Modeling Business Processes for Automated Crisis Management Support: Lessons Learned

Elena Kushnareva
Centre de Recherche en Informatique,
Université Paris 1, Panthéon-Sorbonne
90, rue Tolbiac, 75013, Paris, France
Elena.Kushnareva@malix.univ-paris1.fr

Irina Rychkova
Centre de Recherche en Informatique,
Université Paris 1, Panthéon-Sorbonne
90, rue Tolbiac, 75013, Paris, France
Irina.Rychkova@univ-paris1.fr

Bénédicte Le Grand
Centre de Recherche en Informatique,
Université Paris 1, Panthéon-Sorbonne
90, rue Tolbiac, 75013, Paris, France
Benedicte.Le-Grand@univ-paris1.fr

***Abstract*—Process-aware information systems (PAIS) supporting knowledge-intensive processes are gaining importance nowadays. Crisis management process is an example of a knowledge-intensive process that is grounded on vast experience of multiple actors (e.g., city services, volunteers, administration) and their collaboration. Automated crisis management systems have to comply with various norms and regulations; at the same time, they need to constantly deal with uncertainty and adapt the process scenario to a current situation.**

In this paper, we consider the example of a flood management process implemented as a part of the COS Operation Center (COSOC) - a smart city solution that works with knowledge-intensive processes. We examine how the activity-oriented modeling paradigm underlying COSOC supports process flexibility. We propose an alternative way to specify the flood management process based on state-oriented paradigm and the statecharts formalism and discuss the advantages and limitations of the two paradigms.

Keywords—business process modeling; crisis management processes; activity-oriented modeling formalism; state-oriented modeling formalism; BPMN; statecharts.

I. INTRODUCTION

Crisis management is the process followed by an organization to deal with a major event that threatens to harm the organization itself, its stakeholders, or the general public. [1].

City administrations are particularly concerned with crisis management. Examples of crisis city may have to deal with include natural disasters (earthquakes, floods, landslides etc.), technological accidents (e.g., power plant accident, among others.

Crisis management process is safety-critical – its failure could result in loss of life, significant property or environment damage. To ensure safety and security, the activities performed during crisis management are highly regulated at the city or federal level. These activities and their order of execution are described in operation scenarios, procedures, emergency plans, which are used by the concerned public services (fire fighters, rescue, police, etc.) for regular drills and field trainings. According to the situation, a concrete scenario is selected from a predefined set: to assess the current situation, a number of critical parameters is taken into consideration (e.g., traffic condition, water level). Unforeseen situations, i.e., the situations that are not or only partly covered by the predefined scenarios and procedures, are handled by human actor(s): in simple cases, the process manager is authorized to define and launch a new workflow based on her experience; in more complex situations, which require a higher level of expertise, a decision is made by a specially assigned committee (e.g., a board of experts).

Modern city administrations seek to automate crisis management, implementing it as a part of their process-aware information systems (PAIS). In the following, we examine the characteristics of crisis, in order to define the requirements for PAIS that would support automated crisis management processes.

Crisis management is widely addressed by researchers in management science: in [34][35][36] leading ideas on crisis management in a business environment are presented; in [37][38] the context, concepts and practice of risk and crisis management in the public sector are discussed; in [39], a multidisciplinary approach to crisis management research, using psychological, social-political, and technological-structural research perspectives is defined. These works are mostly oriented on federal agencies, city administration, policy makers, practitioners and researchers in management and business administration. Up to our knowledge, only a few works are discussing the challenges of crisis management or its supporting information systems. An example is [40], where a lack of context-awareness (meteorological data and rainfall sensors) leads to a failure to adapt water release plans produced by a context-aware information system (CAIS), resulting in a severe flood event in Brisbane, Australia.

In our work, we consider crisis management as a particular case of knowledge-intensive process and focus on its "process" perspective.

Crisis management process is an example of a *knowledge-intensive process* (KiPs) [41]: it is unstructured and based on collaboration between actors; tacit knowledge of human actors, which is not embedded in the process model à priori, plays the central role in this process.

Davenport evaluates knowledge intensity by the diversity and uncertainty of process input and output [3]. A knowledge-intensive process is characterized by activities that may change on the fly, are driven by the scenario the process is embedded in and, most importantly, depend on the completeness of available contextual information. The set of users who should be involved in each step of the process may not be defined in advance and rather needs to be discovered as the process scenario unfolds.

Therefore, the requirements for PAIS supporting crisis management can be also applied to PAIS supporting KiPs.

A threat occurrence, the element of surprise, and a short decision time are common to a crisis [2]. These characteristics define the unpredictable nature of the management scenario. Instead of creating strategies for events that might occur in the future, crisis management involves reacting to an event once it has occurred.

During the crisis management, its supporting PAIS has to provide control over the process execution in order to guarantee the compliance with norms and regulations. Process flexibility is another fundamental need for a system supporting crisis management. According to [4], process flexibility can be summarized as *three abilities*:

(a) the ability to deal with uncertainty

(b) the ability to adapt process scenarios at run-time and

(c) the ability to evolve processes.

Explicit specifications of underlying processes play an important role in PAIS: they allow for better communication between stakeholders, enable process analysis and support redesign efforts [5]. In this work we show that the capacity of PAIS to support process flexibility is inherent to the underlying process modeling paradigm. In particular, we consider activity-oriented and state-oriented paradigms and discuss their advantages and limitations in supporting *the three abilities* mentioned above.

We illustrate our findings on the example of a flood management process – a crisis management process, implemented as a part of the COS Operation Center (COSOC). COSOC is a process-aware information system developed by COS&HT [6] in Russia. Since 2013, the COSOC solution has been used by the city administration of Novgorod. Underlying processes for COSOC (including flood management process) were designed with BPMN, following a widely accepted activity-driven paradigm.

To assess *the ability of the system to deal with uncertainty*, we need to study how the system collects and processes data in order to adequately assess the current situation. In our case study, we formulate it as our first research question:

1) How does the flood management process model support the run-time assessment of a crisis situation?

To assess *the ability to adapt process scenarios at run-time*, we need to study how the system supports the user, i.e., the process manager, in his/her "on-action" decision-making (especially in the situations not covered by the predefined operation scenarios). We formulate the second research question:

2) How does the flood management process model support the run-time assessment of a crisis situation?

To assess *the ability of the system to evolve processes*, we need to study how the system supports the user, i.e., process manager, in his/her "after-action" analysis and knowledge management activities. We formulate our third research question:

3) How does the flood management process model assimilate the new scenarios proposed by the process manager? What is the cycle of process evolution or redesign?

We examine the current specification of the flood management process designed using BPMN (our case study) focusing on the defined research questions. Then we propose an alternative model based on the state-oriented paradigm and discuss its advantages with regard to the activity-oriented paradigm.

The remainder of this paper is organized as follows: in Section 2 we discuss process modeling paradigms and study their support of flexibility in process models. In Section 3we present the COS Operation Center and the example of flood management process. In Section 4 we discuss the BPMN specification of the flood management process used in COSOC and examine how the activity-oriented paradigm supports the process flexibility. In section 5 we propose an alternative way to specify the flood management process based on the state-oriented paradigm and the statecharts formalism. In Section 6 we discuss the advantages and limitations of the two paradigms, present our conclusions and directions for the future work.

II. BACKGROUND

In this section, we present various process modeling paradigms and discuss their capacity to support flexibility in process models.

Within *the activity-oriented paradigm*, a process is specified as an ordered set of activities that the system has to carry out. Examples of activity- oriented formalisms include BPMN [7], YAWL [8], activity diagrams in UML [9] and other languages based on workflow concepts.

Activity-oriented process modeling implies that data emerges and evolves within a process according to a predefined control flow. Events are supposed to occur (or be processed) at specific moments of the execution predefined by the model. This paradigm suits predictable and highly repeatable processes. Crisis management processes are unpredictable [10]: events and process inputs can occur at any time during the executions; the order of activities can therefore not be predefined and depends on the current situation. Such behavior can thus not be captured by the workflow formalism.

In order to increase process flexibility and to better address unstructured and knowledge-intensive processes, activity-oriented formalisms have been extended with declarative parts, such as constraints [11], business rules [12] or configurable elements [13]. These formalisms can handle process variability within a potentially large number of configurations or scenarios. However, either such scenarios must be well identified upfront or the set of business rules (or configuration elements) must be regularly maintained by an expert. This can be seen as a limitation for crisis management processes.

According to *the product-oriented (or state-oriented) paradigm*, a process is seen as a product life cycle (i.e., a set of product states and transitions between these states). Examples of product-oriented modeling formalisms include state machines in UML [14], generic state-transition systems or state machines, such as FSM [15] or Petri Nets [16], and statecharts by D. Harel [17] created for the specification and analysis of complex discrete-event systems.

Traditional FSMs and their corresponding state-transition diagrams are efficient for tackling small problems. However, the complexity of a FSM model tends to grow much faster than the complexity of the problem it represents. This "state explosion problem" can be overcome by the introduction of multiple hierarchical levels for states and transitions. Indeed, this hierarchy gives a possibility to reuse some common behaviors across many states and, thus, to reduce the model complexity. This idea is explored in the formalism of statecharts [14][17].

The statecharts formalism specifies hierarchical state machines (HSM); it extends classical FSM by providing:

- *depth* – the possibility to model states at multiple hierarchical levels, with the notion of abstraction/refinement between levels;
- *orthogonality* – the possibility to model concurrent or independent submachines within one state machine;
- *broadcast communication* – the possibility to synchronize multiple concurrent submachines via events.

Within the state-oriented formalism, carried out activities depend on the current state of the product and the process scenario is adapted at run time, according to the evolution of the product. This paradigm suits well reactive systems specification [18] since the system's response to an event shall be defined not only by the type of this event but also by the current situation of the system i.e., its state.

Several research groups have reported on approaches to design and specify unstructured, knowledge-intensive processes based on the product-oriented paradigm:

- In [19], process instances are represented as moving through state space, and the process model is represented as a set of formal rules describing valid trajectories. This approach is grounded on the theory of automated control systems.
- In [20], a group of researchers from IBM propose an approach that incorporates process- and data-centered perspectives and is based on the concept of business artifacts.
- In [21], the Product-Based Workflow Design is presented. This approach explores the interaction between a product data model that reflects the product design and the process to manufacture this product represented by a workflow.
- The authors of [22] present case handling as a paradigm for supporting knowledge-intensive business processes. The authors compare case handling to workflow management and identify four problems. In particular, they recognize the lack of flexibility of workflow management systems and acknowledge the important role played by the "product" - the case - in the case handling. Their view on the case, however, remains activity-oriented: the proposed case definition explicitly includes the list of activities and their precedence relations assuming that they are known in advance.

The decision or goal-oriented paradigm extends the product-oriented view on the process: the successive transformations of the product are looked upon as consequences of decisions leading to some goal [23].

Goal-oriented modeling formalisms (examples include i*[24], KAOS [25], MAP [26]) support decision making by specifying goal hierarchies and tracing each decision within these hierarchies. Context-driven goal-oriented process models [27][28][29] support automated recommendations and user guidance, providing that for each goal all the situations (states) in which this goal is achievable are known. However, due to unpredictable sequences of events and non-repeatable execution scenarios in knowledge-intensive process, it will be hard if at all possible to model relations between various process situations, goals and activities that must/can be executed in order to achieve these goals.

In this work we discuss crisis management processes. In particular, we examine the example of flood management process. While being highly regulated, crisis management requires flexibility and reactivity and never follows the same scenario. While activity-oriented paradigm remains the main choice for process designers, we claim that the state-oriented paradigm has a great potential for specification of knowledge-intensive processes and crisis management processes in particular.

III. COS Operation Center solution for flood management: case study

In this section we present the COS operation center (COSOC) – a cross-domain information system developed by COS &HT in Russia. The system is used by the administration of Novgorod city and is planned for installation for the administration of Moscow Region and Krasnodar Region.

COSOC supports a large variety of processes within the city, including crisis management processes. We also introduce an example of a crisis management process: a flood management process, implemented as a part of COSOC.

A. COS Operation Center

The Operation Center is a process-aware information system used by a government to manage the variety of processes and cross-domain operations within the city, ranging from paper issuing for citizens, garbage collection, public transport management to monitoring and management of large scale emergencies.

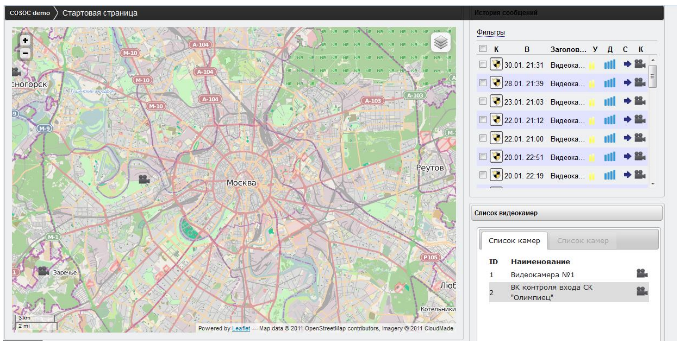


Fig. 1. A screenshot of the COSOC executive dashboard. The dashboard is divided into three areas: the city map, the list of events and the list of data sources for monitoring the situation on the object

The Operation Center provides the authorities with relevant information across the entire city through a common information space (*executive dashboard*). This dashboard contains data from various sources; it allows city service managers to have a full and comprehensive understanding of the issues and to coordinate the operation of multiple agencies in real time.

The functions of COSOC can be roughly divided into three groups: (i) data collection and visualization, (ii) analysis of the situation and decision making and (iii) triggering response processes.

COSOC collects the data related to different areas of the city life in a real time and visualizes this data on the executive dashboard (Fig.1). This dashboard lists the events in the summary table, with an option of sorting on key parameters (level of danger, urgency of response, etc.), and shows their geographical location on the map. Stationary and mobile video cameras, embedded sensors measuring traffic density, pollution concentration, temperature, radioactivity, calls and emails from citizens reporting on anomalies and accidents are examples of data sources used by COSOC.

The collected data is used for calculating *key indicators (KI)* that are used to assess a situation in the different city areas. The key indicators are visualized in a *Colored KI matrix* (Fig.2). This matrix provides an integrated and hierarchical view on the current situation in the city. Each cell of this matrix in Fig. 2 corresponds to an area of the city activity: electricity supply, water supply, healthcare, ecology and meteorology, transport and so on. The color of the cell indicates a situation: green – normal, yellow – alert, red – reaction is required and purple – emergency. The process manager can zoom into a cell for more detailed views, where the sub-areas and the values of their corresponding key indicators are displayed.

When an indicator exceeds some critical value (e.g., a traffic jam is detected or pollution exceeds a certain threshold), the system modifies the Colored KI matrix respectively and, if applicable, automatically triggers a response workflow from the predefined list. If a solution cannot be triggered automatically, COSOC generates a message for the process manager and proposes to choose a workflow from the list of available solutions.

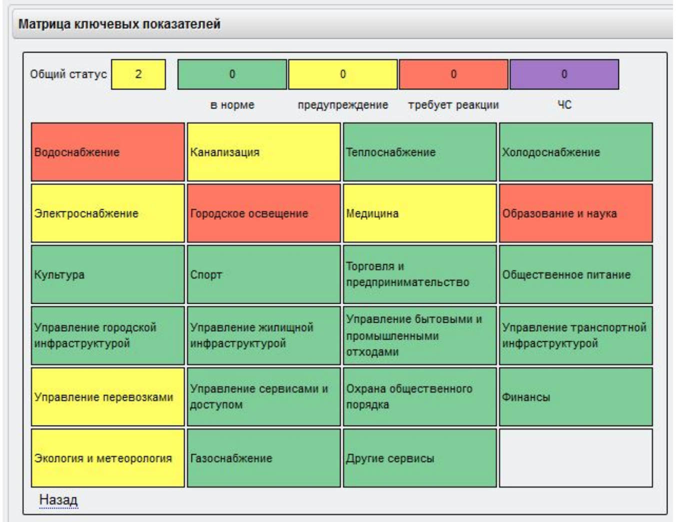


Fig. 2. A screenshot of a high-level view of the Colored KI matrix in COSOC. Each cell indicates an area of the city activity; the color code indicates the status (from normal – green, to emergency – purple). The process manager can zoom-in the cell in order to see the details

The COSOC process manager is a member of the city administration who is responsible for monitoring the situation and handling emerging issues. He/she can accept or decline the solution proposed by the system; when a workflow is triggered, he/she monitors its execution and intervenes when decision-making is required.

When the problematic situation is resolved, the process manager can provide feedback to the system: request for process improvement, modification of KI list, etc. All change requests are treated by the technical support team of COSOC and have to undergo a formal approval process before being implemented.

The following section presents an example of a flood management process on the Oka River in the Moscow region in Russia.

B. *Crisis Management in Case of Floods*

A flood is an overflow of water that submerges a land that is normally dry. Floods on the Oka River in the Moscow region are seasonal events caused by an increase in the flow of the river, provoked by intensive snow melting during the spring months. Cities built along the Oka River are confronted to the risk of flooding and can expect important damages, affecting thousands of people. Floods on Oka also represent substantial risks for the critical infrastructure facilities situated in the area: a railway bridge, a pontoon road bridge, an electric power plant, industrial storage facilities, etc.

Along with other types of crisis, the flood management process is highly regulated by federal authorities, including the ministry for Emergency Situations (MES), the Ministry of Internal Affairs, and the Ministry of Defense. For example, any crisis management process has to comply with the Emergency Management Guidelines [30] defined by MES. This document prescribes the activities that have to be carried before, during and after crisis situations by different public services and agencies of the city.

Contextual parameters such as water level, temperature, characteristics of flooded areas, status of the ongoing response operations etc., are collected by COSOC and displayed on the executive dashboard and colored KI matrix described in the previous section.

The flood alert is triggered when the daily temperature rises above a certain average defined for the season and holds for several days, provoking intensive snow melting in the area. The flood emergency is triggered when the water level in the Oka River rises above 10 cm.

Table I provides a brief description of the major phases of the flood on the Oka River.

MES and other regulating authorities define a distinctive list of operation procedures (i.e., responses) that have to be executed for each of the major phase defined in Table I. Nevertheless, there exist situations where the predefined operation procedures are not sufficient: disrupted telecommunication, electricity and water supply, lack of equipment or impossibility to deploy/relocate the required equipment – are examples of situations that compromise the predefined operation procedures and have to be resolved "on a case basis" by a human actor - the process manager. The process manager monitors the situation using the executive dashboard and the colored KI matrix and proposes the scenarios based on his/her experience and understanding of the situation.

TABLE I. FLOOD SCENARIO DRIVEN BY THE CHANGING WATER LEVEL IN THE OKA RIVER

Water level rise	Threats / Expected consequences	Response
>10 cm	Flood Alert	Inform citizens, deploy the equipment and set up temporary barriers
>10cm and keeps rising	Flood emergency	Declare emergency situation, evacuate people; prepare temporary accommodation
> 25cm	Minor damages in living areas; risk of disrupted water supply	Emergency water supply; patrol flooded zones, provide boats and reinforce water barriers
>40cm	Risk of severe damage in living areas	Rescue operations; secure bridges and organize deviations
>45cm	Disrupted road traffic	Close the pontoon bridge; secure strategic infrastructure facilities (industrial storages, electric power plant, etc.)
>60cm	Severe damages in living and industrial areas; Risk of presence of toxic substances in the river; Disrupted electricity supply	Rescuing operations; chemical control of water; evacuation of industrial storage facilities; temporal accommodation for citizens
> 75cm	Disrupted railway communication	Close the railway bridge

The flood crisis terminates when the water level gets back to normal, the response operations are terminated and the post-crisis reconstructions begin.

IV. MODELING THE FLOOD MANAGEMENT PROCESS : ACTIVITY-ORIENTED PARADIGM

The underlying processes in COSOC (including flood management process) are specified using the activity-oriented modeling paradigm. In this section, we examine the current model of the flood management process specified using BPMN. We analyze to what extent this specification supports flexibility by answering the questions stated in the introduction.

A. BPMN model of Flood Management Process in COSOC

The goal of the flood management process supported by COSOC is to *dispatch* the assignments for operation procedures according to the crisis development and in agreement with the rules defined by MES and other regulating authorities. The selected procedures are carried out by actors or groups of actors involved into crisis handling (MES, police taskforce, fire brigades, etc.).

Following the activity-driven modeling paradigm, the flood management process is explicitly modeled as a preordered set of tasks with predefined triggering conditions (BPMN events). Each process task represents an assignment that will be carried out by a designated actor or team.

Fig. 3 illustrates the flood management process as specified and implemented in COSOC. This diagram has been made with BizAgi Process Modeler [31]. For the purpose of this article, we simplified the original model while preserving the process logics, structure and main process elements.

There are four main actors defined for the flood management process: COSOC, MES, Police Taskforce and Environment. The Environment actor represents the infrastructure that provides in a real-time the information about the flood (e.g., social networks, wireless sensors, video cameras, other measurement equipment). Each process actor is modeled as a separate pool in Fig. 3. We show the process details only for the COSOC actor; other actors appear as "empty" boxes.

Following the flood scenario in Table I, the BPMN model identifies the major flood phases (based on the events received from the environment or from the MES and Police Taskforce) and specifies the operation procedures accordingly. The list of events is presented in Table II. The "water level" alerts (e.g., E0, E2, E3, E6, E8, E12 in the model) are generated automatically by the infrastructure, once the corresponding threshold is reached for the first time. Other events are extracted from the reports provided by the human actors. They typically indicate the beginning and the end of the operation procedures and report on specific issues during the execution of these procedures that require immediate reaction (e.g., E4, E7a, E9a, E10a, E11a in Table II).

The flood management process is triggered when the water lever h in the Oka River rises above 10 cm (*E0: Flood Alert*). In response to this alert, temporary flood barriers are set up (T1). If within the following 12h the water level h goes back to normal – the process terminates by sending *E1: End of Flood Alert* message to MES. Otherwise, the state of emergency is declared, Police Taskforce starts to patrol the area and the evacuation of citizens from the flooded zones is carried out (T2-T4).

The next phase of the flood management is triggered when the water level h rises above 25 cm. Here, COSOC generates assignments for Police Taskforce to provide boats and to start the emergency water provisioning (T5, T6). For simplicity, we omit the message flows between the actors in Fig. 3.

From this point, the Police Taskforce regularly reports on the situation sending messages to COSOC. The system uses these messages for a detailed situation analysis and for planning the next activities. The decision-making logic is modeled with a complex gateway G4 in the BPMN diagram.

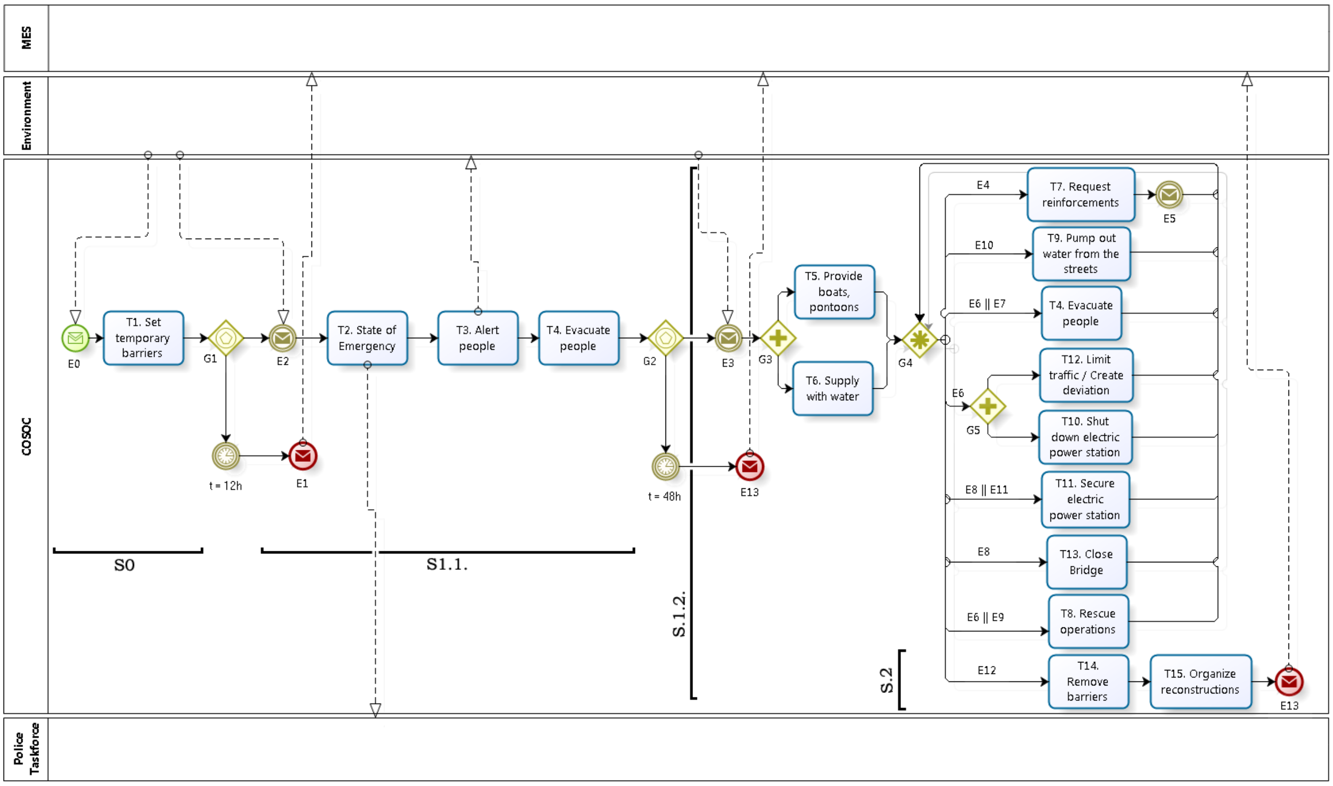


Fig. 3.The BPMN model of the flood management process implemented in COSOC

TABLE II. LIST OF EVENTS PROCESSED BY COSOC

ID	From	To	Description
E0	Environment	COSOC	Flood Alert: h > 10 cm
E1	COSOC	MES	End of Alert
E2	Environment	COSOC	Emergency: h >10 cm and keeps rising
E3	Environment	COSOC	Elevated Risk: h > 25 cm
E4	Police Taskforce	COSOC	Request for resources (e.g., manpower, equipment)
E5	MES	COSOC	Report: resources are sent
E6	Environment	COSOC	High Risk: h > 40 cm
E7	Police Taskforce	COSOC	Request for evacuation
E7a	MES	COSOC	Report: evacuation is terminated
E8	Environment	COSOC	Alert: h > 45 cm
E9	Police Taskforce	COSOC	Request for rescue operation
E9a	MES	COSOC	Report: rescue operation is terminated
E10	Police Taskforce	COSOC	Alert: streets are flooded
E10a	Police Taskforce	COSOC	Report: streets are cleaned
E11	Police Taskforce	COSOC	Alert: electric power plant is flooded
E11a	Police Taskforce	COSOC	Report: electric power plant is secured
E12	Environment	COSOC	Below Critical: h< 25cm
E13	COSOC	MES	End of Emergency

Here, various operation procedures can be (inclusively) selected based on the activation conditions (i.e., a specific event or combination of events occurred). Once the selected operation procedure terminates the control returns to G4 and a new iteration begins.

The same activity can be triggered several times if its activation condition is satisfied.

For example, *T17: Pump out water from the streets* can be triggered several times during the flood. Along those lines, some tasks will not be executed, as their activation condition is not met.

Some operation procedures produce the outcomes (events), which, in their turn, can trigger the other operation procedures. For example, during the execution of *T8: Rescue operation*, the lack of resources can be experienced (E4), triggering *T7: Request for Reinforcements.* Other examples of operation procedures and their associated outcomes are presented in Table III.

The process is terminated when the water level falls below a 25cm threshold (E12). The flood barriers get removed, reconstructions start (T18-T19). The message about the end of emergency state is sent (E13).

B. Analysis

We claim that the level of COSOC flexibility is inherent to its underlying process model and, thus, strongly related to the selected modeling paradigm. In the following, we analyze the BPMN specification of the process by answering the three questions raised in the introduction of this article:

1) How does the flood management process model support the run-time assessment of a crisis situation?

The crisis situation in COSOC is assessed based on the contextual parameters extracted from the events that the system continuously receives.

The number and type of contextual parameters are defined in the process model at design time. For COSOC flood management process, these parameters include the water level in Oka River, the air temperature, the surface and the characteristics of the flooded zones (e.g., presence of medical, childcare facilities, strategic objects etc.), the status of infrastructure in the flooded zones (water supply, electricity supply, telecommunications, roads), the amount of damage, incidents, the status of the ongoing response operations, etc.

Police and MES teams can transmit additional information about the situation. However, this information is not processed by the system. New events, contextual parameters or rules for triggering operation procedures can be introduced on the *model level* (i.e., by changing the BPMN specification). However, these modifications will not have an effect on the running *instance* of the COSOC flood management process. Therefore, no capacity to dynamically add a new contextual parameter is supported.

Our case study shows that the system (based on fixed contextual parameters) and the process manager (based on the additional information and personal experience) frequently come to different conclusions regarding the crisis situation assessment. For example, in certain conditions, emergency water provisioning (T6) might be required even before the *Elevated Risk Alert* (E3) is received. In such situations, the process manager has a choice: to "leave the system decide" (potentially leading to suboptimal or even erroneous scenarios) or to make decision himself/herself and to adapt the scenario proposed by the system.

In the following, we analyze the capacity of the activity-oriented process model to support such adaptations.

2) To what extent does the flood management process model allow for run-time scenario adaptation according to this assessment?

All possible flood management scenarios that the COSOC system supports are explicitly specified in the BPMN model at design time.

Once the flood management process is triggered – the model is instantiated. At run time, the appropriate response (activity) is defined automatically according to the evaluated conditions specified in the model. In certain cases, the system calculates the list of alternative responses and offers the process manager to make a choice.

The complex gateway G4 specifies the flood management scenario in a flexible way: the activities can be executed in various combinations and repeated multiple times before the process is terminated. However, only "predefined by design" activities can be triggered. When required, new activities can be added or the process logic can be changed on the *model level*. However, these modifications will not take effect for the running process *instance* in COSOC. Therefore, no adaptation of the process scenario at run time is supported.

Automated scenario definition reduces the risk of human errors related to decision-making in stressful conditions and information overload. In practice, however, the process manager handles many situations off-line, by communicating with the response team and by determining and adapting the crisis management scenarios according to their experience. Such adaptations and newly discovered scenario are extremely valuable for further evolution of the process and its implementing system.

In the following we analyze the activity-oriented model evolution capacity.

3) How does the flood management process model assimilate the new scenarios proposed by the process manager and what is the cycle of process evolution or redesign?

Whenever an unforeseen situation occurs -the process manager needs to decline the execution of a workflow recommended by the system and to define and execute a new workflow that is better adapted for a situation. For example, due to the road conditions, specific vehicles cannot reach the flooded areas; in this case, the rescue operations (T8) can be carried out by helicopters only, or immediate evacuation has to be triggered. Such situations and the corresponding (adapted) scenarios defined at run time by the process manager have a great value for further process evolution and improvement.

The business process lifecycle in COSOC follows the main stages defined in BPM and involves (re)design, configuration, enactment and evaluation [32].

Our experience shows that the process manager rarely has an expertise to undergo a new workflow definition: he/she often switches to the "off-line" mode and manages the process manually (e.g., by sending messages, making phone calls and so on). Therefore, the new scenarios are rarely getting integrated into the process model during the process enactment. They can be included into the *process model* only during evaluation and redesign. In this case, the new scenarios for crisis management process are designed with the assistance of domain experts (process manager and other actors are involved) than modeled by the process designer, tested and integrated into the system.

Due to the long re-design cycle, the expertise of the process manager cannot be regularly transformed into process improvements and can therefore be lost.

C. Summary

Within the activity-oriented paradigm, the contextual parameters and the decision-making logic are predefined in design-time. No run-time modifications are possible. The model explicitly defines activities associated with decisions. Even though the invocation order of activities is specified at run-time, according to received events, adding new activities or changing the event-activity association at run-time is not supported.

New scenarios can be integrated into the process model after process redesign. However, they will not take effect on the process instances that are currently running.

From the system perspective, the automated run-time assessment of a crisis situation and scenario definition ensures a full compliance with norms and regulations and reduces the risk of human errors. The unforeseen situations, however, have to be coordinated "off-line", with the resulted scenarios poorly traced in the system, preventing from process improvement. The lessons learned are typically discussed "after action" and can lead to the system update only after a long cycle of process model evaluation and redesign.

V. MODELING THE FLOOD MANAGEMENT PROCESS : STATE-ORIENTED PARADIGM

In this section, we apply the state-oriented paradigm for modeling flood management process. We examine the interest of this paradigm for modeling crisis management processes and show that within this paradigm, some shortcomings identified in the previous section can be solved.

Within the state-oriented paradigm, a process is described by a set of states and transitions between these states. Process execution starts at an **initial state** and terminates at a **final state**. A state transition is triggered when some condition is fulfilled. The sequence of states and transitions that leads from the initial state to the final state can be seen as a **process scenario**.

Some state-oriented approaches (e.g., Petri Net) associate a transition with the execution of one concrete *activity (or a group of activities)*. On the contrary, we associate a state transition with the occurrence of *a triggering event (or combinations of events)*. Compared to activity-oriented approaches that encourage the **early binding** of activities (at design-time), state-oriented paradigm supports **deferred binding**: at design-time, the process scenario can be seen as *a sequence of events*. The concrete activities that will produce these events can be selected or event "invented" in run-time.

Following this paradigm shift, a process specification is divided into two parts: *the state-transition part*, defined with a set of states and transitions between states and their triggering events, and *the activity part*, defined by a list of activities specified by their preconditions and outcomes. The process enactment can be seen as a dynamic selection of activities to produce some outcomes (events) that make the process progress towards its (desired) final state.

A. Statecharts model of Flood Management Process in COSOC

We design a state-oriented model of the flood management process preserving the semantics of the existing BPMN process model specified in Fig. 3. Fig. 4 illustrates the specification of the flood management process made in the YAKINDU Statechart Tool [33] using the formalism of statecharts [17].

Statecharts describe the process with a set of states (e.g., *S0: Flood Alert, S1: Flood Emergency, E2: Restoring Normal Functioning,* etc.) and transitions between them. Each state transition can be triggered by a specific event or combination of events (the event *E2: h > 10 cm and keeps rising* triggers a transition from S0 to S1). To maintain the consistency with the BPMN specification, in the statecharts specification we use the same list of events (Table II). Activities are not explicitly modeled in the statecharts diagram. The relationships between activities and events (i.e., possible activity outcomes) are illustrated in Table III.

In the statecharts notation, states are depicted by rectangular boxes with rounded corners. Statecharts use the notion of hierarchical state: this economical visual notation allows to specify real-size systems avoiding state explosion. The substate–superstate relation is depicted by boxes encapsulation. State transitions are represented with arrows and labeled with triggering events.

We define three main states for the flood management process: *S0: Flood Alert*, *S1: Flood Emergency* and *S2: Restoring Normal Functioning* (we indicate the rough correspondence between the states defined in the statecharts specification and the BPMN specification in Fig.3). S1 is refined in two substates: *S1.1.: Preparation* and *S1.2.: Emergency Control*. The former corresponds to the part of the BPMN specification where the preparations of the city facing the flood are carried out according to the MES regulations (i.e., the state of emergency is declared, citizens are informed, the evacuation of citizens from flooded zones is started etc.). S1.2. is triggered when the water level in Oka River rises above 25 cm (E3 in Table II).

Statecharts can model concurrency: *Living Area*, *Transport*, *Electric Power Plant* and *Resources* are four parallel sub-machines that describe the domains of flood management. When entering S1.2., the process simultaneously enters the (default) state in each corresponding sub-machine. Black circles with an outgoing arrow indicate default states.

Living Area sub-machine defines three states: *Elevated Risk, High Risk* and *Unsecured*. The transitions between these states describe how a flood will progress and will be managed: *Elevated Risk* is entered when the water level h rises above 25 cm (E3). The events received from Police Taskforce (e.g., requests for evacuation, rescue operations etc.) or from the environment (further rise of water level) trigger the *High Risk* state. The events E7a, E9a, E10a trigger the transition back to the "safer state" *Elevated Risk*. These events result from execution of some operation procedures (e.g., evacuation, rescue, pumping the water out of the streets or others). The state *Unsecured* is triggered when the event E4 indicating the lack of resources during execution of an operation procedure occurs. This event also triggers a transition in the *Resources* sub-machine from *Crisis Control* to *Insufficient Resources*. Once the resources are obtained (i.e., E5 is broadcasted) – transitions back to *Crisis Control* and *High Risk* are triggered in the respective sub-machines.

Along those lines *Electric Power Plant* sub-machine shows how the electric power plant (a strategic infrastructure object) is managed during the flood. According to the regulations, the power plant must be *Shut Down* when the water level rises above 40 cm (E6). If the water keeps rising – there is a risk that this facility will be flooded. Here the *Unsecured* state is triggered until the successful securing of the power plant (E11a) is reported.

Transport sub-machine is described with three states that are entered based on the water level: first, the *Normal Functioning* is maintained; when the water h rises above 40cm – only *Limited Traffic* is supported; when the water level h exceeds 45 cm threshold – the pontoon bridge has to be closed (*Bridge Closed*).

In our example, each state of the statechart can be associated with the list of obligatory and optional activities that must/can be carried out upon entering, upon exiting and while in this state.

With the state-oriented paradigm, the objective of the flood management process can be reformulated as follows: the process participants (i.e., MES and Police Taskforce) should respond to the events that occur in the environment (e.g., rise of water, weather changes etc.) by executing the operation procedures and producing the outcomes in order to maintain the secure functioning of the city in specified domains.

TABLE III. RELATIONS BETWEEN ACTIVITIES AND EVENTS IN COSOC FLOOD MANAGEMENT PROCESS

Activities	Events
T4. Evacuate people	E4, E9, E7a
T7. Request reinforcements	E5
T8. Rescue operations	E4, E9a
T9. Pump out water	E4, E9, E10a
T11. Secure electric power station	E4, E9, E7, E11a

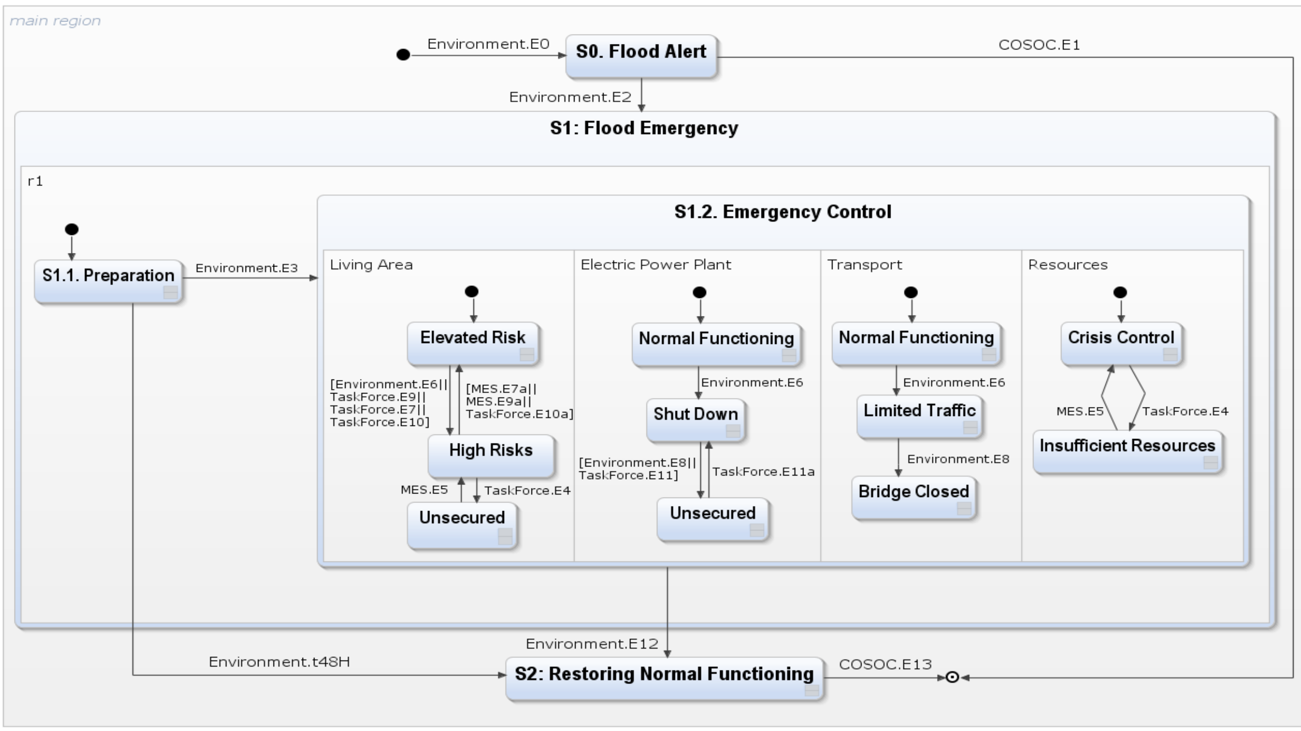
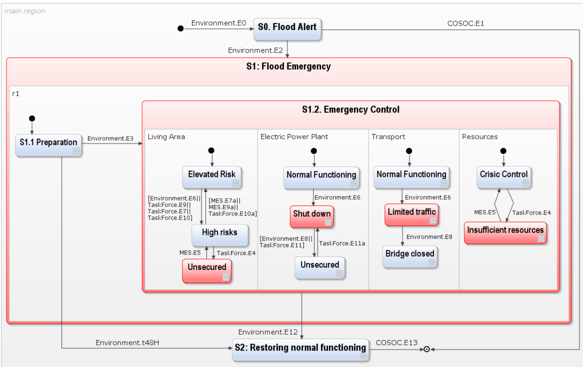


Fig.4. Statecharts diagram of COSOC flood management process

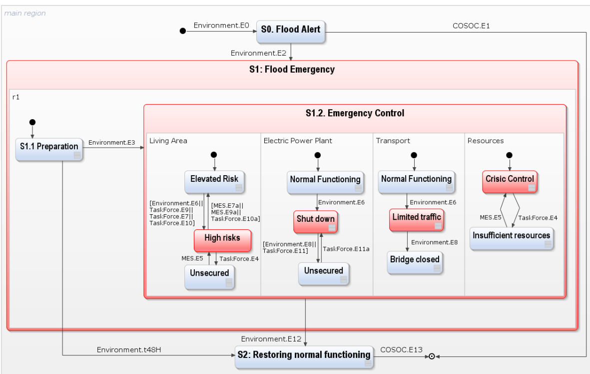
During the execution, decisions about specific operation procedures are taken according to a crisis situation that is described by a current configuration of the statecharts (i.e., the set of active states). To model the decision-making logic (according to the BPMN specification) we complement the statecharts diagram in Fig. 4 with a list of activities that can be executed according to the outcome they can produce (Table III). Contrary to the BPMN specification, activities are bound to the process scenario only at run time.

B. Analysis

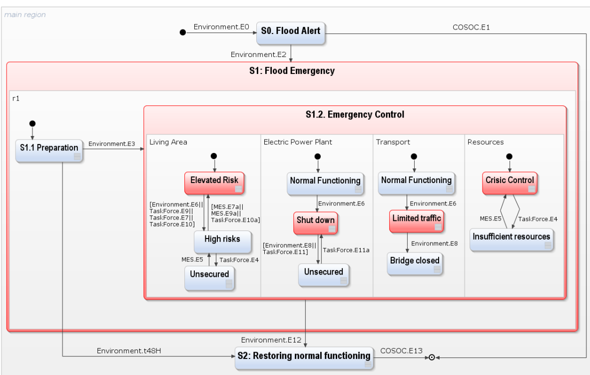
In this section, we explore the advantages and the limits of the statecharts specification. We structure our analysis by answering the questions raised in the introduction.



(a) Scenario: E0→E2→E3→E6→E4



(b) Scenario: E0→E2→E3→E6→E4→E5



(c)Scenario: E0→E2→E3→E6→E4→E5→E7a

Fig. 5. Simulation of flood management process with YAKINDU simulation tool. Current situation is described by the four active (red) states

1) How does the flood management process model support the run-time assessment of a crisis situation?

We have designed the process specification with the YAKINDU statecharts modeler. The YAKINDU simulation environment allows us to instantiate the statecharts specification and to simulate the process.

During the process simulation, the crisis situation in statecharts specification is represented by a current active configuration in the statecharts diagram. In Fig. 5 (a), the active configuration describes the crisis situation where, due to the water level h > 40 cm (E6 received) the Electric Power Plant is *Shut Down* and the *Limited Traffic* is maintained; due to the lack of resources (E4 received), the *Living Area* is *Unsecured*.

Once resources are received from MES (E5), the *Resources* sub-machine goes to the *Crisis Control* state whereas the *Living Area* returns to the *High Risk* state where some operation procedures need to be terminated (Fig. 5 (b)). When the corresponding procedures are terminated (E7a or E9a or E10a are received), the *Living Area* returns to the *Elevated Risk* state.

As for BPMN, the numbers of states and state transitions in statecharts are explicitly specified at design time. Addition or modification of states or state transitions can be done as a part of redesign and does not take effect on the running process instance (i.e., the current active configuration cannot be changed).

2) To what extent does the flood management process model allow for run-time scenario adaptation according to this assessment?

The state-oriented paradigm allows for *deferred activity planning:*an activity can be defined at run time, based on the desired outcome and on the context (i.e., resources, etc.).

TABLE IV. ALTERNATIVE ACTIVITIES INTEGRATED AT RUN TIME

Activities	Events
T4. Evacuate people (by land)	E4, E9, E7a
T4a. Evacuate people (by air)	**E4, E9, E7a**
T7. Request reinforcements	E5
T7a. Call for volunteers	**E5**
T8. Rescue operations	E4, E9a
T8a. Rescue operations (by air)	**E4, E9a**

In responce to unforceen conditions, the process manager can select from the available activities. Thanks to deferred binding, she can alsodefine*a new activity better adapted for a situation*. Table IV shows some examples of alternative activities not previewed by the original flood management procedures (and not specified in BPMN in Fig.3) but proposed by the process manager. For example, when the lack of resources message (E4) was received while pumping out the water, instead of requesting the reinforcements from MES, the city managed the situation by calling volunteers

(T7a). This activity produced the same desired outcome (E5) as T7 while being better adapted for the current situation.

New activities or sequences of activities can be integrated into the process scenario “on the fly”, without redesigning a process model.

3) How does the flood management process model assimilate the new scenarios proposed by the process manager and what is the cycle of process evolution or redesign?

The integration of new activities and events in the process specification can be done on fly, without redesigning the cycle. Definition of new states, transitions, triggering events, and refinement of states is a subject of redesign cycle.

C. Summary

Within the state-oriented paradigm, contextual parameters and decision-making logic is predefined at design time and cannot be modified at run time.

Activities are not associated with state transitions and thus do not have to be explicitly defined by the model. Thanks to the *deferred activity planning* mechanism, the process manager can select a concrete activity at run time, based on the desired outcome and on the context (i.e., resources etc.). Adding new activities or changing the outcome-activity association at run-time is possible.

From the system perspective, the state-oriented paradigm creates a recommendation system where the process manager plays the leading role in scenario definition. Unforeseen situations are handled within the system enabling seamless improvement of the process.

VI. CONCLUSION

Crisis handling requires high agility and reactivity and never follows the same scenario.

Focusing on efficiency, reliability and control of scenario executions, a PAIS supporting crisis management substantially reduces the risk of errors associated with information overload and human decision making under stressful conditions. However, PAIS capacity to support the user (operator or process manager) in recognizing and handling the situations that are not covered by predefined scenarios remains limited.

Our experience with COSOC shows that a concrete flood management scenario relies a lot on the experience and decisions of the process manager. Assessment of a situation, adaptive scenario planning and handling the unpredictable situations represent challenges for the supporting information system.

In this work, we were motivated by the lessons learned from working with COSOC:

a) Crisis management (and flood management in particular) cannot be fully automated by COSOC: while COSOC implements the regulations defined by MES, the process is still largely based on the tacit knowledge of the process manager. *Fully prescriptive process model leaves very little room for this tacit knowledge to be implemented.*

b) In COSOC, we have "full automation", when the system executes the predefined workflows, or "no automation" when the case is getting unpredictable and the process manager goes to “off-line” mode. *Providing recommendations and assisting in decision making are valuable capabilities to develop.*

c) The re-design cycle for COSOC is long and complex. “Best practices” from the process manager are not systematically documented. *Embedded knowledge management is a useful capability to develop in COSOC.*

In this paper, we show on the example of COSOC that the capacity of PAIS to support flexibility of the process is inherent to the underlying process modeling paradigm. We examined the BPMN (activity-oriented) specification of the flood management process designed for COSOC and proposed an alternative (state-oriented) specification of the same process made with statecharts.

While ensuring compliance with the norms and regulations, the activity-oriented paradigm provides very limited support for process flexibility at run time. Unforeseen situations cannot be handled within this paradigm since all activities and triggering events them need to be identified at design time.

The state-oriented paradigm allows us to exclude activities from the process design: we can state that "any activity is good as soon as it produces a desired outcome". In particular, it enables *deferred activity planning*, that gives more freedom to the process manager in choosing an activity that is adapted for a concrete situation.

This paradigm also allows for expanding the notion of a “management” system, providing the knowledge worker with guidance, decision support and knowledge management capabilities. These capabilities are valuable not only for PAIS supporting crisis management but also for PAIS supporting KiPs in general.

The process specification with statecharts has a practical interest for the current COSOC system. The Colored KI matrix used in COSOC for visualization of the current situation (Fig.2) can be obtained from the statecharts specification by associating a color to certain states or state configurations: once such a configuration is visited – the corresponding area of the Colored KI matrix is recolored.

The statecharts formalism has originally been developed for the design and simulation of complex discrete-event systems and thus its visual notation can be considered too minimalistic compared to BPMN. Extension of statecharts for the specifics of crisis management processes specification is a subject of our future work.

In this paper, we used YAKINDU modeling environment for specification and simulation of the flood management process. The possibility to animate the specification, to play different scenarios and to obtain the immediate visual feedback is very appealing and makes the design process extremely interactive and pleasant.

Although the extension of the notation and the adaptation of the simulation environment for the specifics of crisis management processes are desirable, we consider that the state-oriented modeling with statecharts has a great potential for crisis management process modeling.

References

[1] Wikipedia contributors, "Crisis management," Wikipedia, The Free Encyclopedia,accessed February 15, 2015,
http://en.wikipedia.org/w/index.php?title=Crisis_management&oldid=644287003

[2] M.W. Seeger, T.L. Sellnow, R.R. Ulmer, "Communication, organization and crisis," Communication Yearbook 21: 231–275, 1998.

[3] T.H. Davenport, "Improving knowledge work processes", in Sloan Management Review, vol. 37, 1996.

[4] M. Reichert, S. Rinderle-Ma, and P. Dadam, "Flexibility in process-aware information systems," Transactions on Petri Nets and Other Models of Concurrency II. Springer Berlin Heidelberg, 2009. 115-135, 2009.

[5] W.M. Van der Aalst, "Process-aware information systems: Lessons to be learned from process mining," in LNCS, vol. 5460, K. Jensen, W.M. Van der Aalst, Eds. Transactions on petri nets and other models of concurrency II, pp. 1–26, 2009.

[6] A. Helvas, M. Badanin, R. Pashkov, E. Kushnareva, O. Medovik, COS Operation Center, COS&HT, 2012, http://cosoc.ru/home

[7] Business Process Model Notation (BPMN) version 2.0., OMG Specification, Object Management Group, 2011.

[8] W.M. Van der Aalst, A.H. Ter Hofstede, "Yawl: yet another workflow language," Information Systems 30(4), pp. 245–275, 2005.

[9] J. Rumbaugh, I. Jacobson, G. Booch, "Unified Modeling Language Reference Manual," The (2Nd Edition), Pearson Higher Education, 2004.

[10] K. Swenson "Mastering the unpredictable: How adaptive case management will revolutionize the way that knowledge workers get things done," Meghan-Kiffer Press, Florida, USA, 2010.

[11] W.M. Van der Aalst, M. Pesic, and H. Schonenberg, "Declarative workflows: Balancing between flexibility and support," Computer Science-Research and Development, 23(2): 99-113, 2009.

[12] M. Bajec, M. Krisper, "A methodology and tool support for managing business rules in organisations," in Information Systems 30(6), pp. 423-443, 2005.

[13] M. Rosemann, W.M. Van der Aalst,"A configurable reference modelling language," Information Systems 32(1), pp. 1-23, 2007.

[14] D. Harel, E. Gery, "Executable object modeling with statecharts," in proceedings of the 18th International Conference on Software Engineering, pp. 246-257, ICSE'96, IEEE Computer Society, Washington, DC, USA, 1996.

[15] G. Plotkin, "A structural approach to operational semantics," 1981.

[16] T. Murata, "Petri nets: Properties, analysis and applications,"in proceedings of the IEEE 77(4), pp. 541-580, 1989.

[17] D. Harel, "Statecharts: A visual formalism for complex systems," in Science of computer programming 8(3), pp. 231-274, 1987.

[18] M. Reichert, S. Rinderle-Ma, and P. Dadam, "Flexibility in process-aware information systems," Transactions on Petri Nets and Other Models of Concurrency II. Springer Berlin Heidelberg, 2009. 115-135, 2009.

[19] W.S. Humphrey, "Managing the software process," (Hardcover), Addison-Wesley Professional, 1989.

[20] R. Hull, E. Damaggio, R. De Masellis, F. Fournier, M. Gupta, F.T. Heath, III, S. Hobson, M. Linehan, S. Maradugu, A. Nigam, P.N. Sukaviriya, R. Vaculin,"Business artifacts with guard-stage-milestone lifecycles: Managing artifact interactions with conditions and events," in proceedings of the 5th ACM International Conference onDistributed Event-based System, DEBS '11, ACM, pp. 51–62,New York, NY, USA, 2011.

[21] H.A. Reijers, S. Limam, W.M. Van der Aalst,"Product-based workflow design,"in journal of Management Information Systems 20(1), pp. 229-262, 2003.

[22] G. Booch, I. Jacobson, and J. Rumbaugh, "OMG unified modeling language specification," Object Management Group ed: Object Management Group, 1034, 2000.

[23] S. Nurcan, M.H. Edme, "Intention-driven modeling for exible workow applications," Software Process: Improvement and Practice 10(4), pp. 363-377, 2005.

[24] E.S. Yu, "Towards modelling and reasoning support for early-phase requirements engineering," in Requirements Engineering, 1997, proceedings of the Third IEEE International Symposium on. pp. 226-235, 1997.

[25] A. van Lamsweerde, "Goal-oriented requirements engineering: a guided tour,"in Requirements Engineering, 2001,proceedings Fifth IEEE International Symposium on, pp. 249-262, 2001.

[26] C. Rolland, N. Prakash, A. Benjamen, "A multi-model view of process modelling," in Requirements Engineering 4(4), pp. 169-187, 1999.

[27] C. Rolland, C. Souveyet, M. Moreno, "An approach for defining ways-of-working,"in Information Systems 20(4), pp. 337-359, 1995.

[28] K. Pohl, K. Weidenhaupt, "A contextual approach for process-integrated tools," inM. Jazayeri, H. Schauer, Eds. Software Engineering ESEC/FSE'97, LNCS, vol. 1301, pp. 176-192,Springer Berlin Heidelberg, 1997.

[29] P. Soffer, T. Yehezkel, "A state-based context-aware declarative process model,"in Enterprise, Business-Process and Information Systems Modeling, pp. 148-162,Springer, 2011.

[30] Emergency Management Guidelines, The ministry of the Russian Federation for civil defense, emergencies and elimination of consequences of natural disasters, 2013, http://www.mchs.gov.ru/upload/site1/document_file/8Ksr3cH1vE.pdf

[31] BizAgi Modeler version 2.9, BizAgi, UK, 2015, http://www.bizagi.com/

[32] M. Weske, "Business process management: concepts, languages, architectures," 2nd ed. 2012, XV, 403 p. 300 Springer-Verlag Berlin Heidelberg, 2012.

[33] Yakindu Statechart Tools version 2.4. for Eclipse Luna, 2014, http://statecharts.org/

[34] Harvard Business Press, "Harvard Business Review on Crisis Management," 1 edition, 2000.

[35] Harvard Business Review, "Crisis Management: Mastering the skills to prevent disasters," Harvard Business Essentials, 2004.

[36] S. Fink, "Crisis management: Planning for the inevitable," American Management Association, Reed Business Information, Inc., 1986.

[37] K.B. Penuel, M. Statler, R. Hagen, "Encyclopedia of Crisis Management," SAGE Publications, Inc., 2013.

[38] L. Drennan, A. McConnell, "Risk and crisis management in the public sector (Routledge masters in public management)," Routledge; New Ed edition, 2007.

[39] C.M. Pearson, J. A. Clair,"Reframing crisis management,"Academy of management review," 23.1,pp. 59-76, 1998.

[40] K. Ploesser, "A design theory for context-aware information systems," 2012
http://eprints.qut.edu.au/60865/1/Karsten_Ploesser_Thesis.pdf

[41] C. Di Ciccio, A. Marrella, and A. Russo,"Knowledge-intensive processes: Characteristics, requirements and analysis of contemporary approaches," *Journal on Data Semantics*, pp. 1-29, 2014.