Capabilities of C^2 Systems for Crisis Management in Local Communities

Erland Jungert
Swedish Defence Research Agency
jungert@foi.se

Niklas Hallberg
Swedish Defence Research Agency
niklas.hallberg@foi.se

ABSTRACT

Development of information systems support for crisis management in local communities must be based on the needs of the crisis management actors as well as capabilities of information technology (IT). IT provides extensive possibilities, but is difficult to adapt to the users' needs. A profound exploration of the needs of systems for crisis management is often neglected. Hence, the management of major crisis has been held back because of the lack of useful tools. This paper presents an architecture for system for the local communities. The development of the architecture was based on a needs analysis, including interviews with crisis management at a local community level. The architecture is also based on novel design issues that have lead to the development of a conceptual model structure called the mission support model whose main concepts mainly include (1) user roles, (2) services and (3) views.

Keywords

C^2 systems, architecture, service, user needs.

INTRODUCTION

During the last decade needs for more thorough understanding of threats that may cause severe problems to whole societies has become apparent. Widely discussed are threats such as terror attacks and natural disasters caused or amplified by climate change, such as hurricanes or earthquakes. Disasters such as those may all cause considerable damage to the individuals, infrastructures, and vital functions of societies. Further, the damages may require the use of major resources for replacements and rebuilding. Hence, efforts spent on development of efficient tools for crisis management support, are motivated. The interest for information systems aimed at supporting crisis management has steadily increased during the last decade. However, many aspects need to be considered when realizing satisfactory information systems support. In systems engineering, a common reason to failure is the lack of understanding the users' actual needs, which often are taken for granted (Young, 2005, Wybo and Lonka, 2002). This is due to several reasons, e.g., lack of resources, knowledge of how to develop systems and appropriate access to user representatives (Kasser, 2007). In this paper, the term command and control (C^2) system is used for information systems supporting crisis management.

Due to the complexity of crises management, adequate C^2 systems must be founded on novel design theories. Further, the development must be based on the users' needs and requirements, e.g., regarding IT security and usability. Hence, this requires thorough studies for identification of the needs to determine accurate requirements, functional and non-functional, for development of C^2 systems adapted to the particular types of crises that may occur. In this work, the focus has thus been concerned with identification of the primary *needs* for the development of appropriate information systems applicable to the local community level. Beside this, focus has also been put on the development of an adequate architecture, based on modern design concepts. Basically these concepts are user role, view and services, which were used to define a concept model, called the mission concept model that eventually was used for development of the C^2 system architecture. Thus, the outcome of the work on development of a C^2 system architecture and its needs is discussed. In this paper, a local community corresponds to a geographically limited area, where a group of people live, i.e. a municipality. They have governance, which, amongst other things, is responsible for and has capabilities for crisis management.

PROBLEM DESCRIPTION

The local community is commonly the level on which crises must be handled, at least the first responses (Haddow and Bullock, 2006). However, it is difficult to predict the time and the locations of upcoming crises. Time is critical since rapid responses can considerably decrease the effects of a crisis. This means that it is not sufficient to build up a response from scratch when something has happened or will happen in the near future. Neither is it possible to have assigned personnel, equipment and systems on stand-by to be used only when an eventual crisis occurs. Therefore, it is necessary to make use of available resources to handle the situation, sometimes by improvising (Mendonca, Jefferson and Harrald, 2007). Extensive crisis response is a complex activity, efforts of the actors, often from different organizations, must be coordinated so that the resources are used in the right time and at the right place so that the outcome of the actions can be used by other responders to perform further actions (Shen and Shaw, 2004). It is necessary for the involved actors to, to some extent, share the situation awareness, e.g. to be able to make decisions that are coherent, or a least not are counterproductive (Oomes, 2004). The problem to be addressed in this work has been to develop an architecture for C^2 systems to be applied to crises taking place in local communities. A local community may thus correspond to a municipality or county with its own governance with capabilities for crisis management.

METHODS

The work presented in this paper was performed in three steps: (1) a literature study regarding ariktecture, systems engineering and crisis management, (2) a needs analysis based on interviews, and (3) a theoretical construction of an architecture for the development of C^2 systems for crisis management.

SERVICE ORIENTED ARCHITECTURE

Information systems (e.g. C^2 systems) are used to store, distribute, manipulate, analyse and display information. A modern approach for development of complex, extensive and distributed systems are to used so called services, which has been clamed to be useful for C^2 systems (Pilemalm and Hallberg, 2008; Jungert, Hallberg and Hunstad, 2006). Various service oriented approaches exist, among these the Service Oriented Architectures (SOA), which is a formalism intended for the design of service oriented systems. In general, services can be defined as a mechanism by which consumers' needs and producers' capabilities are brought together (MacKenzie, Laskey, McCabe, Brown, Metz and Hamilton, 2006). Thus, the roles of the service providers and the service consumers can be played by humans or systems. Services can be *elementary services* produced by a single unit or *composite services* conducted by other services (Benatallah, Sheng and Dumas, 2003). Services are motivated for several reasons among which the following can be mentioned:

- modularity,
- user orientation,
- effect,
- flexibility,
- robustness,
- interoperability,
- scalability,
- reusability.

In this context, modularity is a good quality from a systems engineering perspective, as it may enhance evolutionary systems development (McGovern, Ambler, Stevens, Linn, Sharan and Jo, 2004). Systems developed in this way are structured and thus simpler to maintain over their life cycles as services provide a standardized interface to modules which in turn enhances the possibility to reuse and evolutionary improve the usability of the systems. The latter means that the systems can be adjusted to changed circumstances. Service-oriented systems are also usage oriented, since they are described based on the effect that the use of them accomplish (Pilemalm and Hallberg, 2008). Actors, other humans or systems, can be service providers; i.e. produce an effect via a service to consumers who do not know how and by whom the service was produced. Hence, how a service is produced is transparent to the consumers. Through the use of services flexibility can also be gained and maintained. An example of this is when sensors are the main information sources. In such cases, the sensors may be of many different types that generate data of diverse types. Thus, the services called for to access such data must, from the users' perspective; work in a sensor data independent way and again this requires transparent systems. As a consequence, the users should not have to consider which type of sensors are producing the information as the sensor type may be subject to change over time, e.g. during the transition from day to night (Jungert and Hallberg, 2008). As a consequence, this requires a high level of flexibility which can be achieved by the use of services. Robustness is a further property required by modern C^2 systems. This means that the context in which the systems are used is subject to change. Such changes may be due to environmental changes which may in turn lead to the need for new functionalities that in the case of service-oriented systems can be achieved by adding or replacing existing services in the system. Thus, the use of service-oriented architectures enhances the capability for interoperability and provides more cost effective and less disruptive application replacements than traditional monolithic systems. Further, it places the focus on what the systems produce rather than how systems are implemented. Conclusively, an architecture aimed at handling operational information by means of a service-based structure can be adapted to different user roles and changing work situations.

Although the use of service oriented system architecture has several advantages, there are disadvantages as well. For instance, when the number of available services has grown to such an extent that they become overwhelming to the service consumers, e.g., which service should be used under certain circumstances and also for what purposes. Another aspect concerns questions such as: "Am I allowed to use a particular service provider considering existing legislation". An example may be that a certain service producer must be the most inexpensive one, which must be the case when called for by users working for e.g. the government. Hence, a service-oriented C^2 system for crisis management to be used in local communities must be able to handle legislation and other circumstantial regulations regarding service availability.

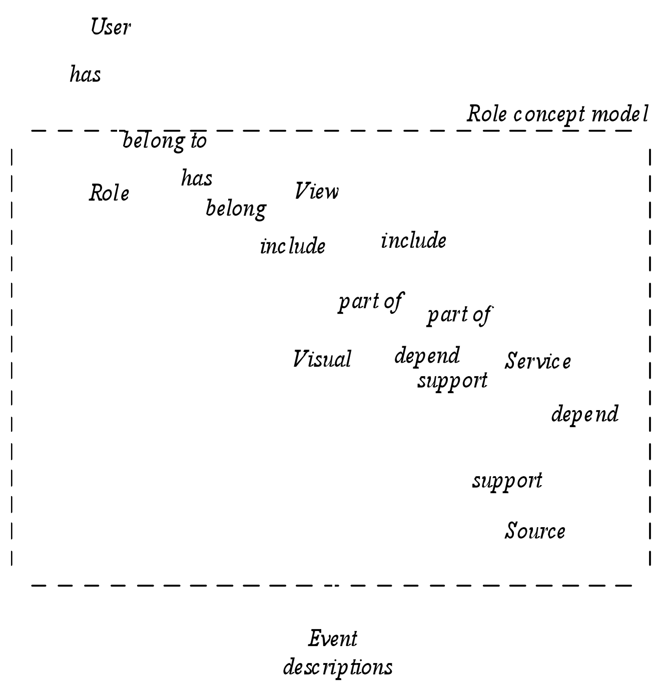

Figure 1. The mission support model and its user relations.

In this work services are associated to views, which in turn are associated to specified user roles. An outcome of these aspects has lead to the development of a model describing the interrelations between them; this model is here called the role concept model (Figure 1) (Jungert and Hallberg, 2008). This model, which can be regarded as an ontology, illustrates that every user is given one or several roles, that to each role a set of views are attached and that each view is given a set of services. Each view is also attached to a *visual,* corresponding to a visualization of view instances connected. In each visual, the information presented is collected through the services attached to the corresponding view. Most relations in the model are of many-to-many type. For instance, several views, including their visuals and services, may be connected to each single role and a particular view may be used by several user roles.

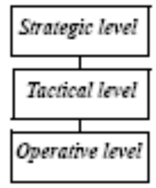

Figure 2. The hierarchical structure of the C^2 model.

The views, as they are proposed here, should be seen as a means to enhance heterogeneous user categories' usage of common operational pictures and, thereby, to enhance their situation awareness (e.g. Artman, 1999; Jungert 2009). The work of Jungert (2009) primarily concerns how to improve C^2 systems design, i.e. a theory that can be used to extend the architecture presented in this work. The theory is founded on a hierarchical C^2 structure (Figure 2) which shows the three level C^2 structure. A further objective of this work has been to provide the users with improved situation awareness based on a structured and shared operational pictures described in terms of the presented theory.

Thus the architecture presented here can be seen as an architecture for systems that handle operational information by means of a service based structure that can be adapted to the roles of the different actors and their work situation. The operational information will be presented by means of the views in a suitable way and will thus supply the users with sets of views, adapted to their particular user categories by means of available services directly associated to the views. Each view includes and presents a dedicated set of information. A view is a representation of a situation from the perspective of a related *set of concerns* (IEEE 1471, 2000). Each view instance is produced by the use of one or several services.

The role concept model demonstrates not just the relations between the roles, the views and the services but describes also how they relate to the data sources. For each role the generation of the individual operational pictures is supported by means of the services attached to the corresponding view via the visuals and down to the event descriptions, where the corresponding operational information becomes available to the users. The event descriptions contain from the users´ perspective collected descriptions of the ongoing events in the crisis, i.e. the operational picture. The model demonstrates how situation awareness, required in command and control processes, can be gained and improved by means of the event descriptions containing information gathered from the various data sources. Information sources may here correspond either to sensors, other external source or other users, playing different roles, who a certain user may share information with. Information sharing can be seen as a service as well. As a consequence, any system based on this model can be seen either as a view or as a service.

NEEDS FOR C^2 SYSTEMS SUPPORT IN CRISIS MANAGEMENT

The development of the C^2 systems for crisis management in local communities must be based on a careful needs analysis. In this project, six interviews was performed with representations for the main actors active at the local level; (1) The Municipality Administrative Office (MAO), (2) The County Administrative Board (CAB), (3) The County Council/Medical Service, (4) The Police Force, (5) The Fire Department, and (6) SOS Alarm. SOS Alarm is responsible for handling emergency calls and coordination of ambulance transportations. C^2 systems for those organizations are to be used in the management of the every day work as well as in situations of crisis. The interviews were based on semi-structured questions, which were critical incidents technique oriented (Flanagan, 1994). That is, the questions targeted issues experienced as problematic and having high impact on the crisis management results. All interviews were recorded and transcribed. From the transcriptions, statements that direct or indirect revealed information about needs were identified. Those statements were carefully analyzed to indentify the needs. Thereafter, the needs were categorized, copies were removed and the wording was unified. The outcome was 14 top level categories of needs containing 344 needs on the lowest level. The 14 top level needs are:

- Elucidate the actors' roles; each actor's mandates, responsibilities, commissions and relationships to other actors must be visible.
- Handle alarms; actors need support to receive and verify, analyze and to send alarm further, e.g. to affect parts within the own organizations.
- Communication; actors need communication support within the own organizations, the other crisis management actors, the public and the media.

- Handle information; actors need to be able to obtain, receive and process information. Including the need to assess the quality of the information and to maintain an updated operational picture. They need to exchange, coordinate and disseminate information.
- Information contents; actors need information about available resource (own and others), the environment, the course of events, chemical substances and earlier occurrences.
- Preventive initiatives; actors need support to perform risk- and vulnerability analyses and to conduct action plans.
- Operational initiatives; each actor needs support to solve their specific tasks, to recreate a normal situation.
- Collaboration; crisis management actors need to collaborate with almost every actor in the local community, but also with some actors on regional and national level. On local level, working methods and systems must support collaborations. Uniform and shared concepts are essential.
- Management; support is needed for planning of actions, exercises and organization. Actors need support for assessment of the situation and to activate crisis management organizations. They need support for operational management including decision-making, prioritization, optimization, resource adaption and to request resources. They need to coordinate efforts and resources and to make resource inventory.
- Education and exercise; support is needed to practice own actions, collaboration with others and information management.
- Evaluation; support to evaluate training, practices and efforts but also to make use of gained experiences
- Information security; actors responsible for crisis management systems need support to handle privacy, usage rights, availability, safety levels and security management processes.
- Documentation; actors need support for documentation of efforts, training and events.
- Act according to legislation; support to assess an act contradicting laws and regulations. Also, what laws and regulations are affected by the activities.

CONCEPTS FOR REALIZATION OF THE SYSTEM

The role concept model demonstrated the three fundamental concepts used for the realization of the C^2 system architecture introduced in this system, i.e. (1) roles, (2) views and (3) services. These three concepts, used to maintain a C^2 system with a common (or shared) operational picture, are further discussed in (Jungert and Hallberg, 2008). The purpose of the role concept model is to supply the users of the eventual C^2 system with customized situation awareness for decision making during crisis management; particularly in applications related to the local community.

Different users have different responsibilities and tasks to carry out throughout the crises. The domain of these activities determines also the corresponding user roles. Further, the different tasks and responsibilities, and thereby the different roles, require different information. Crisis management systems must be sufficiently flexible and adaptable to the roles given to the users. Further, the realizations of the operational pictures must be adapted to the requirements of the different roles. A role is thus generally defined and carried out with respect to its tasks and responsibilities, but it also depends on its organizational belonging. The roles and the views in the architecture make the system usage oriented, i.e. the focus of the systems is to meet the roles' momentary information needs and on the effects of using different functions. This also makes it possible to allow for a modularity that enables determination of the set of views suitable to each role including the services associated to the views. However, this leads to questions like which are the necessary roles that need to be identified. In this work, this has been concerned with identification of the roles of interest to crisis management in the local community; the outcome of this work includes members from the following organizations:

- Medical care providers (MCP)
- Police department
    - Onsite command (OC)
    - Regional command (RC)
- Fire department
    - Onsite command (OC)
    - Local command (LC)

- Information coordinators
- Analysts
- Municipality board (MB)
- County administration board (CAB)

Each role must be assigned a set of views, which also may vary over time as the needs of the roles change. Nine different views have been identified for the version of the system applied to local crisis management. Of these nine views, five are part of the operative section according to the main level structure of the system (Figure 2). The five views present in the operative section are (1) the current operative view (COV), (2) the imported data view (IMV), (3) the sensor data view (SDV), (4) the allocated resource view (ARV) and finally (5) the a_priori data view (APV). APV contains information of general character such as names and telephone numbers of e.g. experts. The four additional views are the *context view* (CXV), the *History view* (HYV), the *Planning view* (PLV) and the *Service availability view* (SAV). These four views reside in the corresponding sections given in Figure 2. The nine views are, depending on the requirements of the roles, assigned to the ten roles identified for the local community crisis management (Table 1).

Role	COV	CXV	IMV	HYV	ARV	PLV	APV	SAV	SDV
MCP	X		X		X	X	X	X	
PD, OC	X	X	X		X		X	X	
PD, RC	X	X	X		X	X	X	X	X
FD, OC	X	X	X		X		X	X	
FD, LC	X	X	X		X	X	X	X	X
SOS-Alarm	X	X	X		X	X	X	X	
Information co-ordinator	X	X	X				X	X	X
Analyst	X	X	X	X	X		X	X	X
MB	X	X	X				X	X	
CAB	X	X	X				X	X	

Table 1: Role/views attachments

Each role is associated with the building blocks of the system, i.e., the views including the services needed to collect the information necessary to carry out the assigned tasks. Through the adaptation of the system to the different roles it will be possible for the users to create their individual operational pictures. That is, the users interact with the entire operational picture by means of the services associated with the views and dedicated to their roles. Further, the operational pictures are maintained by the system momentarily. Thus, for each view instance there must be an indication of whether it is a member of the common operational picture or just a member of the individual operation picture.

THE ARCHITECTURE

The architecture of the view-oriented operational picture system is on the highest system level constituted by the given set of sections (Figure 3). Each section is assigned a number of tasks supported by one or several views. Since this level of the architecture is application dependent not all of the sections are mandatory for all systems, i.e. depending on which application the system is supposed to support different sections have different levels of relevance. The section structure is basically hierarchical with three levels. Further, the service exchange section is available for exchange of information with external users via various service calls and service requests; the intention is consequently to allow requests for and distribution of view instances between users and groups of users. Figure 3 illustrates the architecture of the system, including the sections, the views and their corresponding data repositories. Each section has its own functionality, set of views and services to fit its intended use.

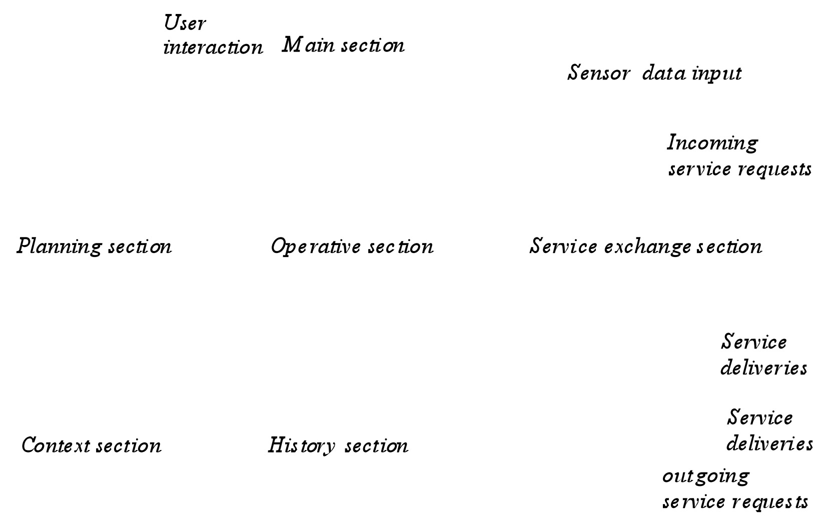

Figure 3. The section structure level of the system architecture.

The main section differs in various ways from the rest of the sections as it does not include any views but is a control and interaction module that may request three major services. These are (1) *the operative section service*, (2) *the allocation planner section service* and (3) *the service exchange section service*. The operative section service supports the users in maintaining their individual operational picture that will include shared information. The support of this is carried out by means of the views available in the operative section. The allocation planner section service, which is not needed for all roles, supports the allocation of the required resources necessary to perform the planning activities. The purpose of the last major service, the service exchange section, is to support requests for information through service calls to collaborating actors as well as to support other actors through the fulfilment of their information requests. The main section also includes an application database where information received from external field sensors reside. The main section can be seen as a C^2 environment with the purpose to control the general activities and to carry out the crisis management process. *The planning section* is the main part of the allocation planning system, which is directly connected to the allocation planner in the main section.

On the lowest level of the architecture there are two sections, the *Context section* and the *History section.* The Context section includes the context information from which background information related to various view instances are selected, transferred and integrated with the current operational information to form a view instance or to be integrated with any view instance that requires background information. The context section may include information such as maps, satellite images, building blueprints and locations of dangerous substances. The history section is a repository for the historic view instances of all views. Hence, when a view is updated a new view instance replaces the old one and the old view instance is transferred and stored in the history section. Whenever necessary, the historic view instances can be reinstalled, e.g., for analysis of incidents and review of taken actions. This is particularly relevant to analyze because of what has happened during a certain time frame; that is, when for instance several view instances for the same views are scrutinized.

The views and the sections included in the proposed architecture correspond to the systems structure in Figure 4. However, certain views and sections may not always be appropriate for certain roles, i.e., users and organizations. For this reason, system versions adapted to different roles must be allowed by excluding system facilities (views and sections) that may not be of relevance for a specific role. Hence, the configuration of the users' access rights can be accomplished by excluding certain views and/or services. This will result in a technique for user adaptation, which only provides the users with the means necessary to perform their given tasks, excluding superfluous information accesses. Commonly, during crisis situations large amounts of data are obtained, provided by different types of sensors and observers. Especially the former are associated with uncertainties of various kinds. To handle these uncertainties, methods for sensor data fusion are required (e.g. Hall and Llinas, 2001). Higher levels of information fusion are also needed, especially to support the services in the operative section but also to perform risk analyses.

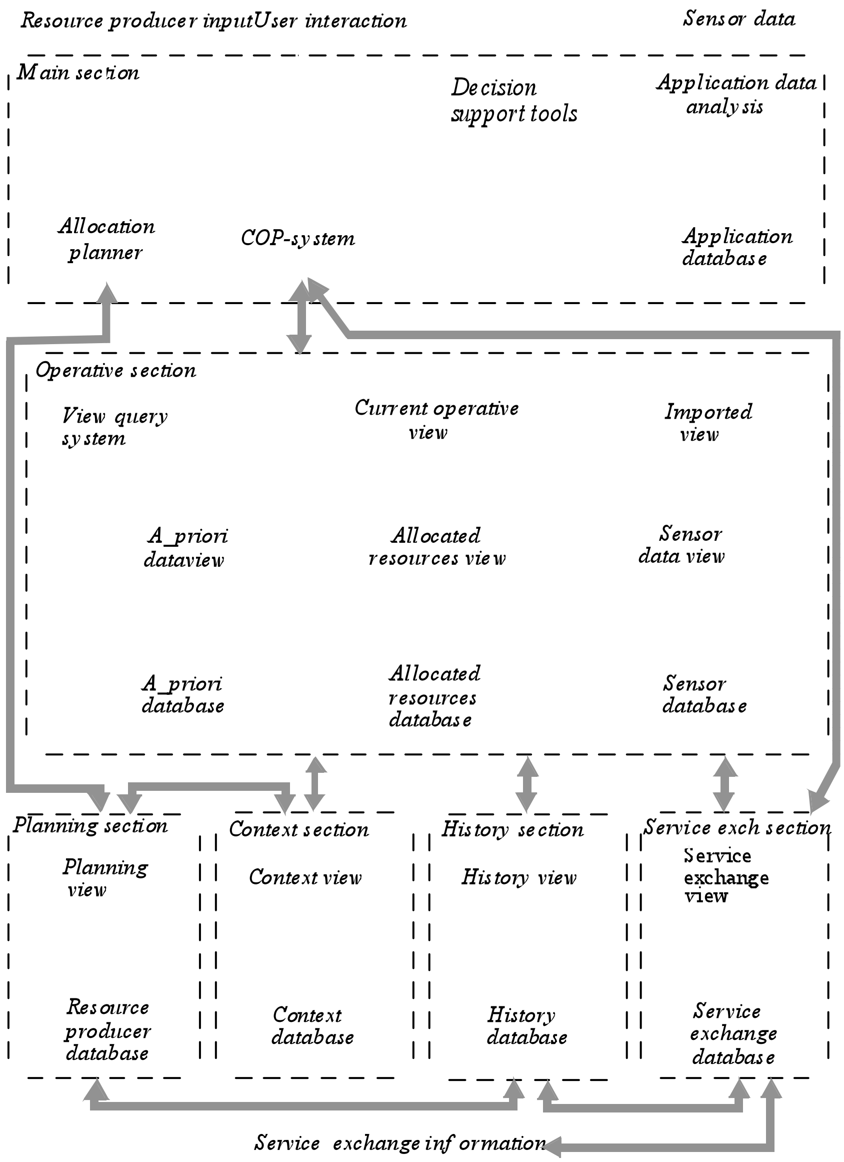

Figure 4. The system overview of the view-oriented system for presentation and maintenance of operational picture information for crisis management.

The query system used in the main section works on streams of incoming sensor data and includes mechanisms for sensor data fusion. It is also designed to work in a sensor data independent way where the sensor data sources generally deliver heterogeneous data that must be possible to transform into a unified structure. A system of this kind requires certain types of decision support tools. In this context the main tool for selection of information from the large amount of information is a query language tool with capability for handling both heterogeneous sensor data from a large verity of sensor types as well as all other data available in the view instances. The principles of the query language used here are described in (Chang Costagliola, Jungert, and Camara, 2009) and from a scenario driven perspective in (Camara and Jungert, 2007).

CONCLUSIONS

This paper presents an architecture for development of C^2 systems for crisis management. It is based on general needs and requirements, but also on our experiences from development of C^2 systems for crisis management for local communities. The architecture is based on the tree concepts (1) user roles, (2) services and (3) views. It is difficult to state the usability of such an architecture, as the one presented, before its usage in several development projects have been evaluated. However, from a theoretical perspective the concept of user roles is aimed support the design of functions that the C^2 system provides to fit specific users' different needs. This is needed, since in crisis management several different categories users and organizations have to work together. The service concept is introduced to enhance information exchange and ability to evolutionary development of C^2 systems. This approach can be adopted to many different situations, but also to extend the systems without first shutting them down. Views are aimed at supporting adoption of shared information between users, but still adapted for specific users and situations. In future and ongoing evaluations we hope to provide evidence showing that these assumptions are valid and that the architecture supports the developments of C^2 systems that are flexible and adaptable to meet the demands of crisis management.

ACKNOWLEDGMENTS

This work has become possible due to grants from the Swedish Emergency Management Agency (KBM).

REFERENCES

1. Artman, A. (1999) Situation awareness and co-operation within and between hierarchical units in dynamic decision making, *Journal of Ergonomics*, 42, 11, 1404-1417.
2. Benatallah, B., Sheng, QZ. and Dumas, M. (2003) The Self-Serv environment for Web services composition. *Internet Computing, IEEE*, 7, 1, 40 – 48.
3. Camara, K. and Jungert, E. (2007) A visual query language for dynamic processes applied to a scenario driven environment, *Journal of Visual Language and Computation, special issue on Human-GIS Interaction*, 18, 3, 315-338.
4. Chang, S.-K., Costagliola, G., Jungert, E. and Camara, K. (2009) Intelligent Querying Techniques for Sensor Data Fusion (accepted for publication), *Intelligent Techniques for Warehousing and Mining Sensor Network Data*, IGI Global, Hershey, PA.
5. Haddow, G. D. and Bullock, J. A. (2006) Introduction to Emergency Management. Butterworth-Heinemann, Boston, MA.
6. IEEE 1471 (2000) IEEE Recommended Practice for Architectural Description of Software-Intensive Systems, Std 1471-2000, New York, NY.
7. Flanagan, J. C. (1954) The Critical Incident Technique. *Psychological Bulletin*, 51, 327-58.
8. Hall, D. L. and Llinas, J. (Eds.) (2001) Handbook of Multisensor Data Fusion, 1007 CRC Press, London.
9. Jungert, E. (2009), A Theory on Management of Shared Operational Pictures for Command and Control Systems Design, *Proceedings of International Conference on Information Systems (IADIS 2009)*, Barcelona.
10. Jungert, E. and Hallberg, N. (2008), An Operational Picture Systems Architecture for Crisis Management, *Proceedings of the 14th International Conference on Distributed Multimedia Systems (DMS'2008)*, Boston, MA.
11. Jungert, E., Hallberg, N. and Hunstad, A. (2006) A Service-based Command and Control Systems Architecture for Crisis Management, *The International Journal of Emergency Management*, 3, 2, 131-148.
12. Kasser, J. E. (2007) A Framework for Understanding Systems Engineering. The Right Requirement, Cranfield, UK.
13. MacKenzie, C. M., Laskey, K., McCabe, F., Brown, P. F., Metz, R. and Hamilton, B. A (eds.) (2006) Reference Model for Service Oriented Architecture 1.0 Committee Specification 1, 2 August 2006 http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm (2008-01-24).
14. McGovern, J. Ambler, S.W. Stevens, M.E. Linn, J. Sharan, V. and Jo, E.K. (2004) A practical guide to enterprise architecture, Prentice Hall, Upper Saddle River, New Jersey.
15. Mendonca, D., Jefferson, T. and Harrald, J. (2007) Collaborative adhocracies and mix-and-match technologies in emergency management, *Communications of the ACM*, 50, 3, 44-49.
16. Oomes, A. H. J. (2004) Organization awareness in crisis management – dynamic organigrams for more effective disaster response', *Proceedings of the 5th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2004)*, Brussels.

17. Pilemalm, S. and Hallberg, N. (2008) Exploring Service-Oriented C^2 Support for Emergency Response for Local Communities. *Proceedings of the 5th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2008)*, Washington, DC.
18. Shen, S. Y. and Shaw, M. J. (2004) Managing coordination in emergency response systems with information technologies, *Proceedings of the Tenth Americas Conference on Information Systems*, New York, NY.
19. Young, R. R. (2001) Effective requirements practice, Addison-Wesley, Boston, MA.
20. Wybo J. L. and Lonka, H. (2002) Emergency management and Information Society: how to improve the synergy?, *International Journal of Emergency Management*, 1, 2, 183-190.