Event-Driven Agility of Crisis Management Collaborative Processes

Anne-Marie Barthe-Delanoë
Université de Toulouse – Mines Albi
anne-marie.barthe@mines-albi.fr

Frédérick Bénaben
Université de Toulouse – Mines Albi
frederick.benaben@mines-albi.fr

Sabine Carbonnel
Université de Toulouse – Mines Albi
sabine.carbonnel@mines-albi.fr

Hervé Pingaud
Université de Toulouse – Mines Albi
herve.pingaud@mines-albi.fr

ABSTRACT

This article aims at presenting a whole approach of Information Systems interoperability management in a crisis management cell. We propose a Mediation Information System (MIS) to help the crisis cell partners to design, run and manage the workflows of the response to a crisis situation. The architecture of the MIS meets the need of low coupling between the partners' Information System components and the need of agility for a such platform. Based on the Service Oriented Architecture (SOA) and the Event Driven Architecture (EDA) principles which, combined to the Complex Event Processing (CEP) principles, it will leads to an easier orchestration, choreography and real-time monitoring of the workflows' activities, and even allows the automated agility of the crisis response on-the-fly —we consider agility as the ability of the processes to remain consistent with the response to the crisis—.

Keywords

Event-driven architecture, Service-oriented architecture, Model-driven architecture, Complex-event processing, Event cloud, Detection, Adaptation, Agility.

INTRODUCTION

As stated by (Devlin, 2006), a crisis situation can be political, military, economical, humanitarian, social, technological, environmental or sanitary. Regardless its nature, a crisis is an abnormal situation, which is the result of an instability impacting a subpart of the world (called ecosystem or system) with unacceptable consequences (Devlin, 2006; Lagadec, 1992). Such a situation implies to deal with the crisis management through a dedicated set of stakeholders in charge of the crisis response. According to (Atlay and Green, 2006; Beamon and Kotleba, 2004) the crisis management lifecycle is composed of four main steps starting with (i) mitigation, followed by (ii) preparedness, succeeded by (iii) a response phase after a crisis breaks out and finally (iv) recovery which aims at restore the ecosystem after the crisis.

The efficiency of the response step is determined by the speed and the accuracy with which information can be managed and exchanged among the partners (i.e. organizations, people and devices involved into the collaboration). Considering the fact that an Information System (IS) is the visible part of an organization, our point is to tackle organizations' collaboration issue through ISs interoperability satisfying the business requirements. Interoperability is defined by the European Network of excellence InterOp as "the ability of a system or a product to work with other systems or products without special effort from the customer or user" (Konstantas, Bourrières, Léonard and Boudjilida, 2005). It is also defined in (Pingaud, 2009a) as "the ability of systems, natively independent, to interact in order to build harmonious and intentional collaborative behaviours without deeply modifying their individual structure or behaviour".

Another point to deal with during the response phase is the evolutionary character of a crisis situation. Due to this fact, the system shall remain compliant with the expectations of the actors. This implies to measure the efficiency of the response, to be able to take in account the changes of the crisis itself or in the crisis cell.

Considering these points, we aim at design and produce a model-driven Mediation Information System (MIS) to support the interoperability among the partners and to keep the response workflows relevant to the crisis situation, through the Mediation Information System Engineering 2.0 (MISE 2.0) project. The first version of the MIS prototype (result of the MISE 1.0 project) was successfully used in the French funded project ISyCri (ISyCri stands for Interoperability of Systems in Crisis situation), whose one objective was to design an IS for several partners who have to solve, or at least to reduce, a crisis into which they are involved. The MISE 2.0 prototype, presented in this paper, is under construction and aims at solve some limits and assumptions of the MISE 1.0 prototype. It will be tested on a realistic use-case (a nuclear plant accident).

MISE DESIGN TIME

Like MISE 1.0 (Barthe, Bénaben, Truptil, Lorré and Pingaud, 2011), MISE 2.0 design approach is based on a Model-Driven Approach (MDA) through an automated model transformation including:

- The collaborative situation characterization by knowledge gathering (situation layer), as fully described in (Mu, Bénaben, Pingaud, Boissel-Dallier and Lorré, 2011). A model of the current crisis situation is realized, using the crisis meta-model (see Figure 1).

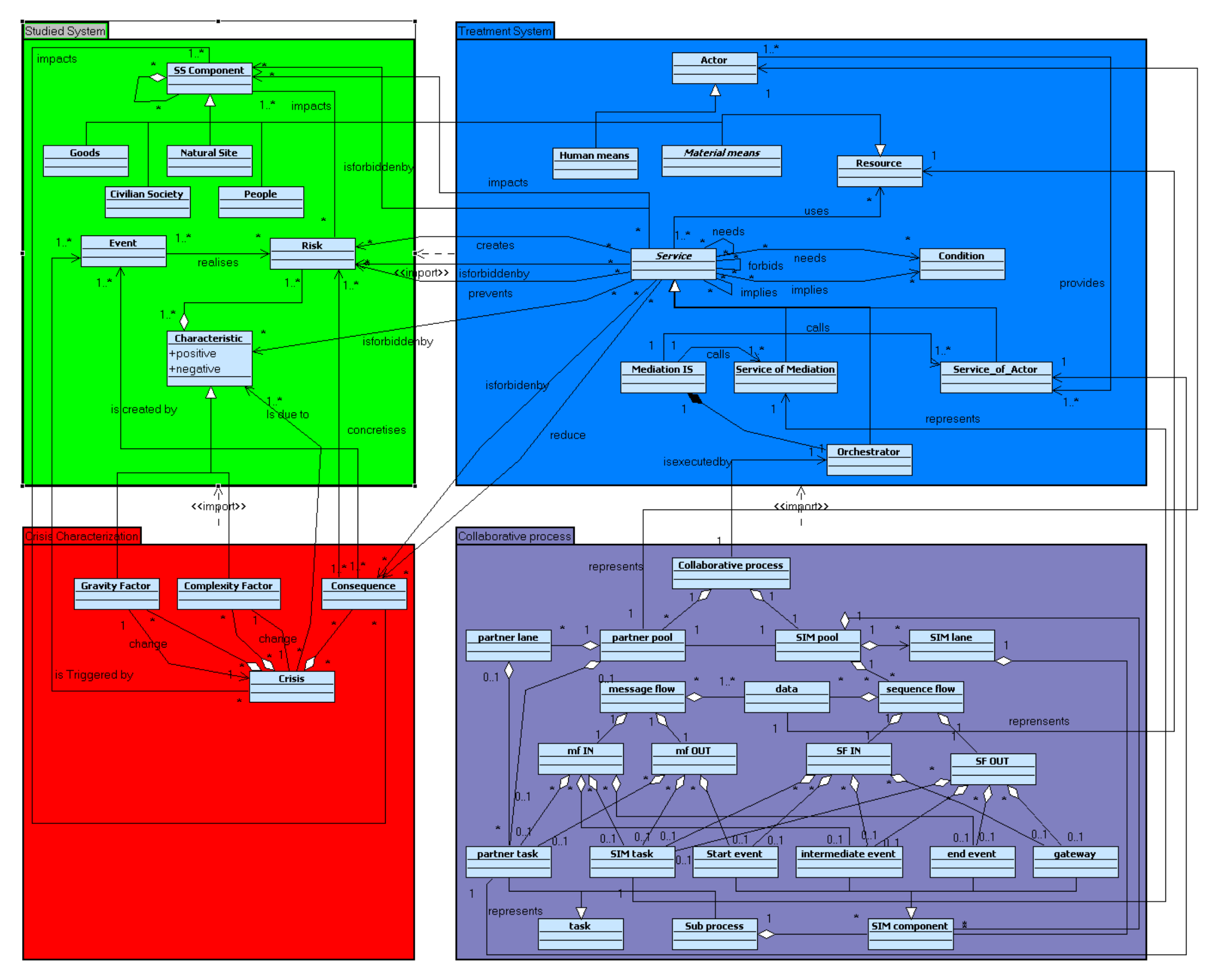

Figure 1. Crisis meta-model (Bénaben, Hanachi, Lauras, Couget and Chapurlat, 2008)

- The design of the collaborative process cartography (solution layer), across the three abstraction layers (business, logic and technological). At this step, a n-to-m semantic reconciliation , explained in (Bénaben, Boissel-Dallier, Pingaud and Lorré, 2012) is made between the deduced business activities —abstract level— and the technical services offered by the partners —concrete level—. There is also a data transformation activity (the data's format from the response of the service A may not be compliant with the expected format of incoming data for service B).
- And finally, the execution step: the deduced collaborative technical processes are automatically implemented as executable workflows (implementation layer).

In the remainder of this article, we will base our presentation on the Figure 2, which summarizes the overall MIS's architecture of our MISE 2.0 prototype, containing three main parts (the crisis situation, the crisis cell – the MIS— and the cloud event processing).

MISE RUNTIME

The main hypothesis of this article concerns the partners' IS that are supposed to follow the same conceptual logical architectural philosophy: the Service Oriented Architecture (SOA) (Vernadat, 2007). If it is not the case, we consider partners' IS may be SOA compliant with the help of interfaces between non-SOA applications and the MIS (whose deployment is based on SOA principles). The MIS will run on a server, more precisely on an Enterprise Service Bus (ESB), which is Petals ESB[1] and all the tools used to design and then run the workflows are services hosted on this ESB (it is used for both design time and runtime).

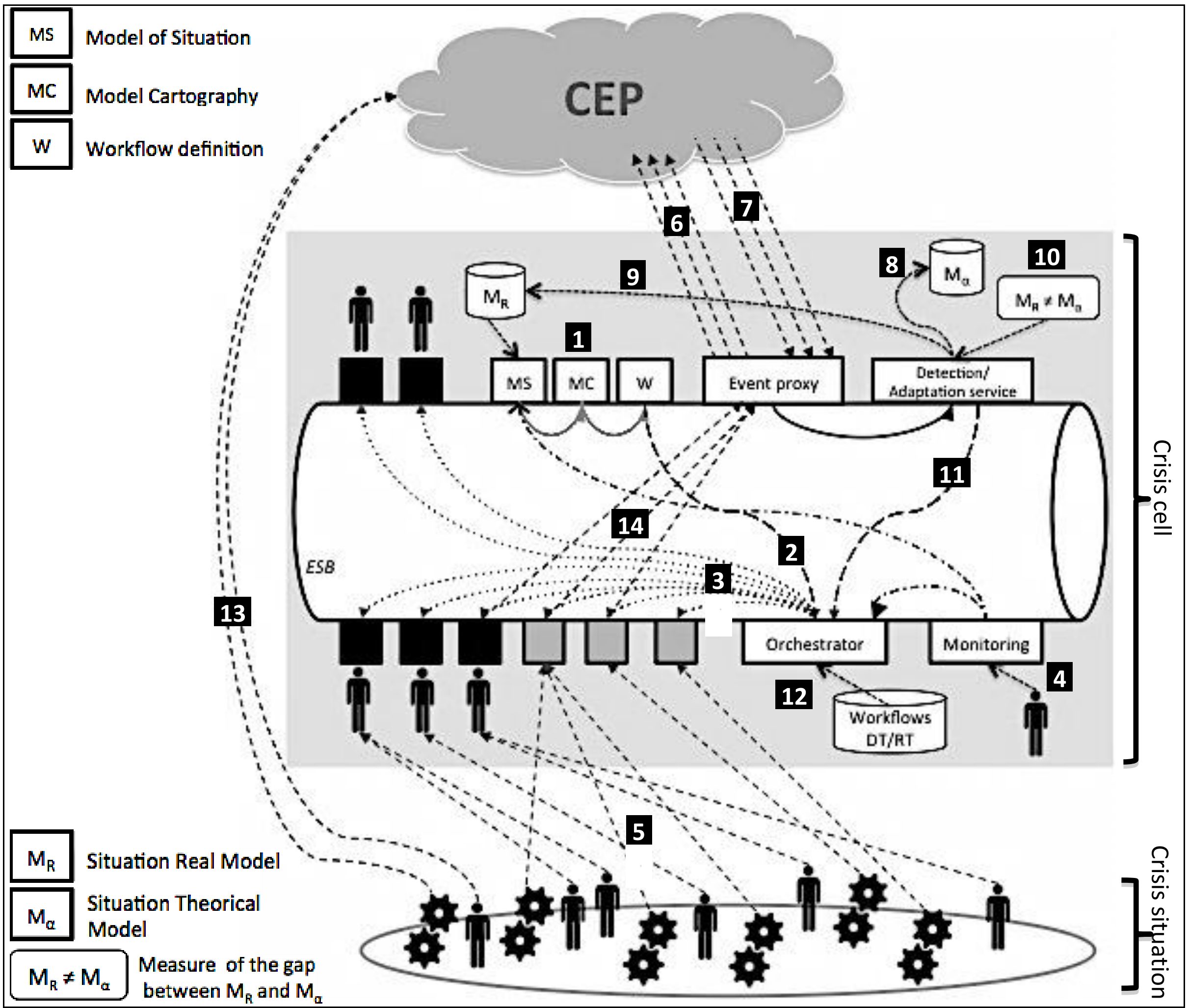

Figure 2. Global view of the event-driven collaborative response to the crisis.

On Figure 2, we can see the MISE Design Time services (1): these are Model Situation (MS) service, Model Cartography (MC) service, and Workflow definition (W) as described in the previous "MISE Design Time" section. These services are called during the Design Time step but they can also be called to redesign the response workflows (see MISE Agility section). Once the collaborative workflows and the MIS are designed, they are deployed on the ESB and executed by the orchestrator component, a workflow engine (2). The execution of the workflows calls numerous services (3), such as technical services (i.e. softwares) or basic interfaces between human beings and the MIS. The monitoring service allows users to check the execution of the response workflows (4).

We can note that the Design Time workflows (used to deduce the response workflows and the MIS implementation) and the Runtime workflows (i.e. the response workflows) are recorded in a database (12) that allows feedback about the crisis characterization and the deduced and run workflows.

[1] Petals ESB is developped by the French open source software editor Petals Link.

MISE AGILITY

By nature and by the effects of the collaborative processes to solve or reduce the crisis, a crisis situation is an unstable and evolutionary phenomenon. So, we can consider that as the crisis situation evolves, the crisis response may be not relevant after a while. According to (Pingaud, 2009b), two kinds of evolutions of such collaborative situations exist:

- The evolution of the crisis situation itself: the perceived characteristics of the crisis, in particular the issues to solve, are not the same at the beginning of the crisis and need a new response to the crisis.
- The evolution of the response to the crisis: the management of the response to the crisis situation may evolve due to (i) an evolution of the structure of the crisis cell (e.g. arrival or leaving of stakeholders, lack of resources), (ii) a dysfunction of the execution of a service (leading to the interruption of the workflow of the response), or (iii) due to a partial initial definition of the process of the response.

Thus any changes, any evolution, any information that could challenged collaborative processes' accuracy and relevancy have to be managed. According to (Chandy and Schulte, 2009; EPTS, 2011; Etzion and Niblett, 2011; Luckham and Schulte, 2008) these elements that happened and that embedded data can be considered (and managed) as events. These events are produced by (i) the people on the crisis field (e.g.: policemen, firemen, EMS team) and by devices (e.g.: radiation sensor, weather measurement station) (point 13 on Figure 2), (ii) the services used by the response workflows (14). These events are collected in an event cloud.

The use of Event-Driven Architecture (EDA) principles (Luckham et al., 2008; Michelson, 2006; Maréchaux, 2006; Josuttis, 2007) to complete the SOA principles —on which the MIS is based— allow us to take the previously described events into account. We use a Complex Event Processing (CEP) engine (point (6) on Figure 2) to consume and manage these events. The CEP engine, which is Esper[2], not only allows the choreography of the collaborative processes by event management (to manage exchange between the processes and to monitor the workflows' execution) (Peltz, 2003; Barros, Dumas and Oaks, 2005) but also filters and applies business rules to detect relevant events or combination of events. For example, two events (a wind direction measure and the caesium 137 rate in ambiant air), which are not seen as risks when seen separately, may have a different meaning if they are considered together. In this example, the CEP can send an event to inform the MIS of a change in the crisis characterization (7).

The CEP generates events, which are received on the ESB through the event proxy. Depending on their content, they will be consumed by the partner's services or by the Detection/Adaptation (D/A) service. In order to be agile, as we consider agility as the combination of event detection and adaptation, the MIS have to detect the events which have an impact on the crisis response and to define an adaptation regarding these events. The D/A's role is (i) to upate both theoritical situation model (the planned and expected situation model at time t) (8) and real situation model (the « what happened actually » situation at time t) (9) through the received events. Then the D/A has (ii) to measure and identify the differences between these two situation models (10) and (iii) to propose a relevant adaptation of the MIS through a complete or partial execution of the Design Time workflows (11). The adapted workflows are not only executed, they are also stored in order to capitalize knowledge (12). In a few words, we can say that the use of CEP allows refining the vision of the crisis situation (i.e. the impacted ecosystem and the crisis cell) while the D/A allows refining the crisis response.

CONCLUSION

People, devices, softwares, internal and external services of the MIS: if all these information channels can be turn into services, or plugged to human-to-machine service interfaces, they can be connected to the MIS's ESB. Then this ESB executes the collaborative processes between services and service interfaces. The CEP subscribes to the events produced by the collaborative processes (as event streams or event cloud), through the event proxy (hosted on the ESB).

We can work on these event streams/event cloud to follow the execution of the services called by the workflows (orchestration), to follow the event exchanges between the processes (choreography) and to allow the MIS monitoring. The low coupling provided by the combination of SOA and EDA in the MIS architecture highly reduces the integration costs and facilitates the connection between heterogeneous services provided by heterogeneous partners. Briefly, the aim of this article was to show that the MIS allows the coupling between systems that cannot ignore each other and that cannot be connected due to the unability to cover all possible connections (explosive combinatory) and their relevance at a given time. Another perspective concerns the crisis

[2] Esper is developped by the American software editor EsperTech.

situation modeling. It is currently human driven through an editor, but it could also be driven by the detection of relevant events in the system, allowing an automated modeling of the crisis situation.

ACKNOWLEDGMENTS

This work has been partially funded by the French Research Agency (ANR) regarding the research project SocEDA (SOCial Event Driven Architecture) (Grant ANR-10-SEGI-013). This project aims to provide dynamic and adaptive workflows to collaborative situations (like crisis situations) through EDA and CEP. The authors would like to thank the project partners for their advices and comments regarding this work.

REFERENCES

1. Atlay, N., Green, W. III (2006) OR/MS research in disaster operations management, *European Journal of Operational Research*, 175, 475–493.
2. Beamon, B.M., Kotleba, S.A. (2004) Inventory modelling for complex emergencies in humanitarian relief operations, International Journal of Logistics: Research and Applications 9, 1–18.
3. Barros, A., Dumas, M., Oaks, P. (2005) A critical overview of the Web Services Choreography Description Language (WS-CDL), *BPTrends.*
4. Barthe, A.-M., Bénaben, F., Truptil, S., Lorré, J.-P., Pingaud, H. (2011) MDI for SOA management of crisis, *Workshop "Advanced results in MDI/SOA innovation" in IWEI 2011 Third International IFIP Working*, Stockholm, Sweden.
5. Bénaben, F., Hanachi, C., Lauras, M., Couget, P., Chapurlat, V. (2008) A Metamodel and its Ontology to Guide Crisis Characterization and its Collaborative Management, *Proceedings of the 5th International ISCRAM Conference,* Washington DC*,* USA.
6. Bénaben, F., Boissel-Dallier, N., Pingaud, H., Lorré, J.-P. (2012), Semantic issues in model-driven management of information system interoperability, *International Journal of Computer Integrated Manufacturing.*
7. Chandy, M. K., Roy Schulte, W. (2009) Event Processing, Designing IT Systems for Agile Companies, McGraw Hill.
8. Devlin, E. (2006) Crisis management planning and execution, Taylor and Francis Group, 528.
9. Etzion, O., Niblett, P. (2011) Event Processing in Action, Manning Publications.
10. Event Processing Technical Society (2011), website: http://www.ep-ts.com
11. Josuttis, N. M. (2007) SOA in Practice: the Art of Distributed System Design, O’Reilly, 134-136.
12. Konstantas, D., Bourrières, J.-P., Léonard, M., Boudjilida, N. (2005) Preface of: Interoperability of Enterprise Software and Applications, *I-ESA’05 Interoperability of Enterprise Software and Applications*, Springer, Geneva, v-vi.
13. Lagadec, P. (1992) La gestion des crises, *Ediscience International*, 326.
14. Luckham, D., Schulte, R. (2008) Event Processing Glossary – Version 1.1, Event Processing Technical Society.
15. Maréchaux, J.-L. (2006) Combining Service-Oriented Architecture and Event-Driven Architecture using an Enterprise Service Bus, IBM Software Group.
16. Michelson, B. M. (2006) Event-Driven Architecture Overview: Event-Driven SOA Is Just Part of the EDA Story, Patricia Seybold Group.
17. Mu, W., Bénaben, F., Pingaud, H., Boiseel-Dallier, N., Lorré, J.-P. (2011) A Model-Driven BPM Approach for SOA Mediation Information System Design in a Collaborative Context, *Proceedings of the IEEE 8th International Conference on Services Computing (SCC 2011)*, Washington DC, USA.
18. Peltz, C. (2003) Web Services Orchestration and Choreography, *Computer*, 36, 10, O’Reilly, 46-52.
19. Pingaud, H. (2009) Prospectives de recherche en interoperabilité: vers un art de la mediation, Plenary Lecture, 8th International Congress on Industrial Engineering (CIGI), Tarbes.
20. Pingaud, H. (2009) Rationalité du développement de l’interopérabilité dans les organisations, Management des technologies organisationnelles, 19–30. Presses de l’Ecole des Mines de Paris, France.
21. Vernadat, F. (2007) Interoperable enterprise systems: Principles, concepts, and methods, *Annual Reviews in Control*, 31, 1, 137-145.