Design of a Collaborative Disaster Response Process Management System

Jörn Franke, François Charoy

Design of a Collaborative Disaster Response Process Management System

Jörn Franke and François Charoy

Abstract We describe in this article a framework for disaster response process management. This framework can be used to develop information systems supporting those processes. It is grounded in several research approaches: literature research, case studies, end user interviews and workshops. We compare disaster response process management with business process management and argue why it is substantial different to it. Another main result of this comparison is that business process management technology, such as flexible workflow systems, are not suitable for disaster response processes. We propose an information system supporting disaster response processes based on our developed framework. Finally we present validation of the information system design and give outlook on our future research.

Introduction

Information Communication Technology (ICT) support for disaster management has recently drawn much attention in research due to the growing awareness of disasters all over the world. The European Union (e.g. within the FP7 research framework) and research agencies of different countries (e.g. Germany or France) have granted several research projects in this area. However, none of them explicitly deals with the management of the activities of different organizations in the disaster response with ICT support. Our research has confirmed that it is an important issue.

J. Franke (✉)
Public Security, SAP Research Center (SRC) Sophia Antipolis, 805, Avenue du Docteur Maurice Donat, BP1216-06254 Mougins, France
e-mail: joern.franke@sap.com

F. Charoy
LORIA-INRIA, Nancy-Université, BP 239-54506 Vandoeuvre-lès-Nancy Cedex, France
e-mail: charoy@loria.fr

Our interviews with fire fighters and police have also shown that current practices for managing activities without or only with unsophisticated ICT support, such as e-mail, have several flaws and current practices are criticized by all stake holders.

The goal of this paper is to propose an approach for a collaborative disaster response process management system. We will follow a design science research approach [1] to achieve this goal. Following this approach, in the next chapter, we propose a framework for disaster response process management. It must support the management of activities within the disaster response on an intra-and inter-organizational level. In the third chapter, we propose an information system supporting disaster response process management based on the framework developed in the previous chapter. In the fourth chapter, we describe evaluation methods for this information system and provide initial evaluation results. Finally, we conclude and give an outlook on our future research.

Framework for Process Management for Disaster Response Processes

Definition

The term disaster is not uniquely defined in the area of disaster management. It is also used synonymously with terms like emergency or catastrophe. We distinguish between these terms and follow their definition by [2]. An emergency is a routine and is part of the day-to-day live of a public safety organization (e.g. fire fighter fight a manageable fire in one house). Each organization involved in solving the emergency is clear about its tasks and how it works together with other organizations. Major concerns in an emergency are the people affected by the event.

A disaster is significantly different to an emergency. The organizations face new and unforeseen challenges. There are far more organizations involved than in an emergency and it is not always clear what are the dependencies between them. Activities of day-to-day emergency routines may become less important than activities to fight the disaster. Goals of the organizations change depending on the evolution of the disaster response. Planning is important, but plans may change arbitrarily and new plans have to be made and integrated with other plans and activities of the own and other organizations. Execution and monitoring of new and old plans as well as activities is challenging. We use the term crisis synonymously with disaster.

A catastrophe is characterized by a very heavy impact on the community and its infrastructure, in particular the communication infrastructure. The organizations and people are overwhelmed by the impact, communication and coordination is almost impossible. One example for a catastrophe is the nuclear bomb in Hiroshima. Our proposed disaster process management approach aims at disasters and is also touches the area of emergencies, because it can be used to mitigate the risk or to manage the case when an emergency evolves into a disaster.

Disaster process management has only recently drawn some attention, but this was usually driven from the technical side without few if not any foundation in the social sciences. Disaster process management can be described by the following lifecycle phases: mitigation, preparedness, response and recovery [3]. The mitigation phase deals with risk reduction, i.e. prevention or attenuating that a disaster can actually happen. Preparedness deals with planning and training for disaster response and recovery within one organization and or between more organizations. In the response phase different organizations with low and high dependency on each other fight the disaster and its consequences. The recovery phase is about debriefing from the response (learning and discussing about the response) and reconstruction of the social processes (e.g. build houses, relocation, funds etc.) of the affected communities. We focus here mainly on the processes in the response phase. In the next section we describe our research methodology for developing the framework.

Research Methodology

In the following subsections, we introduce our different research methodologies. We begin in the next subsection with the literature review, followed by interviews in the third subsection. In the fourth subsection we describe a workshop about cross-organizational aspects of disaster response process management. In the fifths subsection we describe our analysis of disaster response plans.

Literature Review

We searched the literature in the social sciences and business process management on the topic. The goals of the literature review were (1) to understand the domain (2) to confirm the problematic (3) to provide an interdisciplinary view on the problem. Core theoretical foundation is Weick's work on sense making [4]. Weick identified the same problem as we did in our talks with public safety organizations: The main problem is to know who does what and how the corresponding activities are related. We analyzed empirical results (e.g. [5]) and we also investigated selected case studies (e.g. [6–10]). We are not aware of any work that deals with disaster response process management from an information systems perspective. Business process management (cf. [11,12]) was chosen as a reference, because it addresses, among other things, the same problematic, but in another domain. This field also provides mature technology and tools to address this issue. Business process management is used for intra- and inter-organizational coordination of activities. The management discipline encompasses methodological and technological approaches [13]. Business process management follows the lifecycle of planning/change management, implementation and monitoring of business processes [14]. The focus is mostly on operational business processes [15] that are repeated a great number of times. Strategic or tactical processes require different

approaches. Most prominent technical approaches are process-aware information systems or workflow systems [16], where the operational business process is modeled, executed and monitored. Technological solutions based on these approaches have been proposed (e.g. [17–19]), but without any agreement or clear definition what disaster process management could means.

Interviews

We used the results of the interviews within the SoKNOS project for investigating them from the point of view of the problem mentioned before. The interviews were open and the goal was to find out what are the problems and challenges in the disaster and how it can be supported by ICT in general. A scenario of a flood has been developed based on previous disasters together with end users. A set of unified modeling language (UML) use cases have been developed that can be supported by various systems. Response activities have been modeled for getting an overview about the processes and activities during a disaster response. Interviews last usually 1–2 h. Interview partner have been fire fighter commanders and police commanders with several years of experience, because the focus of the project was the command center level.

Workshop

The workshop aimed at cross-organizational collaboration between command centers of different organizations. The workshop lasted 1 day. The workshop itself was open and the goal was to get the big picture of cross-organizational collaboration between public safety organizations in a disaster. It was attended by domain experts, e.g. fire fighter commanders from two different states in Germany and a police chief of one state in Germany. It was also attended by computer scientists and scientists from the information systems area. Topics covered in this workshop were: legal framework for collaboration between public safety organizations and its implementation in real disasters. We covered several real flood disasters where different public safety organizations worked together. The output was a global picture how public safety organizations, in particular police and fire fighter work together within one state and different states in Germany. The results are presented in the next section.

Plans of Organizations

During our research we had the opportunity to look at plans of fire fighters of two different states in Germany for responding to a disaster. Although both organizations have the same purpose (fire fighting), they are highly independent and follow different methods, regulations and procedures, because states in Germany are highly independent. These plans cover generic disaster activities (e.g. evacuation or treatment of injured people), but also disaster specific activities (e.g. for a pandemic).

Disaster Response Process Management

Overview

In this section we present our research results about disaster process management in the response phase. We describe challenges in modeling, executing, monitoring and cross-organizational aspects of response processes. We compare them in each corresponding subsection with business process management and summarize the results in the fifth subsection. Finally, we describe in the last subsection requirements for a disaster response process management system. These requirements can be used as a basis to develop such a system.

Modeling Processes

A first step has been to challenge business process modeling languages, using them to model disaster response processes. The main rationale behind this was that systems supporting management of processes require description of the processes as business process models. We tried to model together with end users, based on plans and their experience, the response processes for a flood and a train accident with hazardous material. We used the event-driven process chain (EPC) modeling language. We did not investigate other business process modeling languages, because research has shown that business process modeling languages are understood as similar by the different users [20]. The main problem was that most activities in the resulting plans are executed in parallel and different temporal relationships exist between activities which cannot be modeled adequately by business process modeling languages. For example, a business process modeling language cannot articulate that two activities have to be executed at the same time or that they should overlap. Business process modeling languages only support data dependencies between activities, but to very limited extent temporal dependencies (e.g. it is only possible to describe sequences of activities). In fact, the modeled processes were useless, because they just illustrated that many activities had to run in parallel.

We also tried to consider governance/management aspects of activities, but the modeling of these aspects was limited using a business process modeling language. For example, in a business process modeling language, it is only possible to model who is responsible for executing an activity, but we also needed to model other mechanisms (governance), such as only an accountable person is allowed to change an activity from a planning phase to and execution phase (i.e. giving an order for execution). When including this, resulting process models where very large (i.e. many activities and connection between activities) and could not be understood without extra explanation by the modeler. The police chief commented the models as follows:

> This is just for you (the interviewers) to get an overview what happens. It does not make much sense to represent it like this. We better do not start modeling the exceptions in this…

We also did not find use of business process modeling or similar languages (e.g. flow charts) in planning documents. Our experience has also been confirmed by others (e.g. [18,19,21]). Modeling or planning is also done in real disasters. Usually, the organizations use whiteboards or geographical maps to model/plan their actions to respond to a disaster. They rely at least initially on existing plans for responding disasters. These plans are written documents, where activities are listed. Sometimes also background information is given (e.g. scientific reasons).

Execution

Once the disaster has happened, a lot of response processes are executed by different organizations. Each organization establishes a command center managing its response. Depending on the dimension of a disaster, more than one command center might be established by one organization (e.g. one for each disaster site). A command center controls one or more field teams, which are themselves coordinated by a team leader. Usually the field teams have the right to act autonomously in case of imminent danger or threat of life (i.e. it is not comparable with a military command and control model). The different planned activities are delegated to the responsible people for execution. Accountability and responsibility (governance) for an activity are usually clear in emergencies, because each organization has clearly defined task. However, in a disaster this is not always the case. Accountability or responsibility may become unclear for several reasons, e.g. organizations have to do new tasks, because disasters have by definition new and unforeseen challenges. Another consequence of this is that availability is unclear, because organizations may be busy with new and unforeseen tasks, e.g. if some of the field team members are injured then rescuing them has priority over responding to a disaster (e.g. continue fighting a fire).

Workflow technology (cf. [15]) enables the controlled execution of operational business processes by means of an information system. It cannot be used to execute disaster response processes. Firstly, it relies on business process models, which cannot be used to model adequately response processes as mentioned before. Secondly, it enforces the execution of the processes, because business processes need to process information (e.g. invoices) in a standardized way. This does not hold for a disaster response processes: The focus is here on detecting violation of dependencies to other activities in order to deal with them adequately. Thirdly, business process models do not provide enough flexibility for change, although several change approaches have been proposed in the literature [22–26]. These approaches only apply where change is the exception and not the rule. During a disaster, change and adaption is the rule. Fourthly, the execution semantics of a workflow do not apply to a disaster response process. Each process execution (workflow) in the workflow system manages information and sequential dependencies between activities (e.g. an invoice need to be received before it can be processed). Disaster response processes do not have information dependencies between the activities, but temporal dependencies, which need a different kind of treatment. The reason

for this is the inadequate description of response processes by a business process modeling language. Hence, it is very difficult to execute disaster response processes in a workflow system. The disaster response process execution in such a system would be meaningless.

Monitoring

The command center of an organization receives feedback, i.e. the state of an activity, from the field. Different organizations have different means to visualize this feedback. Some organizations, such as fire fighters, use geographical maps to present this information. Others, such as the police, use a matrix to present this information. This means they use the same tools for monitoring and modeling/planning. At the moment, they don't always have information technology support, although this is a desired feature, because using printed maps or whiteboards is quite cumbersome and information can get lost or there may be information overload.

We have argued before that execution of disaster response processes in a workflow system is meaningless and thus, monitoring of them in the same system is also meaningless. Monitoring in a workflow system is used to ensure that the processed business objects by a workflow meet business goals and key performance indicators. Disaster response processes do not have business objects and business goals are different from goals of a disaster response.

Cross-organizational Aspects

During our workshop, we discovered the importance and the multi-dimensional aspects of cross-organizational management.

There are three different kinds of collaborations between organizations:

1. Activities of one organization depend on many other activities of other organizations, i.e. temporal dependencies as mentioned already in the modeling section. This came up not only in the workshop, but also in other case studies (e.g. [10]) and interviews. It should be noted that, although activities are dependent on each other, there is no globally defined process by the different organizations. The organizations are highly independent and they deal with activities and dependencies to activities of other organization in a decentralized manner, i.e. they deal with them themselves.
2. One organization may work as a contractor for another organization.
3. One organization becomes part of the other organization, i.e. part of the organizational structure.

The problem of accountability and responsibility becomes also a new dimension on the cross-organizational level, for example, when fire fighters of different states or even countries fight together a fire. There can be a mixture of all three kinds of collaborations, which leads to unforeseen problems if accountability and responsibility

is not describe properly. Security and trust was an important matter when working on the cross-organizational level. Security refers to ensuring confidentiality and integrity of information when transmitting, storing and processing them. Trust is about how much people in one organization trust people of other organizations. This affects information sharing between them. In our workshop it was noted that trust may even change during a disaster. For example, people, who have been trusted before are not trusted anymore, because of an event destroying their reputation.

Inter-organizational operational business processes can be modeled using workflow technology (e.g. [27,28]). They are suited for operational business processes and similar arguments to the ones given above apply also to them. Cross-organizational business processes have usually few well-defined complex organizational and technical interfaces between them. Those interfaces are well-designed for the given purpose. One global processes in which all organizations are part of is designed and all organizations agree to this process (cf. supply chain management). The opposite holds for cross-organizational disaster response processes: There are many simple interfaces, which are created ad-hoc. They are not well-designed for the given purpose, but functional (e.g. using phone, e-mail or fax instead of complex communication system).

Comparison

Disaster response process management is different than business process management. In the Table 1 we summarize important differences. We chose the following criteria for comparing them: Management Lifecycle, Modeling, Execution, Monitoring and cross-organizational aspects. Our experiences with end users confirm that these differences require a new approach to process management and its technology support. It should be noted that we do not only aim at an adequate description of processes, but also at their management (i.e. execution and monitoring).

Requirements

In this section we describe the main requirements for a collaborative disaster process management system based on our research described in the previous sections. These requirements are the following:

- It shall allow simple modeling (i.e. without complex constructs) of the response activities on a shared activity workspace. Humans have problems to understand complex models [29] in real time and a quick understanding of activities and dependencies is required in a disaster response.
- It shall allow the modeling of activities: Activities can be created ad-hoc because new activities, which have not been done before, will occur. Different types of activities (e.g. decision-making activity, evacuation or search and rescue) need

Table 1 Comparison between Business Process Management and Disaster Process Management

Criteria	Business Process Management	Disaster Process Management
Process Management Lifecycle	Planning, Implementation and Monitoring of processes are sequential steps, each taking a lot of time	Planning, Implementation and Monitoring of processes in parallel, no start and end of these steps, highly iterative steps
Modeling	Control-flow oriented: complex routing of information between activities. Processes can be managed in isolation to each other	Temporal dependencies between activities. Processes cannot be managed in isolation to each other
Execution	Frequently, few exceptions, change is seldom	Executed seldom/once, many exceptions, change is the rule
Monitoring	Key performance indicators and business goal violation	Activity status and violation of temporal dependencies
Cross-organizational Aspects	Global definition of inter-organizational processes, few interfaces/interactions between organizations (organization to organization)	No global definition of inter-organizational processes, many interfaces/interactions between organizations (people to people), ad hoc definition of new interfaces/interactions based on personal contacts

to be modeled differently, because there are different management processes for them. For example, activities in the field have a different management process compared to strategic activities.

- It shall allow the modeling of different kinds of dependencies: different kind of temporal dependencies can be established between activities (e.g. activities should be in the same state at the same time). This is the result of our research and currently these kinds of dependencies are not supported by process management systems.
- It shall allow the modeling of governance roles: organizations tend to be involved in networked organizational structures. This requires a clear description and enforcement of governance roles for an activity. This is essential and is still a big issue in current disaster response processes.
- It shall support the execution of activities: Execution is described as state change of activities. State changes of all activities can take place concurrently. In real case, all the activities may run in parallel. Each state change may violate dependencies and this needs to be managed by the system (e.g. by visualizing them). This supports the understanding of the disaster response processes.
- Monitoring of shared activity workspace: 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 monitoring of the response processes. If they are not provided by the system, the end users will not accept it.

- Exchange of activities and dependencies: Activities and dependencies can be exchanged between different shared activity workspaces within and outside the organization. Exchange always takes place between people based on prior private or work-related contacts and not between organizations. Subsequent state changes of the exchanged activities needs to be propagated to all shared workspaces the activity has been exchanged with. Our research has shown that organizations are highly independent, but still they have to exchange these kinds of information. They do not plan processes in a collaborative manner, but keep other people informed about their own activities and integrate the status of activities of other organizations in their own processes.

Based on these extensive studies and requirements elicitation, we have started to work on a model that should answer these requirements. This model is centered around the concept of activities and on their dependencies.

Concept for a Disaster Response Process Management System

Overview

In this chapter we are going to describe the fundamental concepts of the disaster response process management system. We start in the next subsection with a modeling language for describing disaster response activities. A meta-model for the language is described and afterwards an example is given, which explains the elements of the meta-model. In the second subsection, we describe how the model is enacted by the system. In the third subsection, we describe how we can facilitate monitoring of the activities beyond the execution mechanism. In the last subsection we describe how activities can be exchanged between shared activity subspaces to enable cross-organizational process management.

In contrast to business process management systems, our system supports modeling, execution and monitoring at the same time in an integrated manner as it is part of the requirements we have identified.

Modeling

Overview

Our research has shown that business process modeling languages are not able to model disaster response processes adequately. The modeling approach we propose is based on the meta-model described in the next section. It is followed by an example of this modeling approach.

Meta-Model

The meta-model includes the following model elements: activity type, activity and dependency. Activities have an activity type. Activity types are used to model the different kind of activities, their life cycle and their governance rules. All activities may run in parallel as soon as they are created and dependencies can be established between them during their execution.

An **activity type** $at_i = (SA, f, G)$ is described as follows:

- S is a finite set of activity states.
- $SA \subseteq S$ is a subset of activity states for the activity.
- $f : SA \rightarrow SA$ is a transition function defining the possible transitions from one state to another for one activity type.
- $G = \{g_a, g_r, g_c, g_i\}$ describes four governance roles (accountable, responsible, consulted and informed) and their transition functions for changing an activity state. These governance roles can be found in various contexts, such as military or project management [30].
- The idea is that depending on the role only a certain subset of transitions can be made.

  - $g_a \subseteq f$ is the transition function of the accountable role for the activity. Accountability describes who decides ultimately on the activity and also the governance arrangements.
  - $g_r \subseteq f$ is the transition function of the responsible role for the activity. Responsibility describes who executes the activity.
  - $g_c \subseteq f$ is the transition function of the consulted role for the activity. Consulted describes who should be consulted prior a state change.
  - $g_i \subseteq f$ is the transition function of the informed role for the activity. Informed describes who is informed after a state change.

An **activity** $a_i = (uid, name, cs, ad, GA)$ is defined as

- *uid* is a unique identifier of the activity
- *name* describes the activity
- $cs \in SA$ is the current state of the activity
- $ad \in A_t = \{at_1, ..., at_n\}$ one activity type in the set of activity types
- $GA = P \times G$
- P is the set of assigned participants/users

The activity description can be extended by further data (e.g. resources, geographical positions etc.), but we do not define how the data should be structured or interpreted.

Dependencies can be defined between states of two activities. Here, we limit ourselves to temporal dependencies.

They have been identified as very important in the requirements phase. These dependencies are not based on absolute or concrete date/times, but on relative ones (i.e. relative to states of other activities). The main reason is that absolute date/
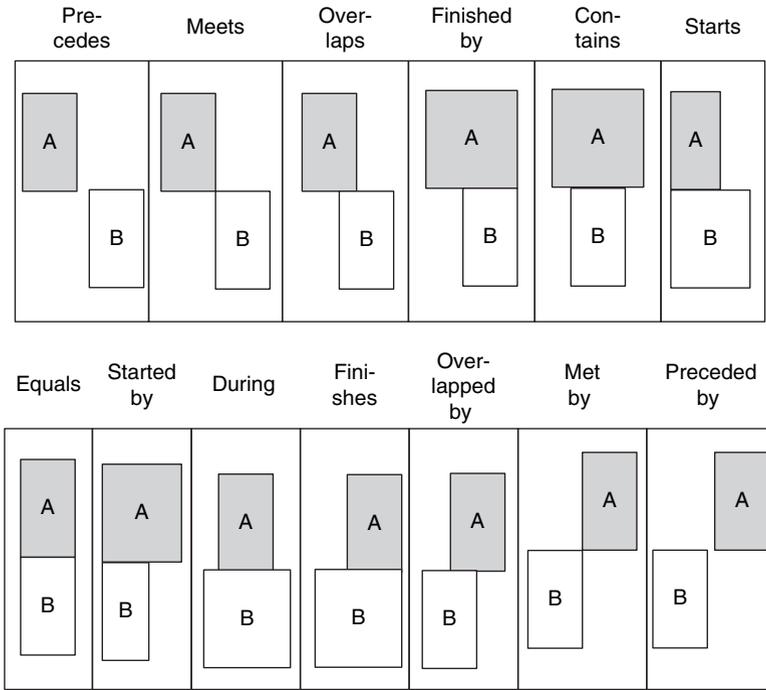
Fig. 1 Allen's proposed time interval relationships

time dependencies are difficult to plan or predict in a disaster response. However, this does not mean we cannot integrate deadlines in our model. We chose Allen's time interval relationship theory [31] as a foundation of our dependency model. Allen described 13 time interval relationships (see Fig. 1), we also use for describing dependencies between states. These dependencies are distinct, exhaustive and qualitative. This is different from business process models and provides much more flexibility. Several dependencies may exist between different states of two activities.

All these elements can be loaded from and stored into a file (similar to a response plan).

Example

Activities and dependencies can be created by users on a shared activity work space at any point in time. The model doesn't distinguish between design, execution or monitoring phase. Activities are based on activity types. Examples for activity types are decision-making activities or operation in the field. Each activity type has its own management lifecyle (see Fig. 2 for an example).
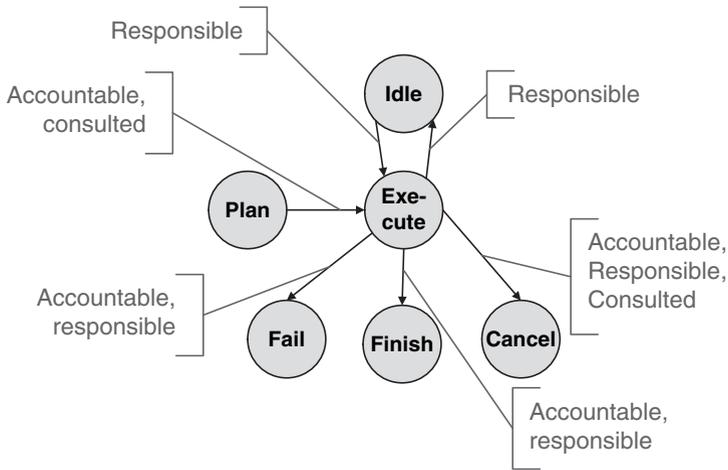
Fig. 2 Example: Activity type

As we have said before, the model is created and modified as it executes. Execution here means changing state of activities and taking care of dependencies. Therefore, we provide an example for modeling and executing activities in the next section.

Execution

Algorithm

Execution is about changing the state of activities. State change of activities has an impact on the dependencies between them. The execution algorithm can be described as follows:

1. State change of activity is requested
2. Verify if state change is allowed by governance role
3. Create a list of violated dependencies by the state change

All violated dependencies are returned by the execution algorithm. We do not describe the technical implementation here, but it has been already implemented. The system has three choices to treat the violation of dependencies: not allowing the state change (enforcing the dependency), visualization of the violation of dependencies (support) or trigger the required state changes of other activities to fulfill the dependency (automation). The treatment of dependency can be modeled together with the dependency in our system, e.g. the user defines if the dependency should be enforced, supported or automated. Selecting one or the other dependency

violation management strategy is very dependent on the kind of activity and on the situation. More experimentation will be needed to understand what the best practices are. However, the need to provide very flexible management mechanisms is mandatory. Especially, always enforcing the kind of constraints that we allow to implement would certainly conduct to unproductive situation.

Each change is also recorded in the execution trace. The execution trace also records the governance role and the participant who performed the state change.

Example

Figure 3 illustrates an example for executing two activities with a dependency between the state "execution" of each activity. The dependency says that both activities have to be in the execution state at the same time. It means that the beginning and the end of their execution should be synchronized.

In the first phase, both activities are in the state "Plan". The dependency is not violated, because no activity is in a state described in the dependency. The responsible role for the activity "Build dam" switches to the state "Execute" in phase two. The systems warns the participants in the shared activity workspace that there is a dependency conflict: if the activities "Build Dam" switches into the state "Execute", the activity "Transport Sandbags" has also to switch into the state "Execute" and vice versa. In the third phase, the responsible role for the activity "Transport Sandbags" switches into the state "Execution" and the conflict is resolved.

Monitoring

Core concept of monitoring is that the participants are informed about the current state of an activity and of violated dependencies. This mechanism can be extended by visualizing the activities in different contexts based on the attached data. For example, one popular visualization method for organizations in the disaster response is to visualize the activities on the map. This is illustrated in Fig. 4. In this example, a flood of the river Seine is threatening several suburbs in Paris. Different organizations try to build a sandbag wall to protect the suburbs against the flood. The activity "Transport Sandbags" failed, because a truck broke down.

Decentralized Exchange

Overview

In this section we describe how our approach works on the cross-organizational level. The basic idea is that there is no globally defined process. People within one shared activity workspace exchange selected activities and dependencies with
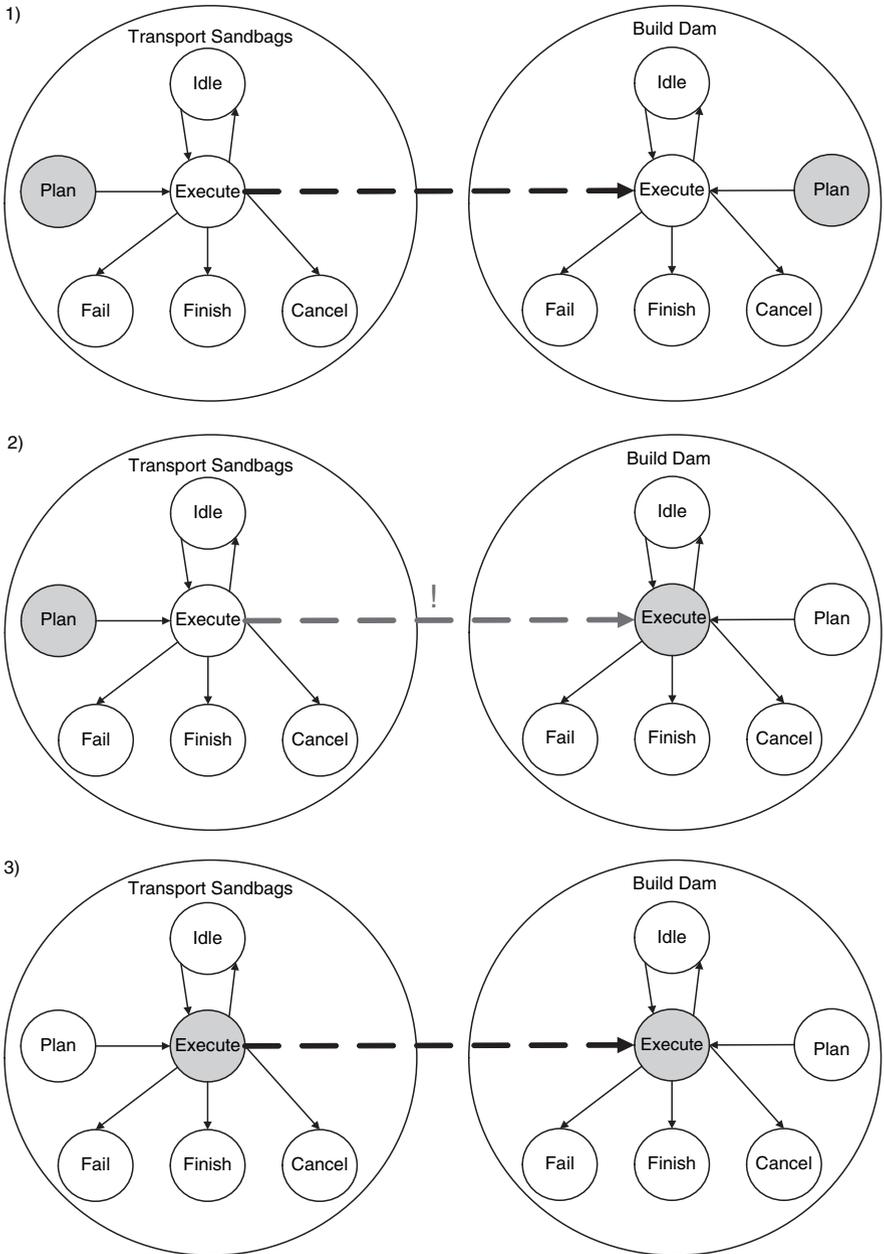
Fig. 3 Example: Execution of activities

people of other shared activity workspace. For instance, the police chief, responsible for the shared activity workspace of the police exchanges this information with the fire fighter chief responsible for shared activity workspace of the fire fighters.
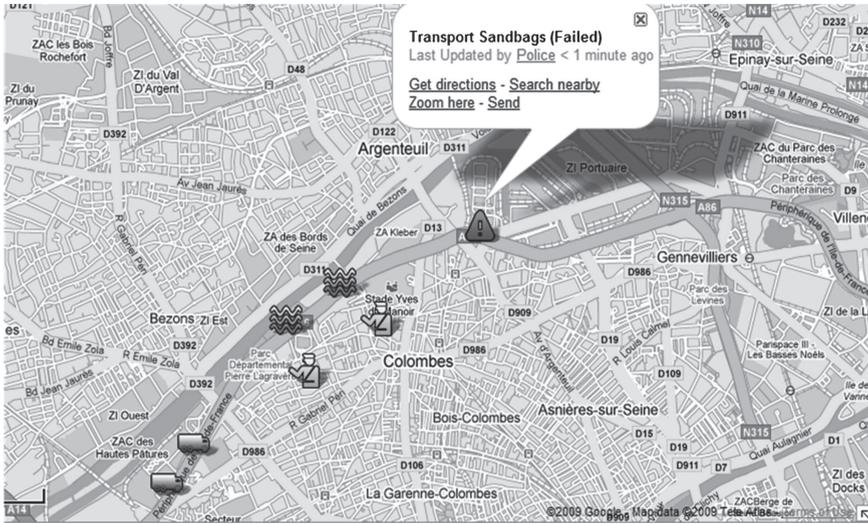
Fig. 4 Example: Monitoring

Each of them can define new dependencies of their own activities with the shared activity ones. Shared activity workspaces are not managed by a centralized server. Thus, appears the problem of the synchronization of activity states when they are shared between different activity workspaces.

In the following sections we describe a protocol for exchanging activities and dependencies. Afterwards, we describe how modeling and execution are affected by exchanging of activities and dependencies. Finally, we describe change propagation of activity state changes.

Protocol

In this section we describe how activities and dependencies can be exchanged between shared workspaces. This can be another shared workspace within one organization, but also a shared workspace of another organization. The exchange is initiated between people based on personal contacts. This is the desired method to exchange information as we have found out in our interviews and workshops. For example, the chief of the police pushes some activities and dependencies to the chief of the fire fighter. Pushing of this information can be done by various communication protocols (e.g. e-mail). This is similar to the current situation, but with the support of our more sophisticated process management approach. Each exchange follows the following process:

- Participant P of shared activity workspace X sends selected activities A_i and dependencies D_i to participant m of another shared activity workspace Y

- Participant m receives the activities A_i and the dependencies D_i and decides which activities $AS_i \subseteq A_i$ and dependencies $DS_i \subseteq D_i$ he/she wants to add to the shared activity workspace

Modeling

Exchanged activities and dependencies need to be represented in the shared workspace. Exchanged dependencies and activities are modeled the same way like not-exchanged ones. Dependencies can be defined arbitrarily between non-exchanged and exchanged activities.

Execution Support

As mentioned before, execution is defined as state change of an activity. State changes of exchanged activities are treated in the same way as state changes of non-exchanged activities. An important point is that execution may have different local effects, because in different shared workspaces different dependencies can be established to the activity. This may lead to violation of dependencies in some activity workspaces. These violations have to be resolved like any other violation (e.g. by changing the state of an activity, by removing the dependency, by communicating with people from the other shared workspace or by waiting).

Change Propagation

Exchanged activities may change their state after the exchange. This state change needs to be propagated. We distinguish between two cases for change propagation, which are based on current practices on how information is exchanged between organizations in a disaster response:

- If the state change occurred in the own shared activity workspace then it has to be propagated to all organizations the activity has been forwarded to (e.g. the fire fighter inform the police about canceling the activity "Protect residential area from flood").
- If the state change occurred in another shared activity workspaces and has been received to update an exchanged activity, it has to be propagated to all shared activity workspace the exchanged activity has been forwarded to (e.g. the police informs the red cross about the cancellation of the activity "Protect residential area from flood" by the fire fighters).

We have identified the following problem cases when propagating change in the shared activity workspace:

- Missing updates: An organization may not receive all updates, because it has been disconnected or other organizations, responsible for propagating the change, have been disconnected.

- Incomplete Propagation: State changes may never be propagated to all parties involved or different shared activity workspaces may have different states of the same activity.
- Conflicting Updates: State changes can be conflicting, for example if, an activity is changed in more than one shared activity workspace to different states. This leads to conflicting updates.
- Ensuring governance on the cross-organizational level (i.e. who is allowed to do state changes).

These problems may also occur by using the traditional methods as outlined above. We have developed several mechanisms for avoiding or mitigating these conflicts, which go beyond the scope of this paper. They are currently evaluated by researchers and end users.

Example

In this example we illustrate the exchange mechanism. Figure 5 provides an example for exchanging activities and dependencies between different shared activity workspaces. In this example the police exchanges activities and dependencies with the fire fighters in the first step. The fire fighters integrate them in their shared activity workspace and create new dependencies to their own activities in the second step.
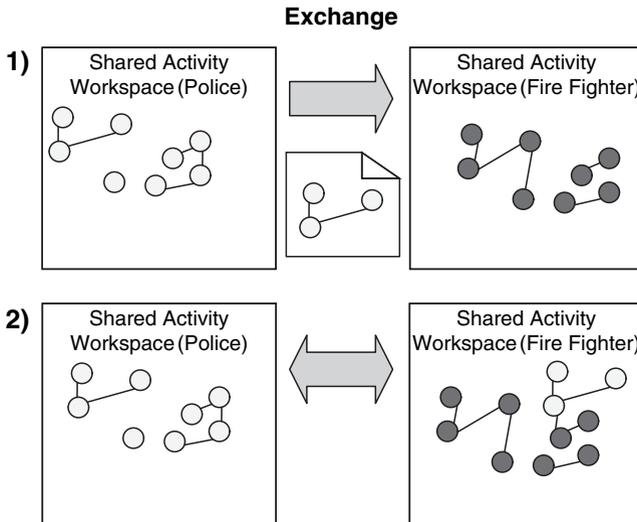

Fig. 5 Example: Exchange

Discussion

Our concept meets the requirements mentioned before. It provides support for modeling, execution, monitoring and cross-organizational aspects of disaster response processes. It should be noted that it just represent one possible solution. We discussed with three business process experts from our lab, not involved in our project, our approach based on a given flood scenario. The experts modeled some response processes based on our solution. A paper prototyping approach has been chosen for this, because at that time our prototype is not usable. We explained them our model approach and based on this approach they drew the activities and dependencies on a piece of paper. The experts agreed that this modeling approach is more suitable to model disaster response activities. They were concerned that, although the models are much more simpler compared to business process models, they may also get complex, because in a disaster there are usually many activities. This can be limited in an implementation by providing filters by criteria, so that only relevant activities are shown. Other critics are based on the limitation of the paper-based approach (drawing takes time, mistakes cannot be easily made undone etc.). We provided some examples of execution of the models based on the models the experts created. They recognized that warning of dependency violation is helpful in those scenarios. We did not evaluate cross-organizational aspects, because they do not really change modeling, execution and monitoring. We are aware that these are just a limited validation, but it shows us that we are heading in a good direction.

Conclusion

Our contribution in this paper is threefold. First, we proposed the foundation of process management model for disaster response that can be used to build information systems supporting disaster response process management. Such a foundation is strongly required, because existing means have their limitations and people from the technical community have little knowledge about disaster response processes. The opposite holds for the disaster domain perspective: They are not aware of possible technological solutions. Secondly, we have described a new approach for managing processes in the disaster response, because our research has shown that existing technology for operational business processes does not fit with this kind of scenario. Our approach encompasses modeling, execution and monitoring of disaster response processes on the intra- and inter-organizational level. Central concepts of our approach are activities on multiple shared activity workspaces and exchange of activity information between them. We strongly believe that we can contribute with this to the recently opened discussion on alternative process modeling and management approaches (e.g. [32]). Thirdly, we described initial validation of our approach by discussion with experts. We are aware that this validation is only a small first step towards a more sophisticated validation, but we think that

even this little validation initially confirms that we are on the right track. We plan to have a similar test of our approach with end users in the near future based on an implemented prototype to avoid limitations of the paper-based approach. We plan to test our approach with different end users, e.g. police and fire fighters. Feedback will be used to improve the concept and the prototype. Finally, we plan to test the prototype in real disaster exercises. Our end users are very open for testing the prototype in a real exercise. We plan two different kinds of exercises. One exercise will have a predefined story, which is known by all participants. This exercise resembles more an emergency than a disaster. Goal of this exercise is to understand the prototype and to learn about different possibilities to model a situation. In the second exercise we will also use a script, but only the high level part of this script is known by the end users who use the prototype. The detailed parts are developed by other end users during the exercise to make it more realistic. With this exercise we want to evaluate how well the prototype works in not predictable scenarios, such as disasters.

References

1. Hevner, A.R., S.T. March, and J. Park, *Design Science in Information Systems Research.* Management Information Systems Quarterly (MISQ), 2004. **28**(1): 75–105.
2. Quarantelli, E.L., *Catastrophes Are Different From Disasters: Some Implications for Crisis Planning and Managing Drawn From Katrina*. 2005, Disaster Research Center (DRC), University of Delaware, Delaware.
3. Choi, S.O., *Emergency Management: Implications from a Strategic Management Perspective.* Journal of Homeland Security and Emergency Management, 2008. **5**(1): 1.
4. Weick, K., *Making Sense of the Organization*. 2000: Blackwell, Oxford.
5. Drabek, T.E., *Strategies for Coordinating Disaster Responses*. 2003: Institute of Behavior Sciences, University of Colorado, Colorado.
6. *Beyond September 11th – An Account of Post-Disaster Research*, ed. J.L. Monday. 2003: Institute of Behavioral Science, Natural Hazards Research and Applications Information Center, University of Colorado, Colorado.
7. GovernmentSaxony, *Bericht der Sachsischen Staatsregierung zur Hochwasserkatastrophe im August 2002 (Report of the government of Saxony on the flood disaster in August 2002)*. 2003, Freistaat achsen.
8. Larsson, S., E.-K. Olsson, and B. Ramberg, *Crisis Decision Making in the European Union*. 2005: Crisis Management Europe Research Program, Stockholm.
9. Townsend, F.F., *The Federal Response to Hurricane Katrina – Lessons Learned*. 2006, The White House, Washington, DC.
10. Wachtendorf, T., *Interaction Between Canadian and American Governmental and Non-Governmental Organizations During the Red River Flood of 1997*. 2000, Disaster Research Center, University of Delaware, Delaware.
11. *Process Management*, ed. J. Becker, M. Rosemann, and M. Kugeler. 2003: Springer, Berlin.

12. Hammer, M. and J. Champy, *Reengineering the Cooperation: A Manifesto for Business Revolution*. 2001. Nicholas Brealey, London.
13. van der Aalst, W.M.P., A.H.M. ter Hofstede, and M. Weske. *Business Process Management: A Survey*. In *1st International Conference on Business Process Management*. 2003. Eindhoven, The Netherlands.
14. Zur Muehlen, M., *Organizational Management in Workflow Management – Issues and Perspectives.* Information Technology and Management Journal, 2004. **2004**(3): 271–291.
15. Dumas, M., W.M.P. van der Aalst, and A.H. ter Hofstede, M., *Introduction*, in *Process-Aware Information Systems*, M. Dumas, W.M.P. van der Aalst, and A.H. ter Hofstede, M., Editors. 2005, Wiley Interscience: Hoboken, NJ.
16. Dumas, M., W.M.P. van der Aalst, and A.H.M. ter Hofstede, *Process-Aware Information Systems*, M. Dumas, W.M.P. van der Aalst, and A.H.M. ter Hofstede, Editors. 2005, Wiley, New York, pp. 3–21.
17. de Leoni, M., M. Mecella, and G. de Giacomo. *Highly Dynamic Adaptation in Process Management Systems Through Execution in Monitoring*. In *International Conference on Business Process Management Systems*. 2007. Brisbane, Australia.
18. Fahland, D. and H. Woith. *Towards Process Models for Disaster Response*. In *Process Management for Highly Dynamic and Pervasive Scenarios*. 2008. Milan, Italy.
19. Georgakopoulos, D., et al. *Managing Escalation of Collaboration Processes in Crisis Mititgation Situations*. In *16th International Conference on Data Engineering*. 2000. San Diego, CA.
20. Recker, J., *Understanding Process Modelling Grammer Continuance – A Study of the Consequences of Representational Capabilities*. 2008, School of Information Systems, Queensland University of Technology, Brisbane, Australia.
21. Denning, P.J., *Infoglut.* Communications Of The ACM, 2006. **49**(7): 15–19.
22. Dadam, P. and M. Reichert, *The ADEPT Project: A Decade of Research and Development for Robust and Flexible Process Support – Challenges and Achievements.* Computer Science – Research and Development, 2009. **23**(2): 81–97.
23. Grigori, D., F. Charoy, and C. Godart. *Anticipation to Enhance Flexibility of Workflow Execution*. In *Database and Expert Systems Applications (DEXA'2001)*. 2001. Munich, Germany.
24. van der Aalst, W.M.P., et al., *Proclets: A Framework for Lightweight Interacting Workflow Processes.* International Journal of Cooperative Information Systems, 2001. **10**(4): 443–481.
25. van der Aalst, W.M.P. and M. Pesic. *DecSerFlow: Towards a Truly Declarative Service Flow Language*. In Lecture Notes in Computer Science, *Web Services and Formal Methods*. 4184: 1–23, 2006.
26. van der Aalst, W.M.P., M. Weske, and D. Grünbauer, *Case Handling: A New Paradigm for Business Process Support.* Data & Knowledge Engineering, 2005. **53**: 129–162.
27. Montagut, F., *Pervasive Workflows – Architecture, Reliability and Security*. Computer Science and Networks, Ecole Nationale Supérieure des Télécommunications, 2007. Sophia Antipolis, France.
28. Schulz, K.A. and M.E. Orlowska, *Facilitating cross-organisational workflows with a workflow view approach.* Data & Knowledge Engineering, 2004. **51**(1): 109–147.
29. Miller, G.A., *The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information.* Psychological Review, 1956. **63**(2): 81–97.
30. Smith, M.L. and J. Erwin. *Role & Responsibility Charting (RACI)*. 2007 [cited 03.06.2009]; Available from: http://www.pmforum.org/library/tips/pdf_files/RACI_R_Web3_1.pdf.
31. Allen, J.F., *Maintaining Knowledge about Temporal Intervals.* Communications Of The ACM, 1983. **26**(11): 832–843.
32. Fahland, D., et al. *Declarative versus Imperative Process Modeling Languages: The Issue of Understandability*. In *1st Workshop on Empirical Research in BPM (ER-BPM'2009)*. 2009. Ulm, Germany.