A Service-oriented Architecture for Emergency Management Systems

Stephan Braune,[*] Achim D. Brucker,[†] Gerald Kleser,[†] Keqin Li,[†] Theo Dirk Meijler,[†] Heiko Paulheim,[†] and Florian Probst[†]

[*] B2M Software AG, Germany
StephanBraune@gmx.de

[†] SAP Research, SAP AG, Germany
{achim.brucker, gerald.kleser, keqin.li, theo.dirk.meijler, heiko.paulheim, f.probst}@sap.com

Abstract: The complexity and openness of today's modern societies result in the threat of serious cascading effects when natural disasters or terrorist attacks strike. Thus, there is a high demand for state of the art IT support in the field of emergency management systems.

In this paper, we identify the core requirements of future emergency management systems and present a new generation of modular, service-oriented and semantic-web-based architecture for emergency management systems. Our emergency management system offers innovative functionality in the context of distributed information sources, collaborative work environments, and consistent situation pictures.

1 Introduction

The complexity and openness of today's modern societies result in the threat of serious cascading effects when natural disasters or terrorist attacks strike. Modern information technologies have changed everyday life fundamentally, yet in the field of emergency management, such technologies have not yet been brought to their full potential. This is especially the case for handling large incidents, e. g., natural or man-made disasters that exceed the capabilities of the individual emergency response organization. For supporting such situations, IT systems must, on the one hand, support the various relief organizations, police, fire brigades, as well as other emergency organizations and private organizations. On the other hand, such systems have to integrate, in a reliable, secure, and user friendly manner, a large number of both internal and external information sources and services that operate on these information sources.

In this paper, we present a detailed list of requirements that novel emergency management systems (EMSs) should fulfill (Section 2). In Section 3, we list the architectural decisions we take to fulfill the previously identified requirements and introduce the resulting service-oriented architecture. This architecture was developed during the research project *SoKNOS* (Service-oriented Architectures supporting public security networks) and realized as the *SoKNOS EMS*. The requirements for the SoKNOS EMS have been developed together with potential end users such as fire brigades and the police, who also participated in the evaluation of the system. Finally, we give a brief overview of related EMSs (Section 4) and close with some concluding remarks (Section 5).

2 Requirements for Emergency Management Systems

Large-scale emergencies and disasters require the operation of a command structure consisting of multiple emergency management teams, where each team manages the operations of one or more of the major organizations involved, such as fire departments, police, and rescue and technical relief organizations. An EMS will therefore need to support an emergency management team in:

- *Accessing Information*, i. e., finding, exchanging, and evaluating information about an emergency faster;
- *Planning*, i. e., scheduling operations in a clear-sighted and interactive manner;
- *Collaboration*, i. e., simplifying the cooperation between organizations and experts in the public security sector;
- *Decision making*, i. e., reducing the reaction time and increasing the reliability of the actions taken;
- *Situational insight*, i. e., providing a prompt and comprehensive overview of the disaster and the rescue operation status using user-friendly information technology.

Based on user studies conducted in the SoKNOS project together with end users, we have identified a set of essential requirements to achieve a new generation of user-centric, Web service-based EMSs:

R1: *Integrating new information sources during run-time.* A central requirement of EMSs is the ability to integrate new and heterogeneous information services from various providers, such as traffic and weather information, flood warnings, or the availability of resources of other emergency response organizations. This in turn requires that the user can easily find new information services and assess the syntactic and semantic interoperability with the information services that are already integrated.

R2: *Extending the system functionality during run-time.* The nature of large disasters is that full preparation against them is hardly possible. Emergency response plans will usually be insufficient. Hence, the possibility of quick adaptation to the new situation is important. With respect to the EMS, the possibility to extend the offered functionality of a system with new functional modules must be given.

R3: *Enabling collaborative work within and across organizations.* A well known deficit in modern emergency management is the insufficient possibility for organizations to jointly plan appropriate actions and measures without sending liaison officers to the corresponding organizations. Sound IT support for jointly developing a common situational picture and, based on this picture, developing strategies and plans for fighting the disaster is missing. Hence, an important requirement is that multiple team members are enabled to share the same information and work collaboratively on this information.

R4: *Ensuring information consistency.* Information consistency is paramount for establishing a common situation picture. Incidents develop quite fast. Consequently, in current systems, the operational pictures of collaborating organizations differ considerably. Furthermore, information might get lost and inconsistencies arise which might finally lead to wrong decisions. Therefore, functionality is needed which enables to track and eliminate information inconsistency within and across organizational borders.

R5: *Tracking objects across views.* Solutions to the previous requirement introduce the requirement of visualizing the same objects in parallel in spatial, temporal and organizational views. Physical objects like roads, buildings or rescue teams are involved in processes (temporal entities). This involvement can be passive, like a dam involved in a flood, or active, like a rescue team involved in rescuing or protecting. Tracking an individual entity in parallel across these different spatial, temporal and organizational views enables the user to get a holistic and highly realistic picture of the situation at hand.

R6: *Overcoming semantic heterogeneities.* Even when operating in the same region or country, organization often use slightly different vocabulary. The requirements regarding information exchange and collaborative work introduce the problem of semantic heterogeneities within jointly used information sources. Hence, a central requirement for a new generation of EMSs is that they provide methods to overcome the problems that arise when the same word is used with different meanings, or one concept is addressed with different words.

R7: *Managing system reliability.* A failure of system parts should have minimal consequences, i. e., a failure of the EMS for one emergency management team should not spill over to other EMSs. Within a one EMS, temporary network failure between workstations should not result in system break down. Further, the failure of a (local) system component should not lead to a failure of other system components or system functions.

R8: *Managing security dynamically.* Emergency management frequently deals with information that needs to be protected against unauthorized access. Thus, a central requirement is that high safety and security standards are fulfilled. However, in severe emergencies, safety and security standards that are useful for enterprise systems can drastically hinder efficient counter measures. Hence, the access control should support exceptional situations, i. e., the controlled overruling of access control decisions should be supported [BP09]. Furthermore, the communication channels between different EMSs need to ensure the integrity, confidentiality, and non-repudiation of the transferred messages.

R9: *User friendliness via personalization of system functionality and adjustment of level of detail.* Large emergencies are exceptional events, thus, familiarity with a specialized user interface which needs thorough training cannot be assumed. Furthermore, stress is a crucial factor in the management of large incidents. The EMS should therefore minimize the user's cognitive load. The EMS should effectively avoid the "threat rigidity syndrome" where additional stress is caused because of a loss of control over the situation or reduced understanding of reality [TCdWY04]. Hence, the EMS should consider the needs of the various users and the roles they have when engaged in handling the incident.

3 Future Emergency Management Systems

In this section, we briefly explain the most important design decisions that characterize the proposed EMS architecture and depict their relation to the requirements identified above, and we discuss the SoKNOS architecture and prototype.

3.1 Design Decisions

The design decision we are presenting can be classified into design decisions regarding a *single EMS* and decisions regarding the *collaboration between several EMSs*.

3.1.1 Architectural Decisions Regarding a Single EMS

D1 (R1, R2): *Vertical decoupling of components across front-end-, back-end-, and information source layers*. The encapsulation of the visualization and user interface (front-end layer), the data processing (back-end layer), and the access to various information and data sources (data source layer) allows several visualizations of the same back-end services.

D2 (R1, R2, R7): *Horizontal decoupling of components on front-end-, back-end- and information source layers.* The horizontal decoupling, i. e., modularization within each layer, allows for each organization to customize the EMS they use.

D3 (R2, R5): *Loosely coupled component collaboration through event-driven publish and subscribe.* Using an event-driven publish and subscribe service between the different components (both within one layer and across layers) enables the flexible extensibility of the EMS with new functionalities (e. g., new visualizations, new processing services, or additional information services). On the front-end layer, a reasoner implements semantic event processing between UI components and back-ends and serves as an indirection for ensuring loose coupling.

D4 (R1, R2): *Configuring and maintaining the functional components during run-time.* Supporting a flexible mechanism for configuring and maintaining the various components of an EMS at runtime allows for customizing the EMS for different usage scenarios (e. g., different tasks or different user preferences).

D5 (R1, R5): *Storing data relations between independent services separately.* Using a dedicated component for storing the data relations between the different services allows for integrating new data sources implemented as services without the need of modifying the (internal) representation of the service being integrated.

D6 (R4, R6): *Using semantic information as glue between components.* The continuous use of semantic annotations, both across layers and within each layer, allows for seamless interplay of decoupled components even though they have to cope with various heterogeneous types of data and functionalities.

D7 (R5, R9): *Establishing a single point of interaction.* Integrating all user interface components into a homogeneous user interface framework providing a single interaction point

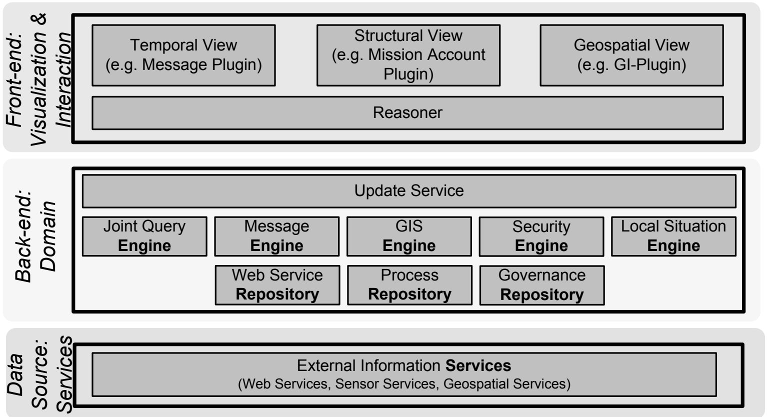

Figure 1: High-level view of the SoKNOS architecture, consisting of three decoupled layers: the data source layer, the back-end layer, and the front-end layer.

avoids the risk of distracting users by heterogeneous user interfaces and usage paradigms.

3.1.2 Architectural Decisions Regarding Collaboration between Several EMSs

D8 (R7): *Each emergency management team operates an individual independent EMS.* To ensure a high reliability even when faced with an unstable infrastructure, we assume an EMS is based on a distributed architecture (in contrast to a centralized one, e. g., using a central data storage for several EMSs). Thus, every EMS can operate (providing core functionality) in isolation, e. g., without network connectivity.

D9 (R3, R4, R9): *Facilitating cross-system information exchange.* To ensure the timely and efficient exchange of information and services between different EMSs, we apply the same principles across systems as we use for the collaboration between the components of one single EMS.

D10 (R8): *Using a decentralized identity and access control management.* On the one hand, EMSs handle sensitive data that needs to be controlled strictly by the organization operating a specific EMS. On the other hand, the efficient handling of an emergency requires the open and efficient exchange between different emergency management teams that might be operated by different organizations. A decentralized management of the security infrastructure (including, among others, the identity management and the access control configuration) enables the reliable, flexible, and secure handling of sensitive data.

3.2 Overview of the SoKNOS EMS Architecture

Following the design decisions outlined in the previous section results in a flexible, service-oriented architecture for EMSs, which we call the *SoKNOS EMS Architecture*. Figure 1 illustrates the SoKNOS EMS architecture, consisting of three decoupled layers, from bot-

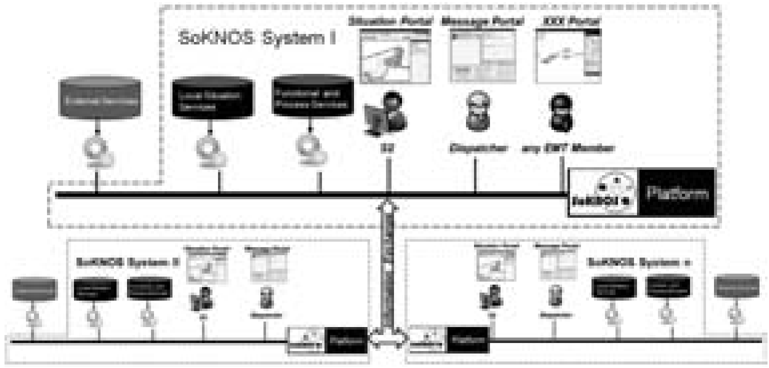

Figure 2: Each SoKNOS EMS is a distributed system; the components are connected via the SoKNOS platform. SoKNOS EMSs can be connected via the same platform functionality that is used to connect components of a single SoKNOS system.

tom to top: the data source layer, the back-end layer, and the front-end layer. In each of the layers, Figure 1 provides also a brief summary of the most important components for handling the different functional and non-functional requirements an EMS has to fulfill.

To our knowledge, the SoKNOS EMS Architecture is the first proposal for a distributed, highly dynamic EMS which makes heavy use of semantic web technologies. Figure 2 shows this important aspect: SoKNOS EMSs are themselves distributed systems, and multiple SoKNOS EMSs being used by different emergency management teams can collaborate on basis of the information exchange between these systems. This information exchange is based on the same platform approach that is used within each SoKNOS EMS.

3.3 Overview of the SoKNOS EMS Prototype

During the SoKNOS project, we have implemented a prototype that provides an EMS implementing the SoKNOS EMS Architecture. As an underlying service framework, also providing the functionalities required for the collaboration of several SoKNOS EMSs in a distributed manner, this prototype is based on Apache Tuscany [PKA$^+$10]. Our prototype integrates various internal and external services that allow, for example, the efficient information gathering, simulation and planning, and thus supports the work of a local emergency management team. Moreover, our prototype supports the efficient and secure collaboration between emergency management teams either using the SoKNOS EMS or working with other systems.

On the one hand, we are using this prototype for evaluating the SoKNOS approach in collaboration with the German fire brigades and police forces. On the other hand, this prototype is presented in a "Living Lab" at SAP Research, showing the benefits of modern IT in the public security in safety sector to a widespread audience.

4 Related work

Several commercial and open source IT systems exist in the area of emergency management. Most of them, e. g., DISMA [TUV], Sahana [CDSDS+07], and deNIS [DEN11], are mainly deployed in control centers. They are based on a centralized system architecture and thus require a reliable and fast infrastructure. While there are systems that are operated locally within one emergency management team, such as the IGNIS system [IGN06] used in the Berlin Fire Department, they do not support the efficient collaboration across several emergency management teams.

A lot of research is currently being performed for improving emergency management systems, e. g., in projects such as ALARM, LAGE, or MANET (see [Bun10] for a survey). They concentrate on various aspects, such as performing simulations, mobile, ad hoc, and sensor-based networks, and data aggregation and visualization [Und10].

To our knowledge, the SoKNOS EMS architecture is the first attempt to provide decentralized, semantic-web-based EMS that is designed with simplifying the communication and collaboration between several organizations in mind, especially that the SoKNOS EMS architecture focuses more on the flexible integration of different functionalities and data and processing resources.

5 Conclusion

Emergency management systems raise complex requirements that, at least in this combination, are seldom required by today's enterprise systems. However, the conclusions that can be drawn from designing emergency management systems are also valuable for future enterprise systems. For these future enterprise systems, we predict a similar combination of contrary requirements. This prediction is based on our observation that the modern enterprises need to participate in highly dynamic supply chains and, for example, exchange information even with their direct competition. Based on our experience in the emergency management domain, we conclude that service-oriented systems profit from the following fundamental concepts:

- *Decentralized systems:* Decentralized systems provide, compared with centralized systems, a higher reliability combined with a higher flexibility on choosing integrating local services. Moreover, decentralization simplifies the implementation of the strict data separation requirements [BH10].
- *Decoupling through publish-subscribe:* The loose coupling achieved by the publish-subscribe model allows for building flexible systems that, e. g., can quickly integrate new information sources or services.
- *Decoupling through ontologies:* Ontologies, as a machine-readable information representation, enable the effective collaboration across different organizations, as well as across independently developed components [PP10].

For the future, we see two main lines of work that build on top of each other: We want to further focus on the collaborative aspects of the SoKNOS EMS systems: how to share processes, how to share information, such that local data can be kept, but the relationship between such data across organizations is known. Secondly, and based on that, we want to use our knowledge to influence the common set of "core components" [The10] for supporting collaboration through the Internet, as we believe that emergency management is an especially challenging area, but which should still be seamlessly supported by such core components.

References

[BH10] Achim D. Brucker and Dieter Hutter. Information Flow in Disaster Management Systems. In *International Conference on Availability, Reliability and Security (ARES)*, pages 156–163. IEEE Computer Society, 2010.

[BP09] Achim D. Brucker and Helmut Petritsch. Extending Access Control Models with Break-glass. In Barbara Carminati and James Joshi, editors, *ACM symposium on access control models and technologies (SACMAT)*, pages 197–206. ACM Press, 2009.

[Bun10] Bundesministerium für Bildung und Forschung (BMBF). Research for Civil Security – Rescue and Protection of People. http://www.bmbf.de/pub/research_for_civil_security_rescue_protection_people.pdf, 2010. [Online; accessed 06-Feb-2011].

[CDSDS+07] Mifan Careem, Chamindra De Silva, Ravindra De Silva, Louiqa Raschid, and Sanjiva Weerawarana. Demonstration of Sahana: free and open source disaster management. In *Digital government research: bridging disciplines & domains*, pages 266–267. Digital Government Society of North America, 2007.

[DEN11] deNIS – deutsche Notfallvorsorge-Informationssystem. http://www.denis.bund.de/, 2011. [Online; accessed 06-Feb-2011].

[IGN06] Berliner Feuerwehr-Leitstelle. http://www.berliner-feuerwehr.de/leitstelle.html, 2006. [Online; accessed 06-Feb-2011].

[PKA+10] Apostolos Papageorgiou, Tronje Krop, Sebastian Ahlfeld, Stefan Schulte, Julian Eckert, and Ralf Steinmetz. Enhancing Availability with Self-Organization Extensions in a SOA Platform. In *Fifth International Conference onInternet and Web Applications and Services (ICIW)*, pages 161 –166, 2010.

[PP10] Heiko Paulheim and Florian Probst. Application Integration on the User Interface Level: an Ontology-Based Approach. *Data & Knowledge Engineering Journal*, 69(11):1103–1116, 2010.

[TCdWY04] Murray Turoff, Michael Chumer, Bartel Van de Walle, and Xiang Yao. The Design of Emergency Response Management Information Systems (ERMIS). *The Journal of Information Technology Theory and Application (JITTA)*, 5:1–35, 2004.

[The10] The European Future Internet Initiative (EFII). Position paper on the Core Platform needed to support the Future Internet Projects. http://www.future-internet.eu/fileadmin/news/EFII_Draft_Discussion_Paper-Architecture-May2010.pdf, 2010. [Online; accessed 06-Feb-2011].

[TUV] TUV Rheinland. Industrial Risks Assessment and Analysis – DISMA. http://www.tuv.com/ar/en/evaluacion_y_analisis_de_riesgos.html.

[Und10] Sarah Underwood. Improving disaster management. *Commun. ACM*, 53:18–20, February 2010.