International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 4 (2012) pp. 043-054

Domain Specific Service Oriented Reference Architecture Case: Distributed Disasters and Emergency Knowledge Management

Jari Leppäniemi

Tampere University of Technology - Pori,
P.O. Box 300, FIN-28101 Pori, Finland
jari.leppaniemi@tut.fi

***Abstract*: This paper introduces the design of a Domain-Specific Service-Oriented Reference Architecture (DS-SORA) for disasters and emergencies management (4EM). It also gives an overview of the features of the emergency management knowledge. Reference Architecture (RA) models and uses abstract elements in a domain and thus it is possible to have several RAs at different levels of the same domain. The main concern of our stakeholders is to have a reasonable viewpoint for defining the data and information needed for decision-making in emergency situations. The principles of defining the domain-specific reference architecture are presented here and we give a short discussion on the service model. Although the demand for interoperation between systems and organizations based on service-oriented systems is growing, the choice between the available technologies is not straightforward. However, the core concepts of the presented domain-specific reference architecture are highly universal and thus applicable in several domains.**

***Keywords*:** Service oriented, reference architecture, inter-operability, emergency management, domain-specific, service model, service bus.

I. Introduction

The consequences of major accidents, disasters, crises, and catastrophes are almost always global. Injury to people, as well as damage to infrastructure and property, also often affects other people than those living in the area of the accident. It is common that in disasters – natural or man-made - there are several kinds of participants involved in the management of the situation. Participants represent typically not only different authorities (rescue, firefighting, police work, health and emergency, safety, etc) but also the public. Referring to the EIIF Incubator Group Report [7], the major stakeholders of the emergency management ecosystem include: state-based emergency management (EM) agencies, non-governmental organizations (NGO), international coordination agents, ICT solution providers, EM and NGO professional and academic communities. The stakeholder list should be continued at least by adding the public (victims and their relatives), and the financial and insurance companies that are supposed to cover the damages. There are also people – all over the world – who want to help victims, be volunteers, give donations, hear the latest news, and receive information about the disaster.

A. Interoperability challenge

The types and amount of information needed to fulfill all the information needs in emergencies are increasing all the time and thus the demand for the interoperability of different information sources and decision support systems is also growing. The quality of situational awareness in a timely fashion is a major problem in many cases. The nature of information infrastructures and differences in the interoperability of these systems thus play a crucial role in information sharing. Unfortunately, the appropriate interoperability of the information systems (i.e. certain applications in the systems) of the different authorities is not obvious [20, 21 and 30]. The required information is scattered in databases, registers, and other data sources. As a consequence, there is a need to develop situational awareness through combining data retrieval and technologies. Even in smaller and localized accidents, information available from one source could be helpful in solving a problem in another context [25].

Information sources, i.e. systems, may be categorized as open or closed. A system may be defined and implemented as a private and secured system for use only by authorized members of a certain organization. Thus it can be categorized as a closed system. These kinds of systems are not interoperable as such. An information source may also be defined as closed if its interface(s) is/(are) implemented by using a proprietary or private technology instead of a standardized one [15]. These kinds of closed systems are thus technically limited and opening them up for interoperability may require a huge amount of reworking. However, the current trend is for new information systems to be developed and also for legacy information systems to be redeveloped so as to be more interoperable than ever. The high interoperability of information systems could be seen as a prerequisite for seamless communication between authorities and other parties. However, interoperability (and the supporting technologies e.g. web services) is a very large issue. Some of the related research and technical work on these topics include ORCHESTRA [29], Wide Information Network [34], the ATHENA Interoperability Framework [3], and IEC TC65/290/DC [14]. The technology diffusion for disaster management has been researched by Albarayak [1]. According to work done by the IEC TC65/290/DC technical committee, the interoperability of two or more devices could be described as shown in Figure 1 below. For the purpose of this research, the term "*devices*" has been replaced by "*application*".

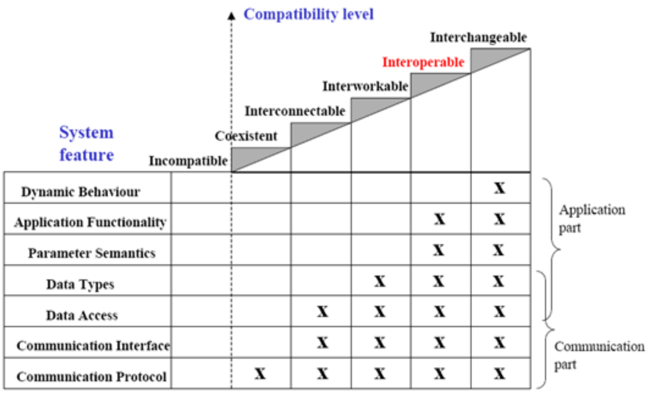

Figure 1. Interoperability defined as a level of compatibility (adapted from [3 and 14]).

Figure 1 shows that the technical challenge of interoperability with applications is layered. The fact is that there are both network-related and application logic-related issues to be solved when trying to obtain interoperability between diverse systems. Chen wrote in [6]: "The interoperability within or between enterprises is largely determined by their systems interoperability". In Wikipedia[1] the term *interoperability* is defined as: "a property referring to the ability of diverse systems and organizations to work together (inter-operate). The term is often used in a technical systems engineering sense, or alternatively in a broad sense, taking into account social, political, and organizational factors that impact system to system performance." Yet another definition is given by Merriam-Webster dictionary[2]: "Ability of a system (as a weapons system) to work with or use the parts or equipment of another system." It is worth to notice this definitions similarity to the idea of Service Oriented Architecture (SOA) [28]. Interoperability is now defined - not only as an ability to work together – but also as an ability to *use* the parts (i.e. services in SOA context) of another system. To summarize interoperability we first refer to the interoperability framework of ATHENA [3], which focuses on (i) interoperability at the enterprise/business level, (ii) interoperability of processes, (iii) interoperability of services, and (iv) interoperability of information and data. In Figure 2 the abstraction of layered interoperability services is applied to the emergency management domain.

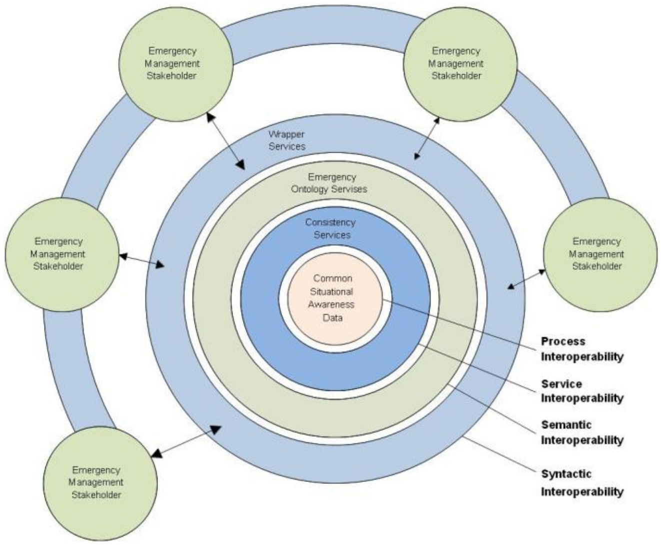

Figure 2. Interoperability services for Emergency Management Stakeholders.

Figure 2 describes an abstracted view to an interoperability approach which is planned to be achieved via introducing several sub-services. These sub-services are for example: wrapper services for syntactic level interoperability, ontology services for semantic level interoperability and consistency services for service level interoperability. Only after these services will the various stakeholders be able to have a common situational awareness of an incident. One example of a service based system meant to support medical information exchange in real time was described by Hauenstein & al. in [11]. Cataldo and Rinaldi gave another example of knowledge sharing using P2P approach and semantic web services [5].

B. Other challenges

Manoj and Baker studied [20] communication challenges in the response phase of the emergency management life cycle (see subsection 2) and concluded that in addition to these technical communication challenges there are also *organizational* challenges and *sociological* challenges. Organizational challenges are faced for example in post-disaster relief actions where groups accustomed to hierarchical decision making are forced to work in a flatter, more dynamic ad-hoc organization. Sociological challenges include for example: the availability, affordability, security, or applicability of the (current or to be developed) technology for emergency communication and/or better interoperability of systems. The lack of common vocabulary and shared semantics between response organizations and also between the organizations and public is also an example of a sociological-type challenge, which may be solved by icon languages or similar social science research [20]. Carver and Turoff in [4] gave an example of the complex environment and challenging context of the emergency management system and the emergency management process described in Figure 3.

[1] http://en.wikipedia.org/wiki/Interoperability
[2] http://www.merriam-webster.com/dictionary/interoperability

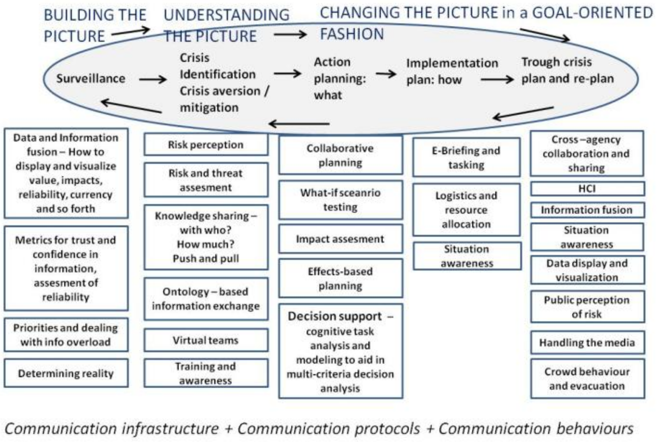

Figure 3. The Emergency management process. [4].

The detailed discussion by Carver and Turoff in [4] states - among other significant observations - that the design of emergency management information systems should be based on human – computer interaction. Grant [10] gives a proposal for a checklist when comparing EM systems and Feng and Lee discussed in [9] next- generation service-oriented emergency management systems.

The structure of this paper is as follows: The next chapter gives a brief introduction to our approach to the information needs related to the different phases of the emergency management lifecycle and the type of emergency functions at different levels of the management dimension. Then we present the basic service model used in the reference architecture and describe its packages. Finally, we state our topics for further research.

II. Emergency management lifecycle

The type of required information varies depending on the phase of the emergency management cycle, which is presented in Figure 4a [23] and [2]. The emergency management cycle divides the lifecycle of an emergency into four different phases shown in the middlemost ring. Before the event, when something bad happens (the impact), there is usually time to assess and plan things in the *prevention and mitigation* phase and sometimes start preparations for what is expected and might be coming in the *preparation* phase. The information systems, applications, and information needs in these phases are very plentiful. The wide spectrum of services and applications starts from measurement devices capable of giving automated alarms to huge anticipation applications using e.g. data mining and grid technologies in order to forecast for example mudflows or weather. The alerting function is also a very complicated communication task. A good discussion of an emergency alerting system is given by [32]. According to them, properties of a good alerting system are: support for second languages, spontaneity, automated operation, locality, ubiquity and non-intrusiveness [32].

An example of the monitoring of the performance of emergency call-taking system is described in [16]. They used a neural network (Self Organizing Map) based method to detect anomalies in emergency call data.

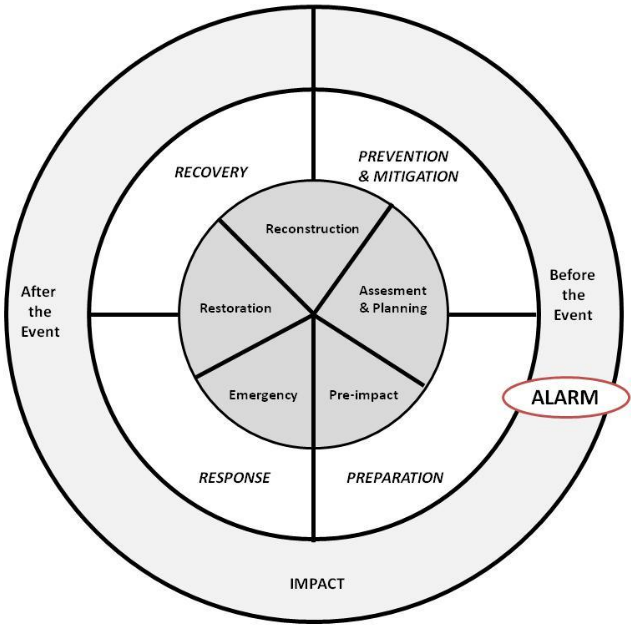

Figure 4a. The emergency management cycle (adapted from [23]; [2]).

The phases (Fig. 4a) after the impact are referred to as *response* and *recovery*. The main tasks during the response phase typically include at least the following: life saving and supply of vital goods and services, and the protection of the environment, assets, and property. During the recovery phase, the community returns back to its normal operation [2]. The spectrum of information systems and applications used in these lifecycle phases is also very wide. The same kind of variety is true for the timeliness of the information that is needed and available. For example, the need for real-time information when planning possible actions for emergency training in the prevention and mitigation phase is much smaller than when rescuing people in a hectic response phase [2].

The spectrum of information systems and applications used in these lifecycle phases is also very wide. The same kind of variety is true for the timeliness of the information needed and available. For example, the need for real-time information when planning possible actions for emergency education in prevention and mitigation phase is much smaller than rescuing people in a hectic response phase. An example of the design of a distributed system for combining real-time data with predefined information sources in order to anticipate possible crisis or emergencies (WIPER) has been described by Schoenharl & al. in [31].

The type of emergency task (i.e. rescue, police, health, etc.) and its function (strategic, operational or tactical) also places demands on the information required in emergency management. It is also important to differentiate between the types of information intended for authorities and the information directed to the public.

One classification of the emergency functions in different levels is presented in [23] (Figure 4b). According to their classification, emergency functions can be divided into three levels: strategic, operational, and tactical.

Figure 4b. Levels of emergency functions (adapted from [23]).

The strategic functions focus on the high level co-ordination of emergency management (e.g. financial provision, determination of strategy, public inquiries, environmental assessment and restoration). At the tactical level, the immediate functions and activities in the response phase are focused on e.g. search & rescue, firefighting, first aid and emergency treatment, casualty triage and also the longer term clean-up activities in the restoration / recovery phase (e.g. safety of site decontamination, restoration of scene). The operational level covers management and decision support functions and activities for co-ordination, prioritization, and supporting tactical level functions (e.g. coordination of response, hospital services, media information, investigations). Good quality and a sufficient quantity of timely, accurate, and reliable information is necessary for successful operations at this level.

In order to support the decision makers, good situational awareness is needed. One challenge is to identify the critical information sources and requirements in dynamic situations. Basically, there are three different approaches for this problem:

1. Model the processes and identify the information flows in advance (e.g. a response plan)
2. Model the processes and identify the information flows afterwards (e.g. accident investigation report)
3. Identify and utilize the critical information in real time (i.e. during the crisis situation, e.g. ad hoc information provided by the public)

The last approach could be partly handled by lightweight development processes utilizing what are known as web 2.0 technologies (mashup platforms, RESTful web services etc.). In [18 and 19] we used the first approach, illustrated by an example of the usage of a response plan. A visual example using Business Process Modeling Notation (BPMN) is given in Fig. 5.

Response plans are instructions for the Emergency Response Center (ERC). The plans are made beforehand by the other responsible authorities. Response plans describe which resources the ERC has to alert in a particular emergency case. A response plan is a composition of tactical and operational level activities. By means of modeling, we tried to emphasize the interfaces and information flows between different roles and organizations.

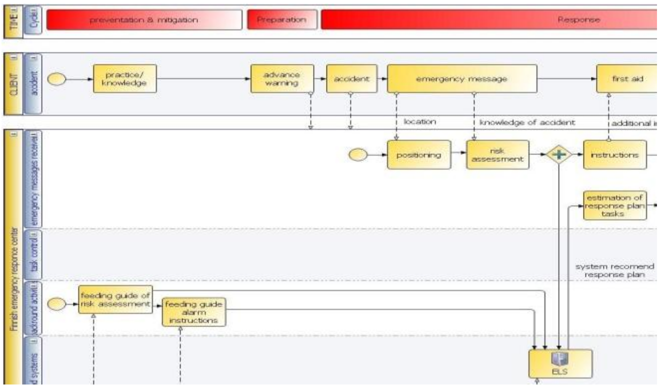

Figure 5. An example process diagram of several tasks related to Emergency Response Center response plans modeled using BPMN notation. [18].

In this example case (the response plan), we were able to model not only the tasks of the participating organizations but also to locate them in the corresponding phases of the emergency management cycle (red headings of the model). A more detailed description of the modeled case is given in [19].

In summary, the quality of timely situational awareness is a major problem in many cases. The nature of information infrastructures and differences in the interoperability of these systems thus play a crucial role in information sharing. Unfortunately, the appropriate interoperability of the information systems (i.e. certain applications in the systems) of the different authorities is not obvious.

III. Service Oriented Reference Architecture

According to OASIS [24] "A reference architecture models the abstract architectural elements in the domain independent of the technologies, protocols, and products that are used to implement the domain … It is possible to define reference architectures at many levels of detail or abstraction, and for many different purposes". This was the starting point of the work on reference architecture for emergency knowledge management. OASIS defines the key principles for an SOA ecosystem focused on large-scale distributed IT systems where the participants may be legally separate entities (as is the case in large scale emergency management, see Fig. 6):

- "SOA is a medium for the exchange of value between independently acting participants;
- participants (and stakeholders in general) have legitimate claims to ownership of resources that are made available via the SOA; and

- the behavior and performance of the participants are subject to rules of engagement which are captured in a series of policies and contracts [24]".

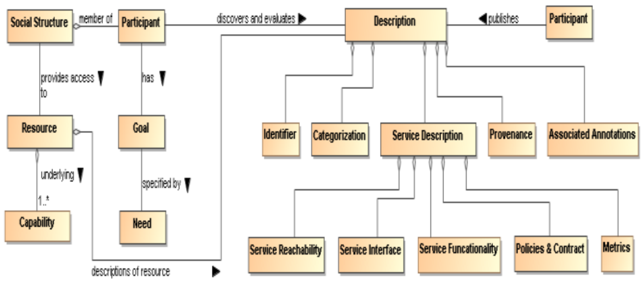

Figure 6. Business, Description and Willingness. [24].

The concepts and models like the OASIS service description (Fig. 7) could be utilized in our service design process.

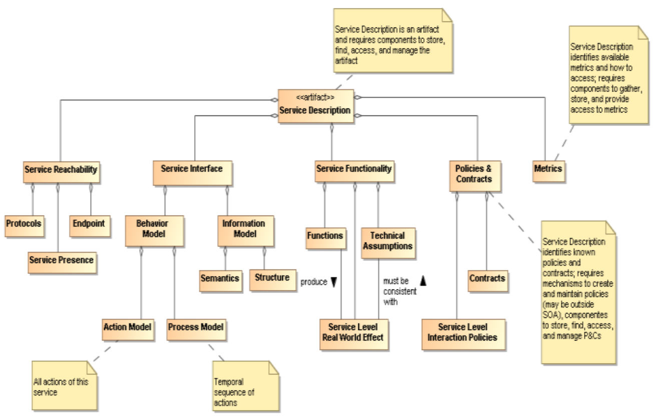

Figure 7. Service description. [24].

As our approach is to reuse - and extend when necessary - currently available service oriented architecture frameworks e.g. [28] and [24], we selected the OASIS SOA foundational framework for the basis of our situational awareness extension and for the derivative work for the service design guidance tools defined in the following chapters. Our focus is a service oriented (and enabled) analysis and decision making ecosystem for major disasters. Referring to the main concepts of the Conceptual description of Architecture (IEEE 1471-2000) [12], the main *concern* of our *stakeholders* is to have a reasonable *viewpoint* for defining the data and information needed for decision-making in emergency situations. Thus we defined the "Situational Awareness for Emergency Management (SA4EM)" viewpoint which is described in Table 1. In addition to the DS-SORA4EM viewpoint, we defined a "Situational Awareness" *view* that could comprise several *models*, which could be classified either as an information model (IM) or as a behavior model (BM) in figure 7, for example:

- data exchange model (BM)
- data classification model (IM)
- data availability model (BM)
- data transfer model (BM)
- decision data model (IM).

Each model is provided as UML diagram and/or ontology when available and needed. By ontology we understand an approach like [22]:

- Decide what to talk about.
- Decide on a vocabulary of predicates, functions, and constants.
- Encode general knowledge about the domain.
- Encode a description of the specific problem instance.
- Pose queries to the inference procedure and get answers.

Table 1. Viewpoint specifications for the OASIS Reference Architecture Foundation for SOA [24] added with emergency management specific extension (DS-SORA4EM) to the OASIS foundational work.

View-point Element	**OASIS View-point**	**OASIS View-point**	**OASIS View-point**	**DS-SORA4EM Viewpoint**
Name	*Service Ecosystem*	*Realizing Service Oriented Architecture*	*Owning Service Oriented Architecture*	*Situational Awareness for Emergency Management*
Main concepts	Captures what SOA means for people to participate in a service ecosystem.	Deals with the requirements for constructing an SOA.	Addresses issues involved in owning and managing an SOA.	Defines knowledge service design process for emergency management
Stake holder	People using SOA, Decision Makers, Enterprise Architects, Standards Architects, and Analysts.	Standards Architects, Enterprise Architects, Business Analysts, Decision Makers.	Service Providers, Service Consumers, Enterprise Architects, Decision Makers.	Business Analysts, Standards Architects, Service Providers, Service Consumers, Enterprise Architects, Decision Makers.
Concerns	Conduct business safely and effectively.	Effective construction of SOA-based systems.	Processes for engaging in an SOA are effective, equitable, and assured.	Assures that situation analysts and decision makers have enough information.
Modeling	UML class diagrams.	UML class, sequence, component, activity, communication, and composite structure diagrams.	UML class and communicatio n diagrams.	UML class, activity and communication diagrams. Business Process Modeling Notation (BPMN).

The OWL- standard (Web Ontology Language) describes ontology as: "An ontology formally defines a common set of terms that are used to describe and represent a domain. Ontologies can be used by automated tools to power advanced services such as more accurate web search, intelligent software agents and knowledge management." In later (section 4) is mentioned a "light weight" candidate as ontology for emergencies [17]. It could be positioned to the point of the red arrow in Figure 8 which is a typology provided by Uschold and Gruninger [in 17].

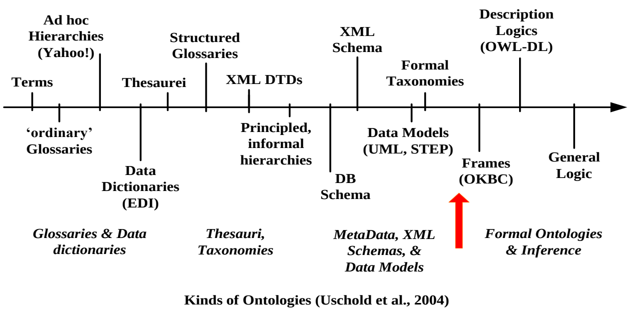

Figure 8. Kind of Ontologies. [17].

The formality of an ”ontology” grows to right. On the far left are different lists of concepts and terms and on the rightmost are concept systems allowing logical inference.

A. Decision data service design process

The emergency management knowledge service design process comprises several steps. These may include both domain-specific conceptual thinking and the evaluation of technical alternatives. An example of the upper level knowledge management service design process is given below: ***For each task*** involving a decision, the main *concern* of a *decision maker* or *analyst* (the stakeholders) is to ***discover***:

1. What data, information and knowledge are most essential for this situation and is it possible to support the analysts and decision maker merely by presenting that information?

2. Is it possible to enhance and/or accelerate the decision-making process by filtering, composing, or otherwise preprocessing the data before showing it to the analysts and decision makers at the tactical, operational, or strategic level?

The decision tasks explored are usually identified based on current practices, plans, or process models. However, it is extremely important that the processes are analyzed from the interoperability viewpoint rather than sub-optimal solutions based on tight organizational limits. In other words, the information flows must not be chained up by tight organizational governance rules and resources. As a consequence, the best solution may need several iterations before the desired interoperability can be achieved. The modeling is also affected by the technology solutions that are available and affordable.

The service model used in this presentation is loosely based on ERL's [8] division of different service types into *task, entity, and utility* services. An example of the domain-specific application of that service model could be an emergency situation described in the next figure (Figure 9).

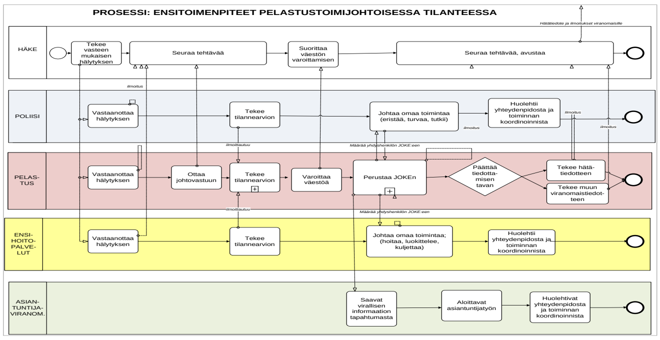

Figure 9. Tasks of Emergency management processes divided to different layers. [26].

In this simplified service model (or service layering) example based on the processes modeled in Figure 9, the first tasks of a rescue process are modeled in the swim lanes. The tasks of the emergency response function could be modeled as a *task*-centric service in the red swim lane that coordinates the emergency management in close co-operation with other *task*-centric services (ambulance services, police force and other authorities (yellow, blue and grey swim lanes). A service categorized as an *application* service might be performed by the emergency response center, which carries out the alerting and resource reservation services. Both the *application* and *task* oriented services might need *entity* services which are typically data and information retrieval from and updates to their own or partner's databases.

Positioning of the different parties and services to certain service types and/or classes depends on several factors. For example, those functionalities performed by the emergency response center and rescue services that are connected to the overall management of emergency operations could be abstracted and modeled as a separate control process and named as a layer of the orchestrated emergency management task services. Of cause it is possible and also more realistic to use even detailed service model. For example, another reference type service model is developed by CBDi and it is always possible to develop a customized solution.

IV. Service oriented reference architecture for emergency management

The next figures (10 and 11) present an example of a possible structure of the applications for an emergency management knowledge ecosystem. It is modeled as a traditional MVC (model – view – controller) architecture. Thus the main components of an application could be located in the MVC layers as follows: According to Figures 10 and 11, we can position the “view” in the user plane, the task-centric “control” in the process control and service mediation plane implemented by a service bus, and finally the “model”, which is a composition of different types of entity and utility services and legacy systems, is positioned in the infrastructure plane in order to achieve the interoperability between different stakeholders described in Figure 11.

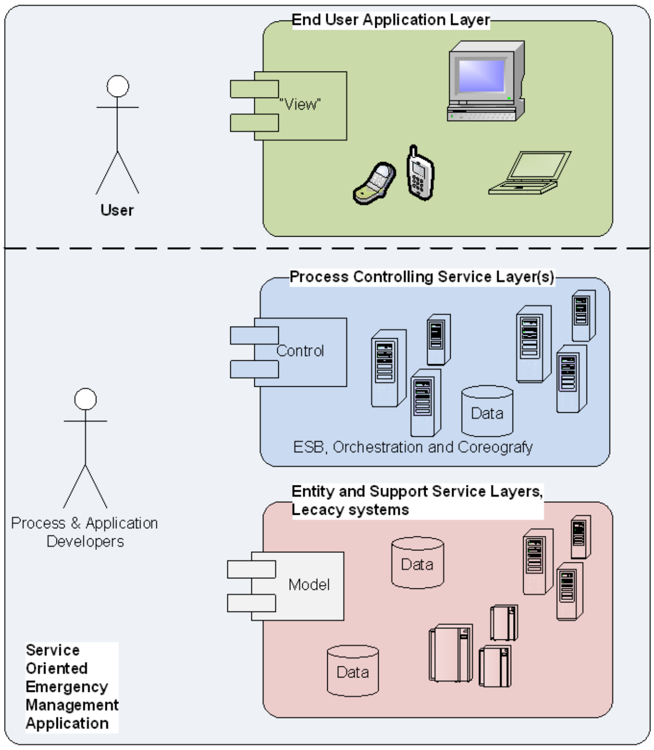

Figure 10. Service Oriented Emergency Management Application and MVC model.

In Figure 11, different users communicate via applications implemented according to the principles in Figure 10. The coordination of the shared (and modeled) business processes is based on the special control and service mediation layer. For example, AAA services (Authorization, Authentication and Accounting), and also monitoring, mediation, routing and data translations could be implemented by this layer as well other crucial features like fault tolerance, load balancing, replication, and logging, as described in Figure 12.

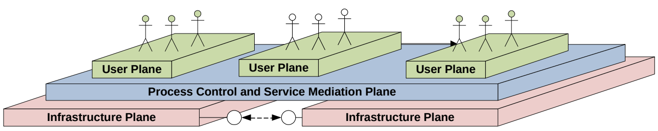

Figure 11. Shared business process supported by service bus and process control engine.

In Figure 12 some of the services are especially designed for supporting the modeled emergency management processes, in other words controlling the performance of the use cases, data and message translations, and routing messages to correct endpoints. A significant percentage of the services are for meeting non-functional requirements (load balancing, fault tolerance, replication, data security etc.). They are implemented as utility services.

As mentioned above, the emergency management knowledge ecosystem consists of different types of stakeholders. Some of them work in close relationship / interaction daily and have established ways of working and shared concepts.

Some of the stakeholders are interested in emergency information only at certain times or when certain conditions and limits are met and then only temporarily and/or in an extreme hurry. For this reason, it should be possible to offer open interfaces to stakeholders and organizations that do not take part / operate daily in emergency management. The same applies to the wider public and third parties who might want to implement "ad-hoc" type mash-ups for information sharing and independent communication.

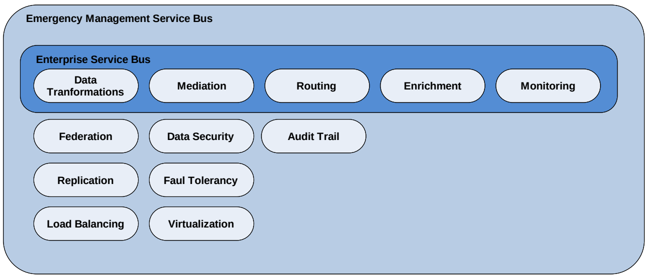

Figure 12. Services of an emergency management service bus.

Figure 13 presents an overview of the system architecture for a service-based emergency management knowledge ecosystem. The main interfaces are EMI-A and EMI-P: an authorized interface for emergency management services and a public interface for the general public and non-authorized users of the system(s). The concept and function of a service bus is crucial for this ecosystem so its functionality should be fault-tolerant and distributable.

The legacy systems (authorities, research organizations etc.) located at the bottom of Figure 13 interoperate with a large service-based architecture via open standards. They can preserve their proprietary interfaces and they can open new standard interfaces implemented by means of an appropriate technology (WS-*, RESTful, http, etc.).

Sporadic and temporary users can be connected as an information service provider or to a normal user via a public and open service interface like EMI-P in the Figure 13. A more permanent connection to the ecosystem is established using either the common emergency service bus (EMI-P) or the more controlled interface for authorized users (EMI-A), which supports the coordination and supervision of shared business processes.

The concrete implementations of domain-specific software architectures will be based on the concepts and knowledge required, such as ontologies and data models, by a domain (in this case emergency management), for a shared mission (goals). They are also based on the co-operative work to strive toward that goal (described by use cases and abstracted business processes), the frameworks and design patterns for interoperability and non-functional requirements so that the desired functionality (use cases, orchestrated processes) could be implemented (using SW technologies) via different types of services (task/entity/utility services), which are described and defined by service contracts according to the goals, rationales and constraints of the ecosystem.

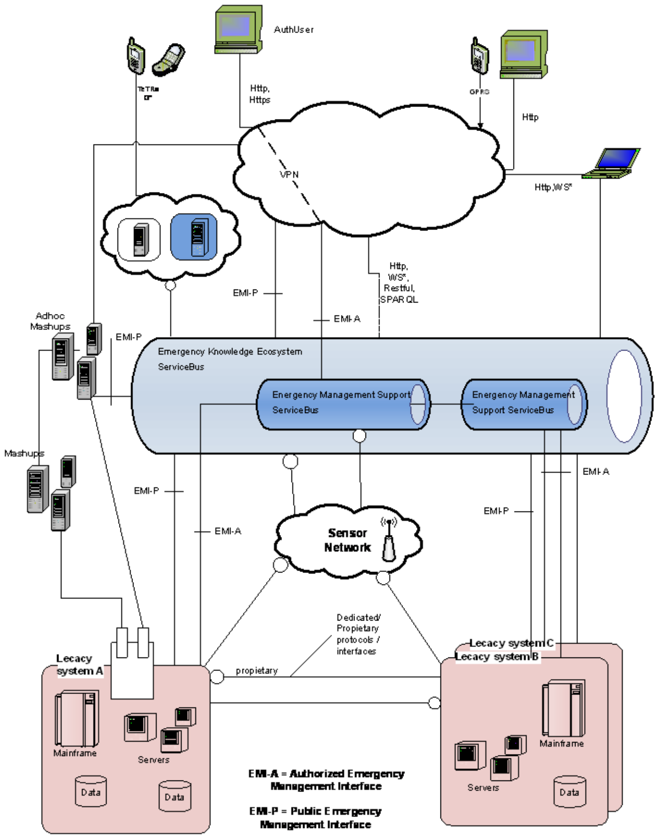

Figure 13. Service oriented system architecture for Emergency Management Knowledge Ecosystem.

Figure 14 consists of the main components of a domain-specific service-oriented reference architecture for emergency management. The main architectural layers, main packages and the possible dependences between them are presented using UML notation. All packages can aggregate sub-packages and services can be composed of other services. As explained in section 3, the service model used in this presentation is loosely based on ERL's [8] division of different service types into *task*, *entity*, *and utility* services.

In the following sub-sections the packages and layers are described in brief. The uppermost layer is the "Orchestration Layer," which basically comprises abstracted process packages that accumulate all the necessary knowledge of a domain-specific application of the reference architecture.

A. Orchestrated emergency management processes

Carefully selected, modeled parts of the emergency management processes (e.g. Figures 5 and 9) are abstracted for orchestration (and automation), and the supervision and monitoring of the processes are implemented by a distributable enterprise service bus (ESB). The functionality of the ESB could (and should) be distributed.

These orchestrated process packages have a dependency on Goals, rationale and constraints, because for example the constraints (resources, legacy systems, etc.) may put serious limits on the possibility of realizing the goals. Orchestrated process packages have a dependency on the Use Cases packages because existing functionality might affect the orchestration alternatives.

The Orchestrated processes package sets a dependency for the Use Cases package, because only the orchestrated parts of the shared business processes are implemented and supported by the use cases of the system.

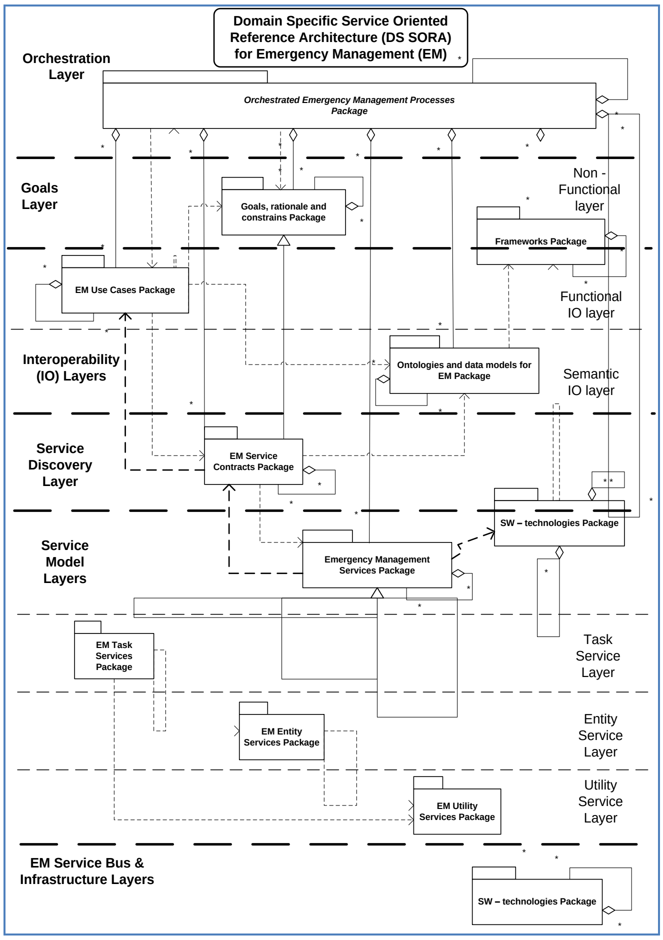

Figure 14. Layers, packages, and dependencies of a domain-specific reference architecture for co-operation by independent stakeholders within an emergency management ecosystem.

B. Goals, rationale and constraints

The goals, reasons and rationale for interoperability and the constraints are located in the Goals layer. It is also possible to set organizational values for these abstract components. The packages in this layer comprise the functional and non-functional requirements that in effect articulate the goals and rationales within the constraints. The requirements are for both the architecture and the business process to be supported by the architecture. These packages can be grouped in many ways: for example using the concepts of architectural standards [12]; or according to the architectural viewpoints, stakeholders, or concerns. The packages are realized via service contracts (and ultimately services) but the Frameworks packages (described below) control the enunciation of the requirements and these packages have a dependency on the Use Cases and the Orchestrated process. Realistic goals are dependent on the use cases of possible legacy systems and on the constraints on the operational environments of the shared business processes.

C. Frameworks and design patterns

These packages are located in two layers:

- In the Goals layer (the non-functional layer)
- In the upper Interoperability (IO) layer (the functional IO layer).

These packages are composed of the selected interoperability framework, design patterns and architecture principles and they are implicitly dependent on the “Ontologies and data models.” On the other hand they are heavily dependent on the “SW technologies” packages, because their maturity determines what kind of semantic problems could be solved. Some examples of possible sources of these packages are: ATHENA and EIF frameworks for interoperability [3], IEC TC65/290/DC work (Figure 1) [14], Orchestra architecture [29], OASIS service oriented reference architecture foundation and reference model [24 and 27], see for example the concepts used in Figures 6 and 7.

D. Use cases

The packages of use cases (based on e.g. Figures 5 and 9) are located in the functional interoperability layer. They are an essential part of the articulation of the requirements and they are also an essential part of the logic for the interoperable processes when they are orchestrated for control and monitoring. Use cases have a dependency on the Goals, rationale and constraint package via the Service contracts package and naturally they have dependencies on both the Orchestrated business processes and the Ontologies and data models packages.

E. Ontologies and data models

These packages are located in the semantic interoperability layer and are needed for interoperability between stakeholders. The selected and/or developed ontologies and data models are based on the principles derived from the Frameworks package. These packages set dependencies for the Use cases and Service contracts. An example of a candidate for an ontology for emergencies is described by Kruchten et al. [17] and part of it is shown in the figure 15 below.

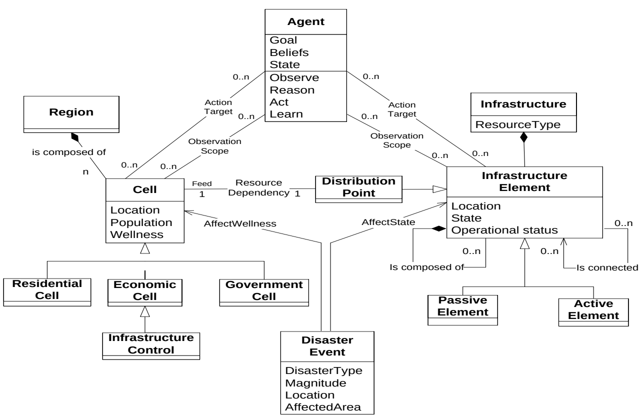

Figure 15. Detailed domain ontology for emergencies. [17].

On the other hand, an open source “tactical situation object” (TSO) [23] may be used as a data model for the same purpose. Figure 16 presents an overview of the TSO data model.

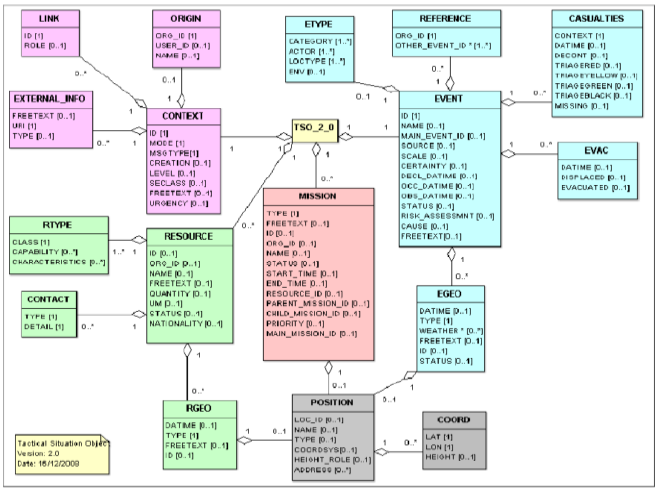

Figure 16. Tactical situation object model. [23].

The data model of the “Tactical situation object” defines concepts that are meant for determining a situational picture of an incident. An editor implemented as an open source project is also available.

F. Service contracts

Service contracts are placed in the “Service Discovery Layer.” This is a separate layer in the reference architecture because it strives to enhance the dynamic properties of service usage such as discoverability, reusability, and formulation of temporary service composition and orchestrations. The features of services are expressed precisely in these packages and in such a way as to satisfy the feasible constraints and concerns of the stakeholders as far as possible. Service contracts set a strong dependency to services, or in other words, the main principle should be: “Contracts First!” This is described by the thick line in the dependency arrow. It is clear that there are situations when the main principle cannot be followed; e.g. the functionality of legacy systems may be wrapped only according to the nature and limitations set by the legacy software. The narrow line is used to describe such a situation. The latter situation escalates (or migrates) to Use cases, too. The existing services very probably affect the overall design of the Use cases, and thus, for example, the service composition might be realized in many alternative ways. The discoverability of the services, understandable functionality and also the functional and non-functional limits of the services provided are based on the service contracts and thus they are essential for realizing the reference architecture. Due to the reasons above, the specification and design of the service descriptions and constraints, service directories, service listings and service categories should be a focal point of interest before starting to deploy a domain-specific service-oriented architecture

G. Services

The real services are positioned based on their types in corresponding layers. Perhaps the simplest but still useful layering (task/entity/utility layers) was shown in Figure 4.5, but the adapted principles of service formulation, capability of the service bus, limitations set by the legacy systems and the non-functional requirements of the service compositions also affect the layering principles of the service model. The formulation and positioning of services should always aim at the service contracts that allow the independent implementation of services. The functionality provided by a service bus is located by design at the dedicated Service bus and infrastructure layer(s) in order to enhance the specific technical implication in the reference architecture.

H. Software technologies

These packages are an essential source, not only for implementation but also for specification and design work. Which standards and SW technologies (e.g. WS-*, W3C, OASIS, IETF) [13, 33 and 35] should be chosen is dependent upon the services to be realized and, in addition, the technologies place a dependency to the Frameworks. In other words, the implementation possibilities of the derivate works of Frameworks are based on the availability and maturity of the SW technologies.

V. Conclusion and further research

A reference architecture differs from a reference model. A reference model describes the concepts and relationships in the domain. To understand the challenges in composing useful reference architecture, Fig. 17 addresses the variety of information and knowledge to be evaluated and accommodated.

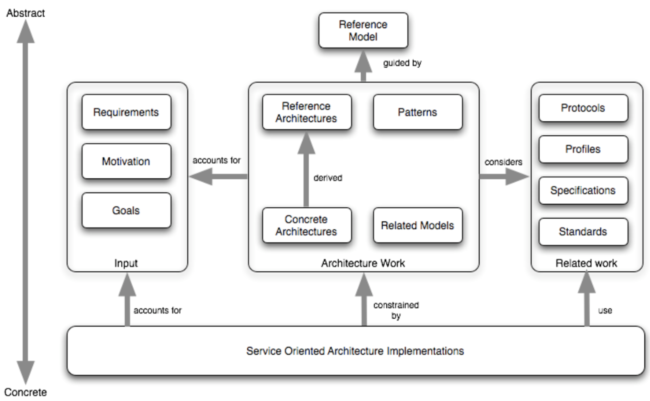

Figure 17. Artifacts and other information in relation to architecture work. [24].

As one can deduce from Fig. 17, the amount of useful and necessary information is massive. For example, there are over 250 drafts, specifications, standards or recommendations for defining and implementing web services [35]. A considerable number of them compete with or are alternatives to each other. This situation is probably a consequence of the competition for market share. As a result, the utilization of reference architecture material (including the references) may be a laborious task. An additional challenge comes from the dynamic nature of the specifications, standards, technologies, knowledge, and practices. In order to protect the investments made in reference architecture description work, it is also necessary to develop and apply other methods rather than simply putting a pile of paper or separate documents into digital form.

During the development of this reference architecture it became clear that the type and positioning of services is a key task when trying to realize the goals set for interoperability between different business operations and to surmount the legacy constraints. For example, it may be impossible to achieve the full potential of the architecture if the business processes of the different stakeholders are not examined in order to achieve an optimized solution based on the interoperability between the different parties. The formulation and positioning of the services based on the goals, rationale and constraints of interoperable business processes is the next focal point of our research. Our intention is to use models on different granularity levels to describe the desired viewpoints of our stakeholders.

VI. Summary

Sharing information and communication in large-scale emergencies is a major challenge [20, 21 and 30]. The necessary data and information are usually distributed and owned by a large number of organizations. The decisions made should be based on the combination of several types of knowledge. As a consequence, there is a need to develop situational awareness by combining data retrieval and technologies. The high interoperability of information systems could be seen as a prerequisite for seamless communication between authorities and other parties. Semantic and service level consistency and seamless interoperability between different parties is necessary when there are several stakeholders participating in an emergency situation.

The emergency management cycle divides the lifecycle of an emergency into four different phases: Prevention and Mitigation, Preparedness, Response, and Recovery. The type of information required varies, depending on the phase of the emergency management cycle. According to the EIIF Incubator Group Report, some of the major stakeholders of the emergency management ecosystem include: state-based emergency management (EM) agencies, non-governmental organizations (NGO), international coordination agents, ICT solution providers, EM and NGO professional and academic communities. The type of emergency task (i.e. rescue, police, health, etc.) and its function (strategic, operational, or tactical) also places demands on the information required in emergency management. It is also important to differentiate between the types of information intended for authorities and the information directed to the public. One challenge is to identify the critical information sources and requirements in dynamic situations.

In addition to our "Situational Awareness for Emergency Management" viewpoint, which indeed defines a knowledge service design process, aiming to "ensure that situation analysts and decision makers have enough information", we have also defined a "Situational Awareness" view that usually comprises several models, such as: a data exchange model, data classification model, data availability model, data transfer model, and decision data model. The technology for implementing service-oriented ecosystems is plentiful. However, there are more challenges than simply agreeing on interfaces and technology. After modeling and analyzing the shared emergency management processes, the division of work possibly in a new way, agreeing which data and whose computing resources are used, and finally reformulating and accepting the new targets, goals and responsibilities to stakeholders are at least equally important as the technology choices. For this reason, the formulation and development of services is a key task when realizing a service-oriented emergency management knowledge ecosystem.

Acknowledgment

The author thanks the SSMC/DDKM project team members and Finnish authorities for their valuable support to this research and the Finnish Funding Agency for Technology and Innovation (TEKES) for partial funding of this work.

References

[1] O. Albayrak. "Management and Diffusion of Technology for Disaster Management". In *PICMET Proceedings*, Istanbul, Turkey, pp. 1742–1748, 2006.

[2] D. Alexander. *Principles of Emergency Planning and Management*, US: Terra Publishing, Oxford University Press, 340 p.

[3] ATHENA Interoperability Framework, retrieved 29/02/11, http://www.modelbased.net/mdi/index.html and http://interop-vlab.eu/ei_public_deliverables/athena-deliverables.

[4] L. Carver and M. Turoff. "Human Computer Interaction: The Human and Computers as a Team in Emergency Management Information Systems", *Communications of ACM* 50 (3), pp. 33–38, 2007.

[5] A. Cataldo and A. Rinaldi. "Sharing Ontology in Complex Scenario using a Peer-To-Peer Approach", *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)*, Vol. 1, pp. 92– 109, 2009.

[6] P. Chen. "Architecture-Based Interoperability Evaluation in Evolutions of Networked Enterprises," C. Bussler & al. (ed.), *BPM 2005 Workshops*, LNCS 3812, Springer-Verlag, Berlin, Heidelberg, pp. 293–304, 2006.

[7] EIIF(W3C) Incubator Group Report 8/ 2009, Retrieved 29/02/11, http://www.w3.org/2005/Incubator/eiif/XGR-EIIF-20090806/

[8] T. Erl. *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall PTR, 760 p, 2005.

[9] Y-H. Feng and C.J. Lee. "Exploring Development of Service-Oriented Architecture for Next Generation Emergency Management System". *IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*, pp. 557–561, 2010.

[10] T.J. Grant. "A Checklist for Comparing Emergency Management Information Systems". In *Proceedings of the 5th International Conference in Information Systems for Crisis Response And Management*, Washington DC, USA, pp. 752–763, 2008.

[11] L. Hauenstein, T. Gao, TW. Sze, D. Crawford, A. Alm and D. White. "A Cross-Functional Service-Oriented Architecture to Support Real-Time Information Exchange in Emergency Medical Response". In *Proceedings of the 28th IEEE EMBS Annual International Conference New York City*, USA, 3, 2006.

[12] IEEE Std. 1471–2000, Recommended Practice for Architectural Description of Software-Intensive Systems, retrieved 09/02/11, http://www.iso-architecture.org/ieee-1471/.

[13] IETF, the Internet Engineering Task Force, retrieved 14/02/11, http://www.ietf.org/.

[14] ISO IEC, retrieved 03/02/11, http://www.iso.org/iso/home.htm.

[15] H. Jaakkola, J. Soini and J. Leppäniemi. "Seamless Solutions and Mobile Connectivity – Towards location independent and flexible system architectures". In *Publication 12, pp. 1–17, Tampere University Press*, 2008.

[16] P. Klement and V. Snášel. "SOM Neural Network – a Piece of Intelligence in Disasters Management", *International Journal of Computer Information Systems and Industrial Management Applications*, Vol. 2, pp. 243–251, 2010.

[17] P. Kruchten, C. Woo, K. Monu and M. Sotoodeh. "A human centered conceptual model of disasters affecting critical infrastructures". In *Proceedings of the 4th International Conference on Information Systems for Crisis Response and Management*, Delft, the Netherlands, pp. 327–344, 2007. They refer to: M. Uschold and M. Gruninger "Ontologies and semantics for seamless connectivity", *Journal of SIGMOD* 33(4), pp. 58–64, 2004.

[18] P. Linna. "Requirements engineering of Emergency Center's response plan (in Finnish: Hätäkeskuksen vastesuunnitelman vaatimusmäärittely*)". Master of Science thesis*, Pori, Tampere University of Technology, 2009.

[19] P. Linna, J. Leppäniemi, J. Soini and H. Jaakkola. "Harmonizing Emergency Management Knowledge Representation". *In PICMET 2009 Proceedings*, Portland, Oregon USA, pp. 1047–1051.

[20] B.S. Manoj and A.H. Baker. "Communication challenges in emergency response", *Communications of ACM* Vol. 50 (3), pp. 51–53, 2007.

[21] A. Meissner, T. Luckenbach, T. Risse, T. Kirste and H. Kirchner. "Design Challenges for an Integrated Disaster Management Communication and Information System". *The First IEEE Workshop on Disaster Recovery Networks* (DIREN 2002), New York.

[22] O. Nykänen. Lecture notes, http://matwww.ee.tut.fi/hmopetus/rdj/2005/arc/rdj05-214-233.pdf. Nykänen references to S. Russell and P.

Norvig. Artficial Intelligence: A Modern Approach, Prentice-Hall, Saddle River, NJ, 1995, 932 p.

[23] OASIS EU-FP6 project, retrieved 15/02/11, http://www.oasis-fp6.org/

[24] OASIS, Reference Architecture for Service Oriented Architecture, retrieved 15/02/11, http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-cd-02.pdf

[25] H. Rantanen. "Managing Emergency Response with the Help of Information Technology, A Feasibility Study in the Finnish Perspective". *Licentiate thesis*, Department of Computer Science, University of Kuopio, Finland, 2004.

[26] H. Rantanen and T. Viikinkoski. "On modeling emergency situations, In Finnish: Erityistilanne prosessina", *Emergency Service College, Research publication*. 1/2010.

[27] Reference Model for Service Oriented Architecture 1.0, retrieved 19/02/11, http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.html.

[28] OPEN Group, retrieved 23/02/11, http://www.opengroup.org/projects/soa/.

[29] ORCHESTRA, retrieved 23/02/11, http://www.eu-orchestra.org/.

[30] L. Palen, S.R. Hiltz and S.B. Liu. "Online forums supporting grassroots participation in emergency preparedness and response", *Communications of the ACM*, vol. 50 (3), pp. 54 – 58, 2007.

[31] T. Schoenharl, R. Bravo and G. Madey. "WIPER: Leveraging the Cell Phone Network for Emergency Response", *International Journal of Intelligent Control and Systems*, Vol. 11(4), 2006. http://www.nd.edu/~dddas/Papers/IJICS_WIPER2.pdf.

[32] P. Verma and D. Verma. "Internet Emergency Alert System". In *Proceeding of Military Communications Conference*, Hawthorne, NY, USA, 2005.

[33] W3C, retrieved 23/02/11, http://www.w3.org/.

[34] WIN, retrieved 11/02/11, http://cordis.europa.eu/fetch?CALLER=FP6_PROJ&ACTION=D&RCN=71841&DOC=4&CAT=PROJ&QUERY=1147880767036.

[35] WS-*, Web Service specifications, retrieved 24/02/11, http://en.wikipedia.org/wiki/WS-*.

Author Biography

Jari Leppäniemi holds a Licenciate of Technology and M.Sc. (Tech) Degrees in Software Engineering from Tampere University of Technology in 2011 and 1994. Since 2004 he has worked as a project manager and senior researcher in Tampere University of Technology Pori unit.