAWARE. 2017. "beAWARE Newsletter."

Chen, Hsinchun, Roger H. L. Chiang, and Veda C. Storey. 2012. "Business Intelligence and Analytics: From Big Data to Big Impact." *MIS Quarterly* 36 (4): 1165–88.

Conforti, Raffaele, Giancarlo Fortino, Marcello La Rosa, and Arthur H M Ter Hofstede. 2011. "History-Aware, Real-Time Risk Detection in Business Processes." In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7044 LNCS:100–118. doi:10.1007/978-3-642-25109-2_8.

Desai, Yogeeta, Steven Jiang, and Lauren Davis. 2016. "Evaluation of Dashboard Interactivity for a Local Foodbank." *Proceedings of the Human Factors and Ergonomics Society*, 2032–35. doi:10.1177/1541931213601463.

Dori, D. 2016. *Model-Based Systems Engineering with OPM and SysML*. *Springer*. doi:10.1007/978-1-4939-3295-5.

Dusse, Flávio, Paulo Simões Júnior, Antonia Tamires Alves, Renato Novais, Vaninha Vieira, and Manoel Mendonça. 2016. "Information Visualization for Emergency Management: A Systematic Mapping Study." *Expert Systems with Applications* 45: 424–37. doi:10.1016/j.eswa.2015.10.007.

Kuwata, Y, Y Ishikawa, and H Ohtani. 2000. "An Architecture for Command and Control in Disaster Response Systems." *Iecon 2000: 26Th Annual Conference of the Ieee Industrial Electronics Society, Vols 1-4: 21St Century Technologies and Industrial Opportunities*, 120–25. doi:10.1109/IECON.2000.973136.

Mordecai, Y., and D. Dori. 2017. "Model-Based Requirements Engineering: Architecting for System Requirements with Stakeholders in Mind." In *2017 IEEE International Symposium on Systems Engineering, ISSE 2017 - Proceedings*. doi:10.1109/SysEng.2017.8088273.

Nascimento, Bruno S., Adriana S. Vivacqua, and Marcos R.S. Borges. 2016. "A Flexible Architecture for Selection and Visualization of Information in Emergency Situations." In *IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016*, 3317–22. Budapest, Hunngary: IEEE. doi:10.1109/SMC.2016.7844746.

Nazemi, Kawa, Dirk Burkhardt, David Hoppe, Mariam Nazemi, and Jörn Kohlhammer. 2015. "Web-Based Evaluation of Information Visualization." *Procedia Manufacturing* 3 (Ahfe). Elsevier B.V.: 5527–34. doi:10.1016/j.promfg.2015.07.718.

Novotny, M. 2004. "Visually Effective Information Visualization of Large Data." *8th Central European Seminar on Computer Graphics (CESCG 2004)*, 41–48.

Purdy, Grant. 2010. "ISO 31000:2009 - Setting a New Standard for Risk Management: Perspective." *Risk Analysis* 30 (6): 881–86. doi:10.1111/j.1539-6924.2010.01442.x.

Saaty, Thomas L. 2008. "Decision Making with the Analytic Hierarchy Process." *International Journal of Services Sciences* 1 (1): 83. doi:10.1504/IJSSCI.2008.017590.

Schlegelmilch, Jeffrey, and Joseph Albanese. 2014. "Applying Business Intelligence Innovations to Emergency Management." *Journal of Business Continuity & Emergency Planning* 8 (1): 31–41.

Shan, Siqing, Li Wang, Ling Li, and Yong Chen. 2012. "An Emergency Response Decision Support System Framework for Application in E-Government." *Information Technology and Management* 13 (4): 411–27. doi:10.1007/s10799-012-0130-0.

SRA. 2015. "SRA Glossary."

Surakitbanharn, Chittayong, and David S Ebert. 2017. "Improving the Communication of Emergency and Disaster Information Using Visual Analytics." In *Advances in Intelligent Systems and Computing 592: Advances in Human Factors and Systems Interaction: Proceedings of the AHFE 2017 International Conference on Human Factors and Systems Interaction*. Vol. 592. Los Angeles, CA, USA: Springer-Verlag London. doi:10.1007/978-3-319-60366-7.

APPENDIX: OBJECT-PROCESS LANGUAGE SPECIFICATION OF THE PROPOSED EDS ARCHITECTURE

1 Emergency Dashboard System [EDS] is physical.
2 Emergency Dashboard System [EDS] exhibits Visualizing Decision-Supporting Information.
Visualizing Decision-Supporting Information consists of Receiving Data Streams, Routing Data Sets,
3 Analyzing Data Sets, Staging Data Sets for Presentation, Displaying Dashboard, and Managing Data Access.
4 Receiving Data Streams requires Application Server and Database.
5 Receiving Data Streams consumes Metric Dataset, Analyzed Data Stream, and Data Stream Message.
6 Receiving Data Streams yields Stored Data Stream.
7 Routing Data Sets requires Stored Data Stream and Routing Servier.
8 Routing Data Sets yields Retrieved Data Stream.
9 Analyzing Data Sets requires Routing Servier.
10 Analyzing Data Sets consumes Retrieved Data Stream.
11 Analyzing Data Sets yields Analyzed Data Stream.
12 Analyzing Data Sets invokes External Data Analytic Service.
13 Staging Data Sets for Presentation requires Application Server.
14 Staging Data Sets for Presentation consumes Retrieved Data Stream.
15 Staging Data Sets for Presentation yields Metric Dataset.
16 Displaying Dashboard requires Application Server.
Displaying Dashboard consumes Access, Dashboard Display Request, System Definition Set, and Retrieved
17 Data Stream.
18 Displaying Dashboard yields Dashboard.
19 Managing Data Access requires Authentication Server.
20 Managing Data Access yields Access.
Visualizing Decision-Supporting Information consumes Dashboard Display Request, System Definition Set,
21 and Data Stream Message.
22 Visualizing Decision-Supporting Information yields Dashboard.
23 Data Stream Message is environmental.
24 Dashboard consists of many Metric Displays.
25 System Definition Set is environmental.
26 Dashboard Display Request is environmental.
27 Receiver is physical.
28 Receiver exhibits Receiving Data Streams.
29 Router is physical.
30 Router exhibits Routing Data Sets.
31 Analyzer is physical.
32 Analyzer exhibits Analyzing Data Sets.
33 Stager is physical.
34 Stager exhibits Staging Data Sets for Presentation.
35 UI is physical.
36 UI exhibits Displaying Dashboard.
37 Authorizer is physical.
38 Authorizer exhibits Managing Data Access.
39 Access can be granted or denied.
40 Decision-Maker is environmental and physical.
41 Decision-Maker handles Decision-Making and Visualizing Decision-Supporting Information.
42 External Data Provider is environmental and physical.
43 External Data Provider exhibits Generating Data Messages.
44 Generating Data Messages is environmental.
45 Generating Data Messages yields Data Stream Message.
46 Database is environmental and physical.
47 Application Server is environmental and physical.
48 Authentication Server is environmental and physical.
49 Routing Servier is environmental and physical.
50 Decision-Making is environmental.
51 Decision-Making requires Dashboard.
52 Decision-Making yields Dashboard Display Request.
53 External Data Analytic Service is environmental.
54 Managing System Definitions is environmental.
55 Managing System Definitions yields System Definition Set.