A Model for Temporal Coordination of Disaster Response Activities

Jörn Franke
SAP Research Sophia Antipolis, France
joern.franke@sap.com

François Charoy
LORIA-INRIA, UHP, Nancy-Université
charoy@loria.fr

Cédric Ulmer
SAP Research Sophia Antipolis, France
cedric.ulmer@sap.com

ABSTRACT

One problem for public safety organizations in a disaster is the management of response activities and their dependencies on an intra-and inter-organizational level. Our interviews with end users have shown that current solutions for managing activities are complicated to use in the crisis by teams in the field and also in operation centers, when facing continuous unexpected events and cross-organizational activities. We propose an activity centric system for managing crisis response activities for such situations. We give an example how this system is used in a crisis within one organization and cross-organizations. Afterwards, we explain the evaluation of the solution. This research contributes not only to the crisis management domain, but also to the business process management domain by providing an alternative view on activities in highly dynamic scenarios.

Keywords

disaster, ad-hoc, activity, governance, dependency, management.

INTRODUCTION

During and after a disastrous event, response organizations start to coordinate themselves according to plans. However, these plans can be challenged by the dimension of the event, by its unexpected consequences and by the appearance of related or unrelated new events. The plans have to be adapted, new plans have to be decided and new organizations have to be incorporated. In large scale disasters, such as September 11, hundreds of organizations and thousands of people can be involved. The activities of these different organizations may depend on each other. These dependencies can be of different kinds: temporal, spatial, resources etc. The idea to support the description of the relationships between these activities using information technology is not new, but leads usually to very large models with many different elements, which difficult to manage and change, in particular in situation of continuous changing events (i.e. new activities are executed, old one fail or are idle etc.) and sense-making. This is even more difficult on the cross-organizational level, mixing different cultures, procedures and regulations.

The goal of this paper is to present an activity management system for disaster response activities grounded in various research approaches, such as interviews, case studies and workshops. Activities can be (re-)combined depending on the need of the situation. We do not focus on how these activities should be visualized, but on technical support for temporal coordination of these activities by a system. The main motivation for our work is that we have identified in our research that there is a need for an ad-hoc activity management system for response teams in the field, operating centers and command centers on the intra-and inter-organizational level. We investigate the problem domain and related work in the following section. In the third section, we describe the design of the activity management system. The design is evaluated in the fourth section and this includes the description of advantages and limitations. Finally, we give an outlook of our future research.

Reviewing Statement: This paper has been fully double blind peer reviewed

PROBLEM DOMAIN

First, we will investigate the problem domain by explaining our research approach in the following subsection. We describe the issues related to the use of business process management systems for managing response activities in the third subsection. Finally, we will describe a problem case illustrating the difficulties for applying business process management systems to crisis response activity management in the fourth subsection. This is the foundation for designing our activity management system for crisis response activities in the next section.

Research Approach

Our research approach is grounded on constant relationships with actors that have already been involved in crisis management (Franke and Charoy 2010). Several interviews and workshops have been conducted to investigate integrated information technology support for crisis response management in the SoKNOS project (SoKNOS 2009). The interviews have been conducted with police man and fire fighters in leadership positions with several years of experience in disaster management. A large scenario for a flood has been built together with end users based on real previous disasters. This included also analysis of real disaster response plans provided by the end users in the project. Afterwards, we modeled together with end users response activities using a typical business process management system (see next subsection).

Related Work

Although many systems have been proposed for resource management during a crisis, little has been published in the area of coordinating disaster response activities. The problem is related to the sense-making problem: Who is doing what and how is it related to the own activities. There have been proposals to extend business process management systems for routine emergency processes (e.g. (Rüppel and Wagenknecht 2007; Sell and Braun 2009)). According to our interviews with police forces (Franke and Charoy 2010), this leads to complex models during dynamic situations. Firstly, they are complex in terms of model size. Secondly, they are complex in terms of number of element types. Thirdly, they are complex in terms of number of modeling elements. Models thus become difficult to manage, in particular for unexpected events. Those are frequent during disasters (e.g. activities fail or new activities are created). Yet approaches based on business process models and workflow systems assume that activity states are changing sequentially (i.e. one activity has to finish before another activity can be in the state execute), processes are mostly fixed and processes do not have any relationship with other response processes. When addressing cross-organizational processes, business process models approaches introduce many requirements (Chebbi, Dustdar et al. 2006; Eshuis and Grefen 2007), because process models need to be exchanged. Assuming a crisis management team in the field changes the process (e.g. adds an activity), and that at the same time a person in the command center also changes the process (e.g. removes an activity), then the system needs to merge at least two different versions of the same process model. This is very difficult to implement and introduces time-consuming conflict resolution mechanisms. This leads also to unnecessary confusion among the stakeholders.

Several publications propose to model different scenarios using process models and to compose these scenarios depending on the situation (Georgakopoulos, Schuster et al. 2000; Fahland and Woith 2008). The scenarios have to be defined and planned in advance. This means the approach requires a modeling phase, which makes it more suitable for predictable emergency scenarios. It is also not able to deal with unexpected events as they occur in disasters. Catarci et al. chose an activity centric point of view governed by logical rules: in highly dynamic situations it leads to a more suitable system support (Catarci, T., de Leoni, M. et al.). Llavador et al. propose a similar approach to this for managing emergency processes (Llavador, Letelier et al. 2006). Both approaches use a logic language to define the rules applied to process models. However it can become very difficult to describe a situation in a textual way in a logic language. They are not effective for controlling and managing a process, because rules systems and logic systems are designed for reaction and less for control. The work of van Someren et al. deals with the information flow between tasks ((van Someren, Netten et al. 2005) and can be seen as complementary to our work. All these approaches do not consider governance, e.g. clearly defined accountability and responsibility.

Lu et al. propose to define temporal constraints for business process models (Lu, Sadiq et al. 2006). These constraints are applied to business process models with sequential activities and high frequency of execution. They assume sequential activities and do not consider activity states and the model still require to be defined precisely in advance. They also do not consider different kind of visualizations for different target groups and it is not clear how their framework can support the execution of activities.

Crisis Management Activities differ from Business Processes

Different types of processes have been identified in the field of computer supported cooperative work (CSCW). McCready describes three types of processes in the business process domain: ad-hoc, administrative and production (McCready 1992). They all require defining sequential activities and isolated processes. Most of the process types can be applied to disaster management for administrative processes. For example, public safety organizations have business processes like controlling, human resource management or training processes.

However, crisis response activities do not seem to fit anywhere in this scenario as our interviews have shown. We tried to model response processes with a business process modeling language (event-driven process chains (Scheer 2000)) together with the end users, such as fire fighter and police within the SoKNOS project. We did not investigate other business process modeling languages, such as the Business Process Modeling Notation (BPMN), because they can be considered as similar (Recker 2008). Modeling the processes is mandatory for managing the activities by a process management system. Our modeling approach was not successful: the models were usually large (i.e. many activities), but meaningless (i.e. every activity runs in parallel without any other information). When integrating governance roles (e.g. accountability or responsibility) they became very complex (i.e. using many different more complex model elements, such as "AND", "OR" and "XOR"-Connector for routing the control-flow). We could also not clearly define isolated processes in a disaster response. Relative temporal dependencies between parallel activities cannot be modeled using business process modeling languages (see also section "Example"). The end users told us that business process models cannot adequately capture response activities. The problem is that we do not have only activity sequences or clearly isolated processes like in business process models. This means systems using business process models for describing and managing business processes (e.g. workflow systems) are not very suitable for managing disaster response activities with many exceptional events. This was not due to the lack of modeling skills, because the models have been developed together with business process experts. The modeling language and elements were not suitable to capture the important aspects in response scenario. This has been also confirmed by others (Georgakopoulos, Schuster et al. 2000; Denning 2006; Catarci, de Leoni et al. 2008; Fahland and Woith 2008). Our analysis of disaster response plans (generic and disaster specific, such as flood response) by police and fighters in the SoKNOS project also has shown that they only contain a list of activities, which cannot be mapped directly to business process models.

Problem Case

We illustrate the problematic of response activities and their dependencies in this section using a simplified problem case (see Figure 1) based on a realistic flood disaster scenario developed together with end users in the SokNOS based on real previous disasters. There are three different organizations responding to the flood.

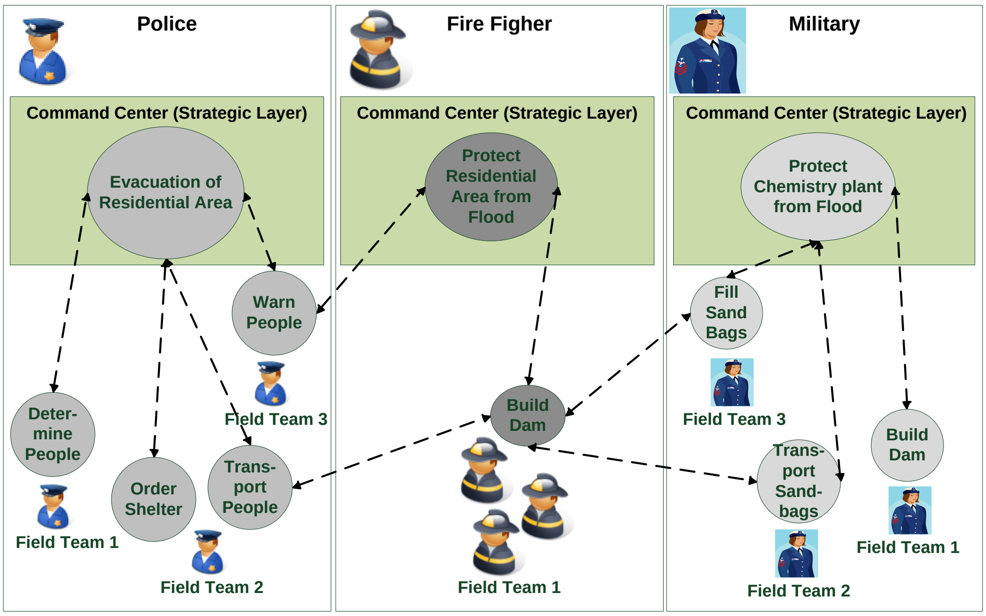

Figure 1 Problem case from an activity point of view

The police are evacuating a residential area threatened by the flood. The fire fighters are protecting the residential area by building a dam.

The military forces are protecting a chemistry plant from a flood by building another dam. They are also responsible for filling and transporting sand bags for themselves and the fire fighters.

Several activities are executed by different organizations consisting of command centers and field teams. Activities are represented as circles in the figure. Every activity runs in parallel to the other activity, when considering the management lifecycle of the activities. For example, in the beginning all activities are planned and then some start to execute. Others can be idle, because they are waiting for something. Others may fail or are cancelled. Different types of activities may have a different management lifecycle. For example, logistics activities, such as transport sandbags, are managed differently than evacuation activities, such as warning people. There is also a need for defining clear accountability and responsibility (i.e. governance) for each activity. For example, the police chief is accountable for the activity "Evacuation of Residential area", but the police field team is responsible for its execution. There are dependencies between activities illustrated as dotted lines.

Currently, we have not identified much research on how these kinds of dependencies can be supported by a crisis response activity management system. As mentioned before, many of these temporal dependencies cannot be managed using business process management systems (see also section "Example"). For example, we cannot model that "Evacuation of Residential area" can be executed as long as "Protect Residential Area from Flood" is executed. We describe in the following section how such as system supporting such a situation can be defined.

A SYSTEM FOR RESPONSE ACTIVITY MANAGEMENT

The goal of this section is to describe the design of our system. First, we will describe the requirements gathered from case studies within the SoKNOS project and outside of it (e.g. (Wachtendorf 2000)), interviews and workshops with end users. Afterwards, we will explain how activities are described in the system. We will explain how the system can support execution of these activities in the third subsection. In the last subsection we will describe how activities can be visualized in different contexts. Our model is simple and generic and can be extended easily.

Requirements

Based on the analysis of the problem domain in the previous section we derived the following requirements for a system for managing response activities:

1. Description of activities and dependencies in the system: Activities can be created ad-hoc (e.g. by people in the field or command center). Different types of activities, such as decision-making activity or operation in the field, need to be described differently, because there is a different management process for them. The way activities are executed, and the governance roles constraining their execution can change depending on the nature of the activity. Even dependencies can occur at a finer level of granularity than just the activity themselves. The resulting model should support these kinds of description. This is the result of our research and currently this is not supported adequately by current business process management systems.
2. Execution: Execution is described as state change of activities (planning, executing, finishing, failing, cancelling etc.). State changes of all activities can take place concurrently but can be controlled through the dependencies that may exist between them. From our interviews we drew the conclusion that most of the activities are running in parallel (i.e. change their state in parallel). Each state change may violate dependencies. This needs to be managed by the system but in a flexible way. Dependency violation may be forbidden, corrected or just notified. This is very situation dependent and need to be taken into account in the model.
3. Visualization: Each user can visualize the activities and their dependencies differently, e.g. by providing a map of activities or an activity matrix. This requirement results from the fact that each organization has already means for visualization of the response activities and they are used to them.

In the following subsections, we explain the design of our system that tries to address these requirements.

Modeling of Activities in the System

The model we propose allows to describe activities and dependencies between them and to manage activity execution. This model can be represented from different perspectives (see section "Visualization"). For example, teams in the field need a simple view on the activities and people in the command center require a more detailed version. We distinguish between three elements that can be described in the system: activity types, activities and dependencies.

An activity type is described as: $a_d = (SA, f, G)$ with

- S is a finite set of activity states
- $SA \subseteq S$ is a subset of activity states for the activity type
- $f : SA \rightarrow SA$ is a transition function defining the possible transitions from one state to another
- $G = \{g_a, g_r, g_c, g_i\}$
    - $g_a \subseteq SA$ is the transition function of the accountable role for the activity. Accountability describes who decides ultimately on the activity.
    - $g_r \subseteq SA$ is the transition function of the responsible role for the activity. Responsibility describes who executes the activity.
    - $g_c \subseteq SA$ is the transition function of the consulted role for the activity. Consulted describes who should be consulted prior a state change.
    - $g_i \subseteq SA$ is the transition function of the informed role for the activity. Informed describes who is informed after a state change.

Figure 2 illustrates an example for an activity type representing the management lifecycle of an activity. In this example we have different states (e.g. "Plan" or "Execute") and roles assigned to the transition between states (e.g. "accountable, consulted" on the transition between the states "Plan" and "Execute"). This means that only these roles are allowed to do the state change. For example, if the commander has the role accountable then only he/she can change from "Plan" to "Execute".

An activity is described as follows: $a_i = (uid, name, cs, type, GA)$

- uid is a unique identifier of the activity
- $name$ describes the activity
- $cs \in SA$ is the current state of the activity
- $type$ is the activity type (a_d) of the activity
- $GA = P \times G$ describes the assignment of governance roles to participants
- P is the set of assigned participants (user of the system, e.g. commander in the field or operating center)

Further data can be attached to the activity (e.g. geographical position, resources, deadlines etc.), but we do not prescribe how this data should look like or interpreted.

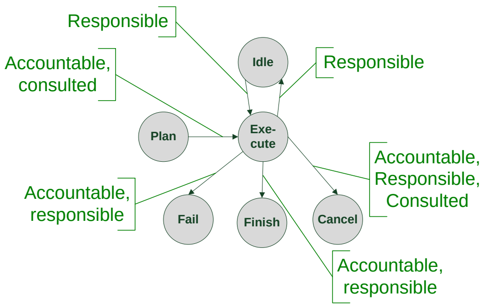

Figure 2 Example: Activity Type

A dependency is described as relationship of states between two different activities: $d_i = (a_s, s_s, a_d, s_d, t)$

- a_s Source activity
- s_s Source state
- a_d Destination activity
- s_d Destination state
- t Dependency Type

We use Allen's proposed thirteen time interval relationships to illustrate our dependencies (see Figure 3 - Please note that only seven dependencies are shown, because the other dependencies are just the inverse of six of the dependencies) (Allen 1983). The dependency type refers to one of these proposed time interval relationships.

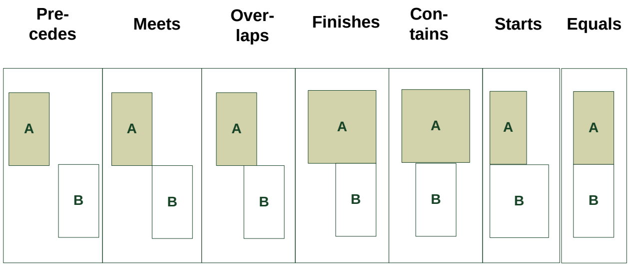

Figure 3 Allen's proposed time interval relationships

These dependencies are qualitative (i.e. they do not require explicit date and time definitions), exhaustive (i.e. there are no other possibilities) and distinctive (i.e. each relationship has a different meaning). It should be highlighted that our temporal dependencies are relative and not absolute. We describe them relative to other activity states. The main reason for this is that it is very difficult to prescribe exact times in such a scenario. However, this does not mean that we cannot integrate deadlines with precise date and times in our concept. The dependencies are established between states to be able to deal with exceptional situations (e.g. a failed activity leads to execution of other activities).

We describe in the following section an example for activities and dependencies as well as a mechanism for executing activities.

Execution

The support of the execution of the activities within the system is an important part of our work. The execution has the following basic steps:

- State change of an activity is requested (by human or machine agent)
- If state change is allowed by governance role then continue with execution
- Setup the list of dependencies which are violated by the state change

We do not describe the technical details of the execution mechanism here, because this is not the scope of this paper. Basically the system offers different choices to deal with violation of dependencies: prevent the state change (enforcement), trigger other state changes (automation) or display violation of dependencies (support). The user can describe in the system, how the system should deal with the dependency. We were able to implement this mechanism using finite state machines representing the dependencies.

Figure 4 provides a simple example for the description of activities and dependencies as well as their execution.

Two activities "Determine People" and "Transport People" have a dependency between their states "Execute". This dependency says that the state "Execute" of the activity "Determine People" overlaps with the state "Execute" of the activity "Transport People". In the first phase, both activities are in the state "Plan".

In the second phase, the activity "Transport People" changes its state from "Plan" into "Execute", which is a violation of the dependency: the activity "Determine People" has to start its execution first. In the example, the user is notified about this violation. It could also notify the role of the other activity to go into execution state.

The user can now change the state of activity "Transport People" back to "Plan" and wait for the activity "Determine People" to be in the state "Execute" as illustrated in the third phase.

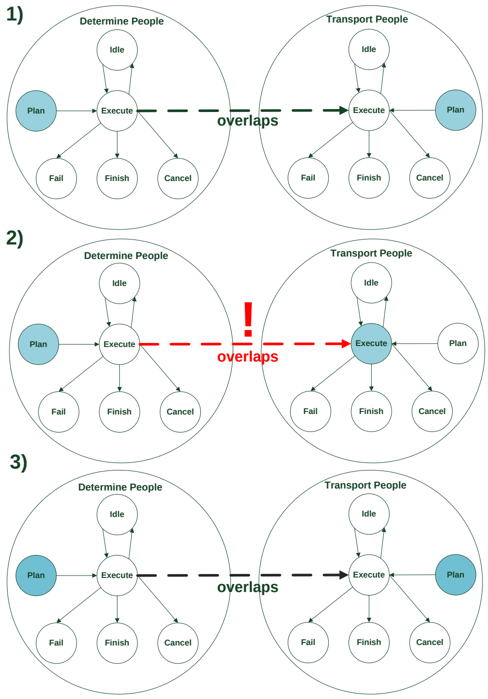

Figure 4 Example for Execution of Activities

However, there might be also other possibilities (e.g. changing the activity "Transport People" into the state "Idle"), which can be suggested by the system. A more constraining situation could for instance force "Determine People" to be in the state "Execute" or would simply forbid the state change from the "Transport People" activity.

There will be of course more activities and dependencies during a response, but this does not change the general concept. It should be noted that it is just an illustrative example and different types of visualizations may exist.

Visualization

It is important that the system is able to visualize the activities and dependencies in a different context. For example, they can be shown on a map or a more simplified view can be provided (cf. Figure 5). The first approach provides more information, but people might get lost in a complex situation with many activities. The latter one is useful for getting a quick overview on activities and dependencies, but it contains less information. Teams in the field need a much more simplified view on the activities, because they have to cope with the situation in the field. This has been a result of the discussion with end user. There may different kind of visualizations depending on the situation and users using the system. Visualization of our activity model is a whole other research area, which requires further interviews and workshops. For our system it is important that different kind of visualization of the activities can be easily supported.

Example

We describe in this section how the system is going to be used in a disaster exercise or a real disaster. Users (e.g. teams in the field or in the operating center) may load a set of activities and dependencies described in a plan. In the beginning responding organizations rely on plans. During the response they may create new activities and dependencies depending on the need of the situation. They can also exchange activities with other people and integrate the exchanged activities with the own ones.

Four activities have been selected from the previous scenario (see Figure 5) to illustrate the concepts of temporal dependencies: “Warn People”, “Protect Area” (Protect Residential Area from flood), “Determine People” and “Transport People”. “Warn People” is the activity for warning the people in the residential area that they are going to be evacuated. “Protect Area” is the activity to build a dam to protect the residential area from a flood. “Determine People” is about searching for people in the residential area to evacuate. “Transport People” is the activity for transporting people out of the area. We assume that these activities have the same activity type, with a “Plan”, “Execute” and “Finish” state. The role “accountable” is allowed to change the state from “Plan” to “Execute”. The role ”responsible” is allowed to change the state from “Execute” to “Finish”.

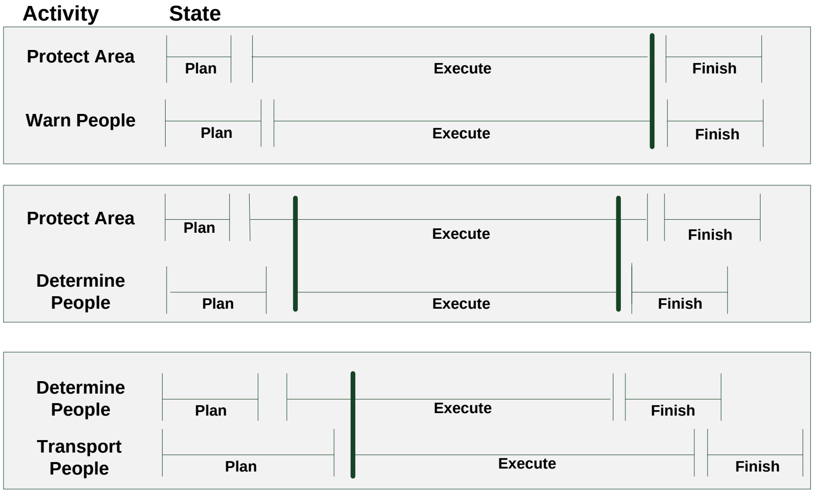

Figure 5 Activities and their relative temporal Dependencies

An activity is in a state for some time, but we cannot prescribe explicitly the time (e.g. we cannot prescribe that the activity “Protect Area” is in the state “Execute” for 5 hours and 5 minutes). The time an activity is in a certain state is illustrated in this figure as a line. There are three temporal dependencies between the activities. Each box in the figure describes one dependency between activities. The first box describes the dependency between states of the activities “Warn People” and “Protect Area”. The activity “Warn People” is first in the state “Plan”. Afterwards it changes into the state “Execute”. The activity can be as long in the state “Execute” as the activity “Protect Area” is in the state execute, i.e. if the activity “Protect Area” changes into the state “Finish” then also “Warn People” has to change into another state (e.g. “Finish” or “Fail” etc.). Otherwise the dependency is violated. This corresponds to the temporal dependency “Finished by”.

The second box describes the dependency between the states of the activities “Determine People” and “Protect People”. The activity “Determine People” can only be in the state “Execute” if also the activity “Protect People” is in the state “Execute”. This corresponds to the temporal dependency “Contains”. The third box describes the dependency between the states of the activities “Transport People” and “Determine People”. The activity “Transport People” in the state “Execute” overlaps with the activity “Determine People” in the state “Execute”. This means transports can start during execution of the activity “Determine People”. The activity “Transport People” can be in the “Execution” state a little bit longer than the activity “Protect Area”, because people are transported out of the area (i.e. they are not directly affected by this activity).

It is very difficult to represent and execute this in business process management systems (cf. related work), because this leads to very large models which have exactly an opposite effect: They make the situation even more complicated and end user tends in this case not to use such tools.

People in the field can exchange activities and dependencies with other people (e.g. their operating center or of other organizations) and they can also create dependencies to their own activities. Field teams usually have one person designated for communication and this person can use our system on a PDA or an even simpler input device to describe activities currently done by the team. This is not an unrealistic assumption, because we found out during our interviews that small simple input devices are already in use by some fire fighter teams in the field. They are used to transfer activity status to the operating center, but do not provide functionality to create dependencies between activities and to warn them when a dependency is violated. If we revisit the example in Figure 1 we might have the situation where one or more of the fire fighters are injured. Other fire fighters stop to build the dam to treat the injured fire fighters. This activity is exchanged (e.g. via email or other communication channels) with people in the command center and these people establish dependencies to other activities (e.g. “Send Ambulance”) and exchange these dependencies with people in the field. This illustrates also another potential of our system from the perspective of vertical coordination: The teams in the field can give better feedback on what they are doing and what the current situation in the field is to people in the command center. The same mechanism can also be applied to the horizontal/cross-organizational level: Fire fighters in the field can exchange activities and dependencies with police men in the field. Since activities and dependencies are lightweight constructs, there is no complex automated translation of concepts of the fire fighters to concepts of the police. We assume that police can make sense out of the activities of the fire fighter (i.e. understand the basic meaning). As a result from our interviews this is usually the case for activities. Case studies have shown that this is usually not a big issue in a disaster (cf. (Wachtendorf 2000)).

DISCUSSION

There are two different kinds of evaluations for our system: analytical and empirical. The analytical evaluation allows to make sure that activities and dependencies are correctly interpreted by the system and that no cyclical dependencies can be created between activities. For instance, analytical evaluation forbids that activity “C” follows after activity “B” and activity “B” follows after activity “A” and activity “A” follows after activity “C”. We ensured this by using methods of interval algebra (Allen 1983). This is also different for analytical evaluation of models in business process management systems, where the focus is the correctness of the control-flow of an isolated process.

We focus in this paper on the empirical evaluation of the system. This requires to interview end users, such as fire fighters or police. During these interviews we want to present them our model. As of now, we have been discussing the model with one senior (i.e. professor) and one junior organizational scientist specialized in firefighting procedures by providing an animated version of our model. Thus, the terminology we used for describing concepts (e.g. activity) is in line with the terminology of organizational scientists. Out of our discussions, they have evaluated our model from an end user perspective. They confirmed that the model from an activity and dependency level can be useful for first responders. They also confirmed that the execution mechanism is useful. They recommended integrating our model with existing visualization systems.

We also discussed with a commander of a large humanitarian aid organization in Germany and a fire fighter commander in France about the model using an animated presentation. Both have several years of leadership experience in the field with real disasters. They said the model can be useful for certain types of disasters, where it is possible to use information technology to coordinate each other (e.g. a storm or low pacing floods). For example, in very fast evolving situations, such as wild fires, it is not possible for people in the field to use information technology, because they have to answer fast to the situation. They told us that the model is useful on higher levels of the hierarchy of organizations (e.g. command center). Different levels of a hierarchy require different level of details on the activity model. For example, people in the field may only need a list of activities, but on higher levels more detailed models are required. Our model allows representing the activities at different hierarchy levels (cf. Figure 1) using, for example, the dependency “contains”. They also confirm that the model is useful for the cross-organizational perspective, but it requires that the organizations and different levels of a hierarchy communicate with each other. They mentioned also the issue of conflicting information on the cross-organizational level or even cross-team level. For example, different organizations perceive the same activity in different states (e.g. the police thinks that the activity “Build Dam” has failed while the fire fighters are still executing this activity). This leads to conflicts which need to be resolved and system support can be helpful for that (e.g. by defining governance mechanisms to resolve these conflicts).

This is an early validation of our model. We will test our system in the context of a disaster exercise, together with identified actors of the crisis management domain. We will evaluate our prototype throughout two exercises. In the first one, there will be a predefined detailed script, known by all participants. The goal within this exercise will be to familiarize users to prototype and its model, and to learn about its possibilities. In the second exercise, there will be a predefined high level script, known by all the participants. During the exercise, moderators will further expand the script with details, to make it more realistic. The goal here is to evaluate the capacity of the prototype for handle unpredictable scenarios (such as disasters). We also want to investigate the limitations of using the system with respect to the dynamics of fast evolving situation.

CONCLUSION

Our contribution in this paper is twofold. First, it contributes to the understanding of information system support for crisis response activities. A balance has to be found between agility and control and between simplicity and expressiveness. We found out, and it confirms other works, that a process centered approach is not well suited to get the right balance as it tends to be less agile and too complex. It appears that in parallel with existing resource centered or event centered approaches, an activity centered approach provides a good level of abstraction to cope with the specific nature of the crisis. Of course, this includes the ability to manage different kinds of dependencies between activities (e.g. temporal ones) and especially between activities of different stakeholders. The solution we propose is based on these conclusions and it is our second contribution. Our model is based on the idea of networks of activities that can be shared and controlled between different organizations. Our framework provides a definition of activities that may have a specific life cycle controlled by adapted-able governance models. This is new compared to process management where only the owner of an activity can change its state. Activities themselves can be considered as specialized micro processes with dependencies towards other activity micro processes, forming a network that should remain manageable. Constraints checking and enforcement policies ensure that activity management can be integrated smoothly with the actual work that takes place during the crisis. Although extended empirical validation has not been conducted yet, we have early indicators that this solution can be considered as an initial step towards a more sophisticated support of dynamic disaster response activities. Cross-organizational activity management is part of our future research, because there are still unsolved governance issues (e.g. conflicting views of different people/organizations on the state of an activity). We evaluated this system approach together with business process experts and did first evaluations with social scientists and end users. We are continuing the evaluation of our approach together with end users, which are also interested in evaluating the prototype as part of disaster exercises in the future. We anticipate that other business domains could require cross-organizational ad-hoc activities, such as the media industries, customer relationship management, the service industries or the critical infrastructure protection area.

Acknowledgement

The research was partially funded by the German Federal Ministry of Education and Research under the promotional reference 01ISO7009. Another part was funded by the French Agence Nationale de la Recherche (CIFRE funding scheme). The authors take the responsibility for the content.

REFERENCES

1. Allen, J. F. (1983) Maintaining Knowledge about Temporal Intervals, *Communications Of The ACM*, 26, 11, 832-843.
2. Catarci, T., de Leoni, M. et al. (2008) Pervasive Software Environments for Supporting Disaster Responses, *IEEE Internet Computing*, 12, 1, 26-37.
3. Chebbi, I., S. Dustdar, et al. (2006) The view-based approach to dynamic inter-organizational workflow cooperation, *Data & Knowledge Engineering*, 56, 139-173.
4. Denning, P. J. (2006) Infoglut, *Communications of the ACM*, 49, 7, 15-19.
5. Eshuis, R. and P. Grefen (2007) Constructing customized process views, *Data & Knowledge Engineering*, 64, 419-438.
6. Fahland, D. and H. Woith (2008) Towards Process Models for Disaster Response, *Process Management for Highly Dynamic and Pervasive Scenarios*.
7. Franke, J. and F. Charoy (2010) Design of a Collaborative Disaster Response Process Management System, *9th International Conference on the Design of Cooperative Systems*.
8. Georgakopoulos, D., H. Schuster, et al. (2000) *Managing* Escalation of Collaboration Processes in Crisis Mitigation Situations, *16th International Conference on Data Engineering*.
9. Llavador, M., P. Letelier, et al. (2006) Precise yet Flexible Specification of Emergency Resolution Procedures. *3rd International Conference on Information Systems for Crisis Response and Management (ISCRAM)*, Newark, NJ, USA.
10. Lu, R., S. Sadiq, et al. (2006) Using a Temporal Constraint Network for Business Process Execution. *17th Australasian Database Conference*.
11. McCready, S. (1992) There is more than one kind of Work-flow Software, *Computerworld*, November, 2.
12. Recker, J. (2008) Understanding Process Modelling Grammer Continuance - A Study of the Consequences of Representational Capabilities, School of Information Systems, Queensland University of Technology, Brisbane, Australia.
13. Rüppel, U. and A. Wagenknecht (2007) Improving Emergency Management by Formal Dynamic Process-modelling, *24th Conference on Information Technology in Construction*.
14. Scheer, A.-W. (2000) *ARIS - Business Process Modeling*, Springer.
15. Sell, C. and I. Braun (2009) Using a Workflow Management System to Manage Emergency Plans, *6th International Conference on Information Systems for Crisis Response and Management (ISCRAM)*, Gothenburg, Sweden.
16. SoKNOS. (2009) Service-orientierte ArchiteKturen zur Unterstützung von Netzwerken im Rahmen Oeffentlicher Sicherheit, http://www.soknos.de, retrieved 12.03.2009.

17. van Someren, M., N. Netten, et al. (2005) A Trainable Information Distribution System to Support Crisis Management, *2nd International Conference on Information Systems for Crisis Response and Management (ISCRAM)*, Brussels, Belgium.
18. Wachtendorf, T. (2000) Interaction Between Canadian and American Governmental and Non-Governmental Organizations During the Red River Flood of 1997, Disaster Research Center, University of Delaware.