An Event-Driven Platform for Agility Management of Crisis Response

Anne-Marie Barthe-Delanoë, Sabine Carbonnel, Frederick Benaben, Hervé Pingaud

An Event-Driven Platform for Agility Management of Crisis Response

Anne-Marie Barthe-Delanoë, Ecole des Mines Albi-Carmaux, University of Toulouse, Toulouse, France

Sabine Carbonnel, Ecole des Mines Albi-Carmaux, University of Toulouse, Toulouse, France

Frédérick Bénaben, Ecole des Mines Albi-Carmaux, University of Toulouse, Toulouse, France

Hervé Pingaud, Jean-François Champollion University, Albi, France

ABSTRACT

This article aims at presenting a whole approach of Information System Interoperability management in a crisis management cell: a Mediation Information System (MIS) may be used 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 needs of low coupling between the partners' Information System components and the need of agility for such a platform. It is based on Service Oriented Architecture (SOA) and Event-Driven Architecture (EDA) principles that are combined to the Complex Event Processing (CEP) principles. This should leads on the one hand to an easier orchestration, choreography and real-time monitoring of the workflows' activities, on the other hand to assume on-the-fly automated agility of the crisis response (considering agility as the ability of the processes to remain consistent with the response to the crisis).

Keywords: Adaptation, Agility, Complex-Event Processing, Detection, Event-Driven Architecture, Event Cloud, Model-Driven Architecture, Service Oriented Architecture

1. 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 unaccept-able 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 & Green, 2006; Beamon & Kotleba, 2004) the crisis management lifecycle is classically composed of four main steps starting with (i) mitigation, followed by (ii) preparedness, succeeded by (iii) a response phase and finally (iv) recovery.

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. 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 & Boudjilida, 2005) (pp. v-vi). It is also defined by 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".

As a crisis situation is an unstable phenomenon (by nature or by effect of the crisis response), it may challenge the relevancy of the collaborative processes run among the ISs of the crisis cell partners.

Mainly, the problem statement is the following: how to ensure the agility of the crisis response and the use of various and heterogeneous data sources to feed the ISs of the collaboration? How to analyse all the gathered data and extract the relevant information for a given crisis situation at time t?

2. MISE APPROACH OVERVIEW

Considering these points, we aim at designing and producing a Mediation Information System (MIS) to support interoperability among partners and to keep the response workflows relevant to the crisis situation, through the Mediation Information System Engineering 2.0 (MISE 2.0) project. This part of this research work is presented in "MISE Design Time" and "MISE Run Time" sections. 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 In-

teroperability 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.

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 possibly changing requirements of the situation. This implies to measure the efficiency of the response, to be able to take into account the changes of the crisis itself or in the crisis cell (*i.e.* the events happening in the studied ecosystem). In a few words, a continuous response to the crisis is needed.

Several research projects focused on the event-management as a mean to be more accurate about the situation at time t on the field of the crisis, but also to be more efficient to manage the events sent by the crisis' ecosystem. Some of these projects are the French funded project SocEDA, the European funded projects PLAY (Truptil, Barthe, Bénaben, & Stuehmer, 2012) and PRONTO (Marterer, Matthias, & Koch, 2012). Such platforms integrate an Event-Driven Architecture layer, that provide them the ability to retrieve, manage and even generate complex events in order to help the users to filter the events and detect any anomaly. But such event-management is quite limited by the fact that only known types of events can be managed by static business rules. We believe that the EDA layer is a real benefit that completes the architecture of existing SOA platform (like MISE 1.0 and MISE 2.0) but its implementation is not limited to the use of a single Complex Event Processing engine (which plays the role of event filter, event manager and event generator). While (Yu & Cai, 2012) focus on the modeling of the collaborative activities and their dependencies and on the event impact on the states of the activities to solve this limitation, MISE 2.0 focuses on the detection of divergences between the whole crisis model and the effects of the events on both the crisis situation and the crisis cell (the partners and their activities). Then we propose a tool to adapt the crisis response if needed (via a partial or complete redesign/a partial or complete re-execution of

the response). This part of this research work is presented in the remainder of this article, in the "MISE Agility" section.

The MISE 2.0 prototype, presented in this paper, is under construction and aims at solving some limitations and assumptions of the MISE 1.0 prototype. It will be tested on a realistic use-case (a nuclear plant accident).

3. MISE DESIGN TIME

Like MISE 1.0 design approach (Rajsiri, Lorré, Bénaben & Pingaud, 2010), MISE 2.0 design approach is based on a Model-Driven Approach (MDA) (Miller & Mukerji, 2003) through an automated model transformation including (i) the collaborative characterization, (ii) the design of the collaborative process cartography, and (iii) the execution step.

3.1. The Collaborative Situation Characterization

The goal of this first step is to gather knowledge about the collaboration. For this purpose, (Mu, Bénaben, Pingaud, Boissel-Dallier & Lorré, 2011) have defined a metamodel that integrates the concepts of a collaborative situation and that allows to characterize a specific collaborative situation (like a crisis situation). The concepts covered by the metamodel are subdivided into three categories:

- The context of the collaborative situation: this is the environment of the subpart of the world concerned by the collaborative situation. In the case of a crisis situation, it allows to define the threats, the risks, the goods, the people, etc.
- The partners of collaborative situation: this part of the knowledge focuses on the public side of the partners, *i.e.* the information and the skills they want to share into the context of the considered collaborative situation.
- The goals of collaborative situation: a collaborative situation exists only to meet a need. Thereby, the goals of the collaborative situation form a critical and essential

knowledge to build the systems that allow the effective deployment of the collaborative situation.

In MISE 2.0, the metamodel is structured in layers, around a central core that represents the general concepts of the collaborative situation (Mace Ramete, Lamothe, Lauras, & Bénaben, 2012). The central core groups all the concepts related to the collaborative situation (environment, partners, goals). This core is shared by any kind of collaborative situation between organizations. However, the layers covering the core are domain specific, and even subdomain specific. They are used to define the specific concepts, which inherit from the general concepts of the core. In MISE 2.0, several layers were defined, including one dedicated to the crisis situation (see Figure 1).

The obtained collaboration metamodel is the container of the knowledge relating to the collaborative. The user will be in charge of designing instances that represent the components of the considered situation, with respect to the metamodel. A software editor will help the user to characterize the considered collaborative situation (such as the response to a nuclear plant crisis situation) and obtain a model of the collaborative situation (Rajsiri, Lorré, Bénaben & Pingaud, 2010).

3.2. The Design of the Collaborative Process Cartography

This second step aims at using the gathered knowledge about the targeted collaborative situation, in order to build a dynamic collective behaviour that meets the modelled context. The semantic gap between the characterization of the needs and requirements (in other words, the targeted crisis response) and the solutions (the business dynamic that meets the defined goals) has to be filled.

The addition of the behaviour concepts to the metamodel (as presented in (Mu, Bénaben, Pingaud, Boissel-Dallier & Lorré, 2011)) provides a direct link between the model of crisis situation characterization and the collaborative

Figure 1. The core and the crisis management layer of the metamodel. (adapted from (Mace Ramete, Lamothe, Lauras, & Bénaben, 2012))
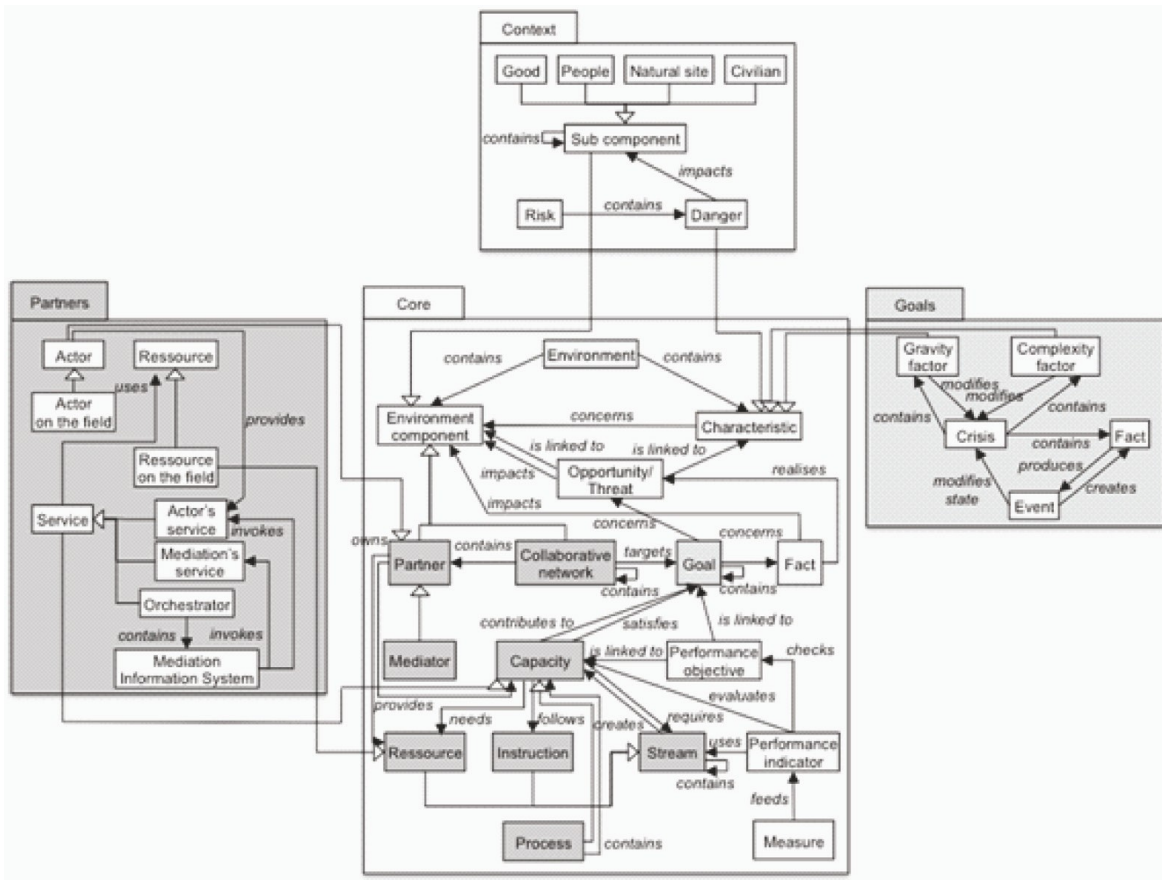
process metamodel. Through a model transformation, a process cartography can be deduced (organized according to the three abstraction levels: business, logic, technological).

In addition to this deduction, a n-to-m semantic reconciliation (as shown in Figure 2), explained in (Bénaben, Boissel-Dallier, Pingaud & Lorré, 2012) is made between the deduced business activities —at the abstract level— and the technical services offered by the partners —at the concrete level—. There is also a data transformation activity: the data's format from the response of the service A is not necessary compliant with the expected format of incoming data for service B. This transformation is performed by services based on semantic matchmaking engine, as detailed in (Bénaben, Boissel-Dallier, Lorré & Pingaud, 2010).

At the end of this step, a set of collaborative technical processes is obtained (Bénaben, Boissel-Dallier, Pingaud & Lorré, 2012).

4. MISE RUNTIME

Finally, the deduced collaborative technical processes are automatically implemented as executable workflows (this is the implementation layer).

In the remainder of this article, we will present the Figure 3, Figure 4 and Figure 5, which summarize the overall MIS's architecture of the MISE 2.0 prototype, containing three main parts (the crisis situation, the crisis cell and the event processing).

The main hypothesis of this article concerns the partners' IS that are supposed to follow the

Figure 2. Principles of N-to-M semantic reconciliation
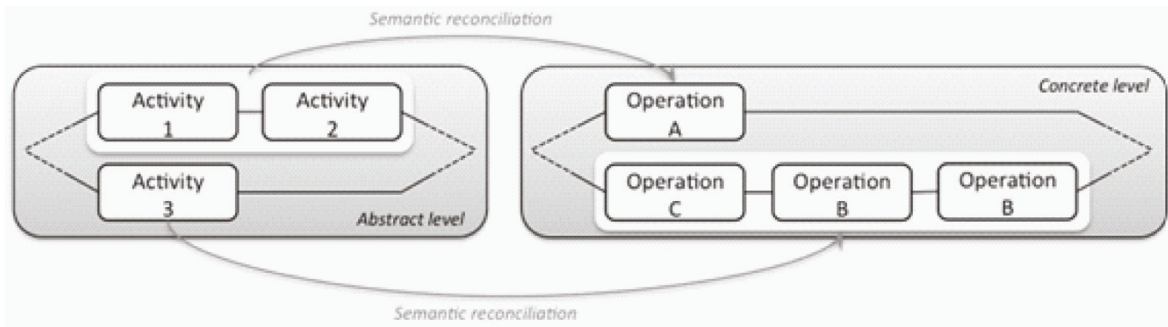
same conceptual logical architectural philoso-phy: the Service Oriented Architecture (SOA) (Vernadat, 2007). If it is not the case, we consider partners' IS may be SOA compliant with the help of interfaces between non-SOA applications and the MIS (whose deployment is based on SOA principles). The MIS will run on a server, more precisely on an Enterprise Service Bus (ESB), which is Petals ESB (developed by the French software editor Petals Link/Linagora) 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).

On Figure 3, the MISE Design Time ser-vices are shown(1): these are Model Situation (MS) service, Processes Cartography (MC) ser-vice, and Workflow definition (W) as presented 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" sec-tion). Once the collaborative workflows and the MIS are designed, they are deployed on the ESB and executed by the orchestrator component, which is a workflow engine (2). The execution of the workflows calls numerous services (3),

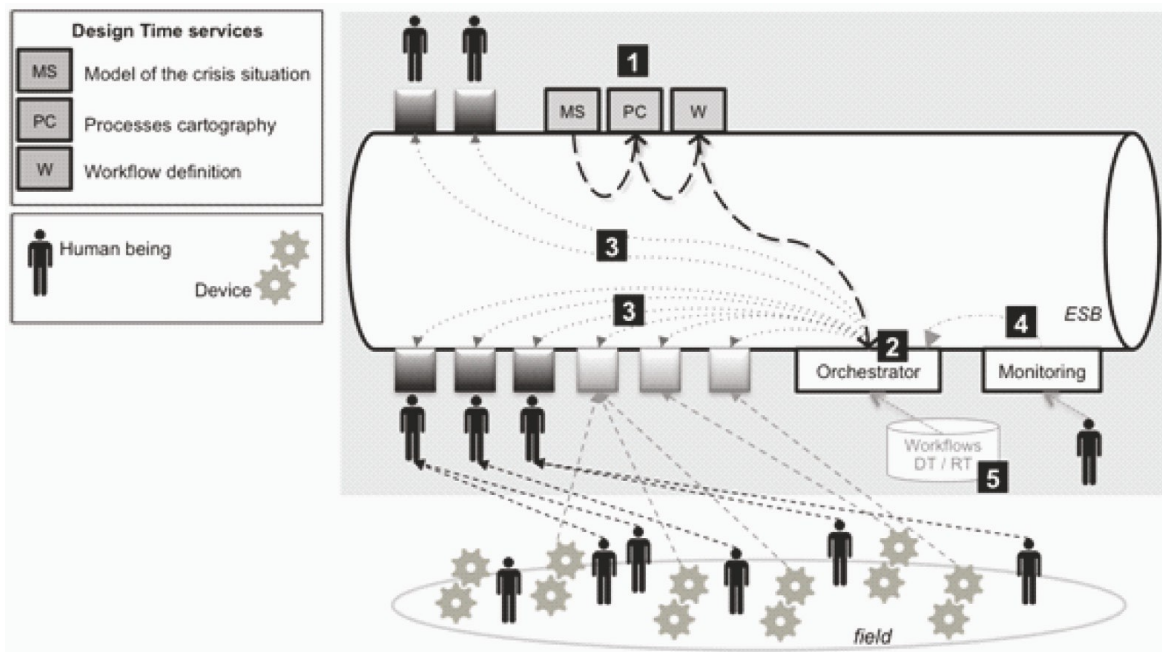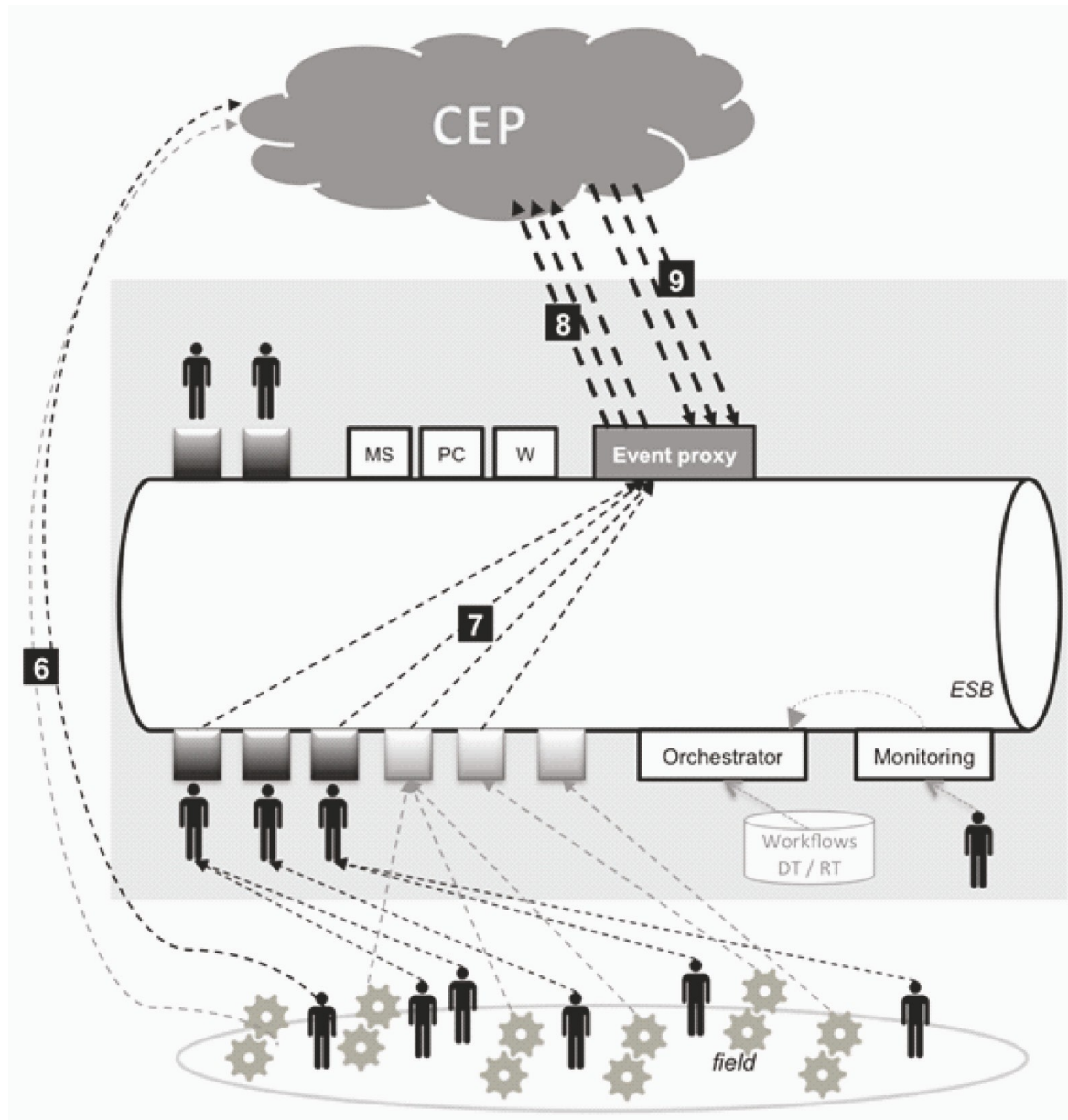*Figure 3. Global view of the MISE 2.0 platform*

Figure 4. The event-driven architecture layer of MISE 2.0 platform

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) (*i.e.* the state of each activity and resource).

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 (5) that allows feedback

about the crisis characterization and the deduced and run workflows.

5. 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 (i.e.

Figure 5. The agility service of the MISE 2.0 platform
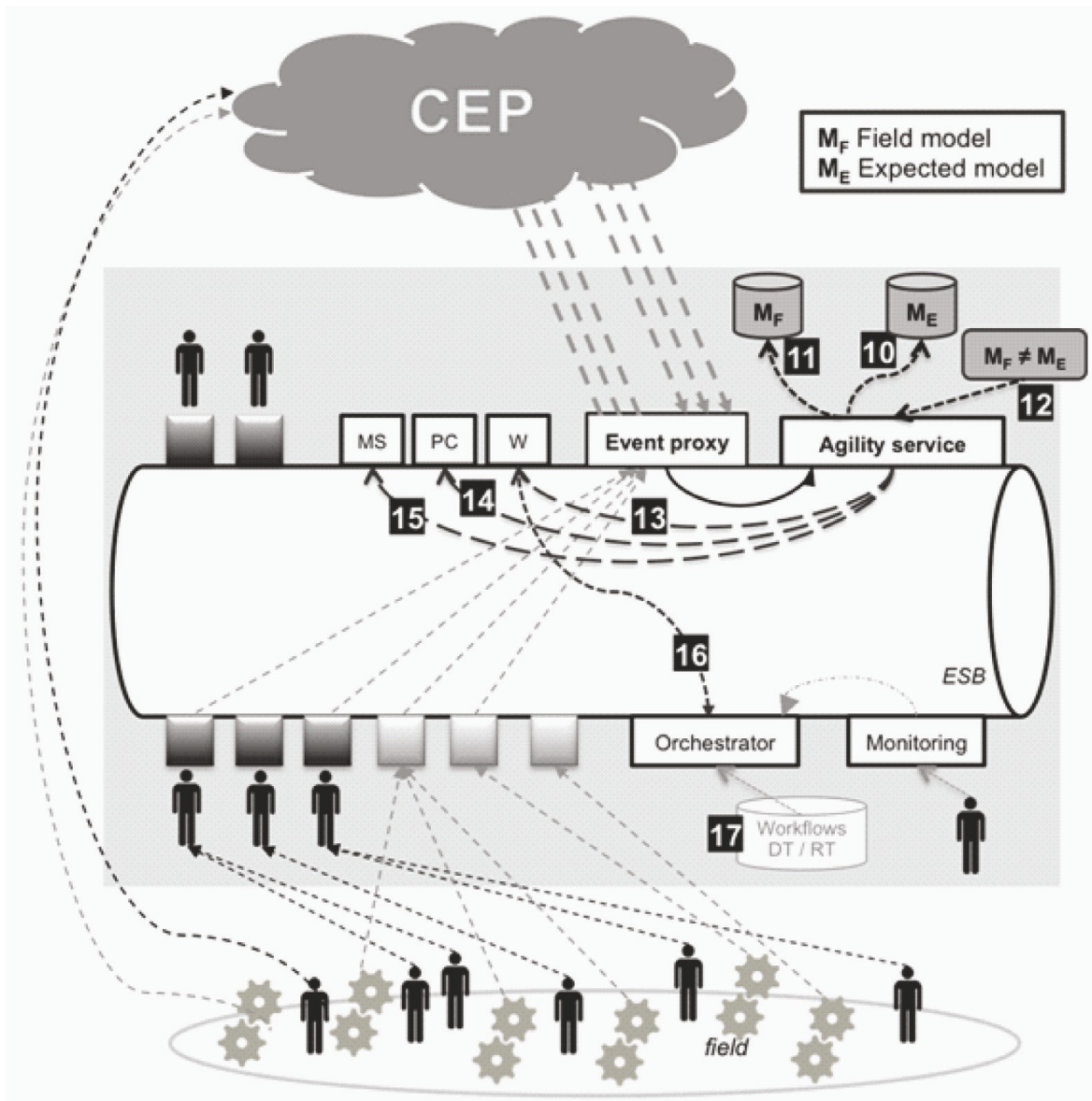
the run collaborative workflows) may be not relevant after a while.

In the literature, the notion of agility was widely discussed. (Badot, 1998) defined it as the reconfiguration of the system to satisfy a need for adaptation. For other authors like (Kidd, 1994), (Lindberg, 1990) and (Sharifi & Zhang, 1999), agility is a need for flexibility, responsiveness or adaptability. In the context of logistics, flexibility is seen as "the ability to meet shor-term changes" (Sheffi, 2004) and it is differentiated from adaptation over time in response to a change (McCullen & Christopher, 2006). Considering these notions of responsiveness (speed of adaptation), adaptation (wideness of this adaptation and detection (moment of the adaptation), we propose the following definition of agility: *agility is the ability of a subject to lead as quickly as possible, on the one hand, to the detection of its mismatch to a given context, on the other hand, to the setting up of the required adaptation.*

In our context, solving the collaborative workflows' agility issue means we need to detect

the moment when a workflow is not relevant anymore regarding to the collaborative goals and the current context of the collaborative situation (detection), and what needs to be done to deal with this issue (adaptation), as fast as possible (responsiveness).

Based on Pingaud (2009b)'s research work, three kinds of evolutions of collaborative situations can be extracted:

- 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 crisis cell: the management of the response to the crisis situation may evolve due to an evolution of the structure of the crisis cell (e.g. arrival or leaving of stakeholders, lack of resources).
- The evolution of the response to the crisis: the management of the response to the crisis situation may evolve due to (i) a dysfunction of the execution of a service (leading to the interruption of the workflow of the response), or (ii) due to a partial initial definition of the process of the response. As an example of a dysfunction, we can cite the case of the nuclear plant of Fukushima Daiichi, where the poured water had not reached the expected goal (*i.e.* to cool the reactors).

Thus any changes, any evolution, any information that could challenged collaborative processes' accuracy and relevancy have to be managed. According to (Chandy & Schulte, 2009; Etzion & Niblett, 2011; Luckham & 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 6 on Figure 4), (ii) the services used by the response workflows (7). These events are collected in an event cloud and/or sent to the CEP through an event stream.

The use of Event-Driven Architecture (EDA) principles (Luckham & Schulte., 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. A Complex Event Processing (CEP) engine is used to consume and manage these events (point 8 on Figure 4).

The CEP engine 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 & 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 considered 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 (9).

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 Agility service. In order to be agile, as agility is considered as the combination of event detection and adaptation, the MIS have to detect the events that have an impact on the crisis response and to define an adaptation regarding these events.

The Agility service's role is (i) to update both *expected model* (the planned and expected situation model at time t) (point 10 on Figure 5) and *field model* (the « what happened actually on the field» situation model at time t) (11) through the received events (respectively through the events received from the monitoring of the crisis response, and the events coming from the field).

Then the Agility service has (ii) to measure and identify the differences between these two situation models (12) and (iii) to suggest a relevant adaptation of the MIS through a complete or partial execution of the Design Time workflows.

Depending on the identified differences, a part of the response workflows may be run once again (e.g. to solve an issue concerning a technical execution that has failed) (13), or modified (e.g. another partner's service may be called to replace a service that is down) (14) or even completely redesigned (because the initial characterization of the crisis situation is incomplete or the crisis situation has considerably evolved) (15).

The adapted workflows are not only executed (16), they are also stored in order to capitalize knowledge (17). 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 Agility service allows refining the crisis response.

6. MISE 2.0 IMPLEMENTATION: CRISIS USE-CASE

6.1. The Crisis Situation

As the MIS' main objective is to be the support of collaborative situation, the focus is set on the response step: the MIS helps the stakeholders in charge of the crisis situation to deduce, via the preparedness step results, and to give the crisis a response.

The use-case is a Chemical Biological Radiological and Nuclear (CBRN) use-case: a large quantity of radioactive substance is accidentally released in the atmosphere, due to a critical accident in a French nuclear plant. To resolve this crisis, a lot of heterogeneous actors may be involved. As many stakeholders are involved in the crisis management process, contact was made with IRSN (IRSN, 2007; IRSN, 2011) and ASN (ASN, 2009), France's public service expert in nuclear and radiation risks: they have provided information about the real emergency plans and processes and the best practices.

The services provided by these actors are also diverse and varied, ranging from psychological assistance to traffic duty. This heterogeneity is probably the main cause of the

difficulty to manage such a crisis situation. But there are many other difficulties to cope with.

For instance, there are a lot of critical dependencies between the actions of these heterogeneous actors. A decision to evacuate will depend on the actual level of radioactivity measured, but also on weather forecasts and assessments of the situation in the nuclear plant.

6.2. MISE Platform Integration in the Existing System

The MISE platform is integrated with the existing system as follows (Figure 6). Numerous and various sensors are used to monitor the radiation levels, the temperature, the wind direction, etc. in a nuclear plant.

These sensors are able to emit their measures as events, i.e. they send their data through a network and the people and devices interested in these data' topics gather them. In a few words, this is the publish/subscribe mechanism.

The PLAY platform (presented previously in section "MISE Overview") is able to gather events emitted by various and heterogeneous data sources, like sensors. The PLAY platform has the role of an event marketplace: it collects events and proposes them to third part systems, like the MISE platform. The MISE platform gathers events found into the PLAY event marketplace. Based on these gathered events, the MISE platform is able to analyse the data coming from the field of the crisis situation (the nuclear plant in our case study).

An interoperability issue can be underlined: how these connections between heterogeneous systems (sensors, PLAY platform, MISE platform) are possible?

This point is solved through use of two majors standards (in an SOA context): Sensor Observation Service (SOS) (OGC, 2012) and WS-Notification (OASIS, 2007a; OASIS, 2007b; OASIS, 2007c). The SOS standard defines a web service interface to query sensor data and embedded a SOAP binding. The WS-Notification standard allows to define the mechanisms to exchange events through a mechanism of subscription to topics of inter-

Figure 6. Integration of MISE within the existing system

Nuclear plant   Sensors
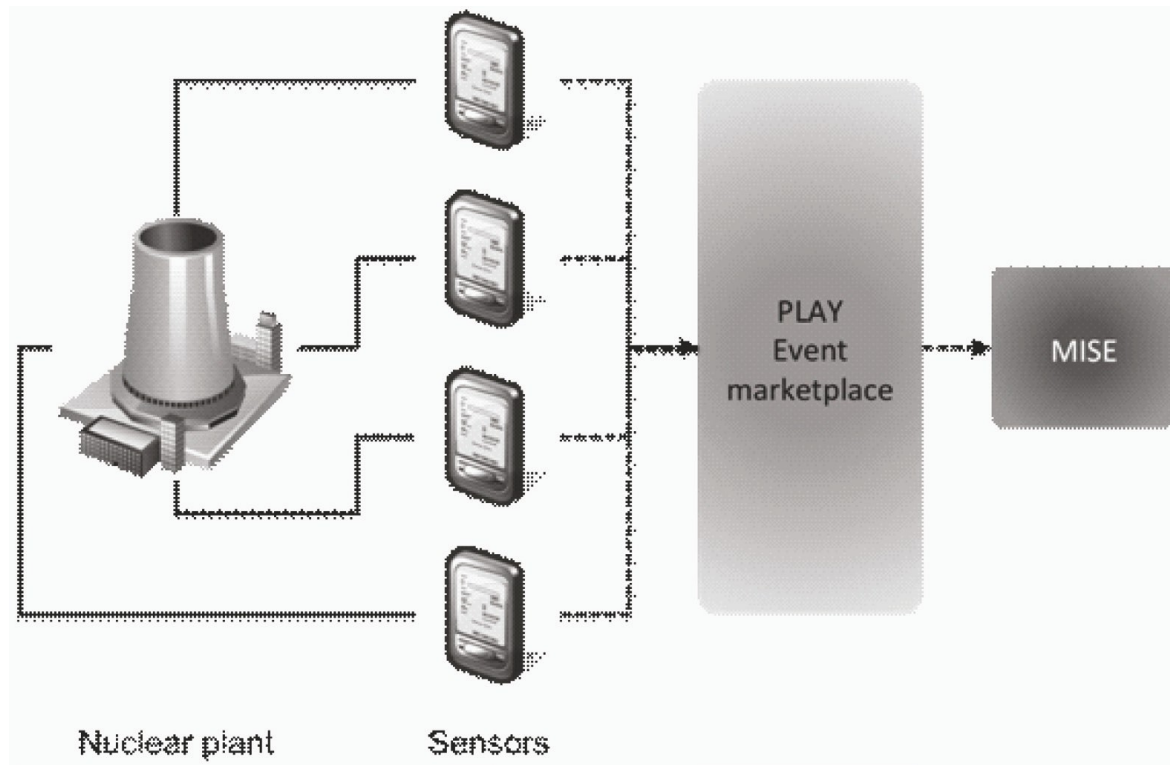
est. WS-Notification will be detailed in the remainder of this article.

6.3. Event Types (In a Crisis Situation Context)

The event type is the structure of the instantiations, represented as a collection of event attributes. As recommended by (Etzion & Niblett, 2011; Luckham & Schulte, 2008), an event type should specify certain predefined data as:

- A unique event identifier used to reference the event
- The type of the event
- The timestamp of the event's creation
- The source of creation for the event

Eight event types covering the whole kinds of events that can happen in the context of crisis situation management were listed: Measure, Alert, Demand, Offer, Resources_Status, Activity_Status, Instruction and Report. They are detailed in the UML class diagram in Figure 7.

These types are classified into four macro types of events (Situation, Resources, Activity and Consequence), all depending of one common ancestor called Event and containing two attributes (TimeStamp and UniqueID). Each event type owns its own attributes and inherits a part of them from its ancestors.

At the implementation level, (Luckham & Schulte, 2008) recommend defining the event types through one of the new computer languages like XML Schema or Java. As the MIS is based on the SOA principles, it is natural to develop the defined crisis response processes in Business Process Execution Language (BPEL) (OASIS, 2007) workflows, run by an orchestration engine. It leads to use the Web Services-Notification (WS-N) standard (OASIS, 2007a; OASIS, 2007b; OASIS, 2007c), which is based on XML formalism, to describe the event occurrences sent by/to the MIS' workflows. So the use of XML Schema to describe the eight event types is completely convenient with the previous architecture choices.

Figure 7. The eight event types concerning crisis situations
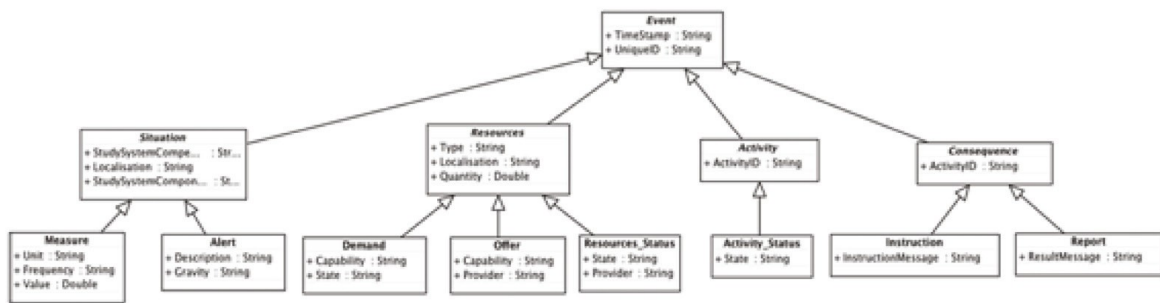
6.4. Event Managing and Business Rules

A Complex Event Processing (CEP) engine is used to consume and manage the events. Esper, an open-source CEP developed by the American software editor EsperTech has been chosen as it allows to manage events in an XML format such as WS-N format.

Events come from both the crisis response process itself and the whole devices, sensors, softwares, partners' information systems implied into the crisis situation. Through a software connector we have developed (in Java language), Esper engine can subscribe to the services that send WS-N formatted events, and so receives this event stream. Esper filters it and applies the business rules that are defined (*i.e.* event processing rules, containing event patterns) on the input event stream (these rules are manually implemented before running the engine). When an event process rule is triggered by input events, an output event is generated. It can be noted that the event types of studied event occurrences are parameterized into Esper according to the structure that is defined in the previous subsection "Event types (in a crisis situation context)". And the events generated by Esper (as a consequence of event process rules triggering) follow exactly the same structure.

This section aims at presenting a short example of such a rule extracted from the nuclear crisis use case. The aim of this rule is to gather the events from the Measure event type, and, according their values, generates an event from the Alert event type, which will be sent by Esper (and gathered by the concerned subscribers). This rule is human produced (after preparedness meetings or extracted from established emergency plans for instance), and it is written in natural language:

```
Condition (measureEvent as
wind, measureEvent as radia-
tion)
[win:time:20s]
∧ wind.measureUnit=='wind di-
rection'
∧ wind.measureValue==8
∧ wind.measureSEID=='METEOF-
TLS-NORD'
∧ radiation.
measureUnit=='Nanosievert per
hour'
∧ radiation.measureVal-
ue>110000
∧ radiation.
measureSEID=='MFRT-TLS-BLA'
Action Create
alertEvent(alertId=uniqueID(),
alertTimestamp=currentTime(),
alertDescription='Radiation
increasing over security level
in combination with a N-N-W
wind',
alertGravity='high', alertLo-
calisation=
radiation.measureLocation,
alertSEID = radiation.measure-
SEID,
componentName = radiation.mea-
sureName)
```

- MeasureEvent represent the Measure event type. It is aliased to match this event's occurrences which are coming from two different sensors (wind direction sensor and average ambient equivalent dose —*i.e.* radiation— sensor),
- MeasureUnit, measureValue, measureSEID, measureLocation, measureName are properties of the measureEvent event type,
- Win:time represents the sliding window property of our pattern
- AlertEvent is an event type. Its occurrence results from the triggering of the studied event processing rule,
- AlertId, alertTimestamp, alertDescription, alertGravity, alertLocalisation, alertSEID, componentName are properties of the alertEvent event type,
- UniqueID() calls a function generating a unique ID, currentTime() calls a function catching the current time (full date – time format).

Then this rule is parameterized into Esper, using the Esper Query Language (EQL), as shown below:

```
insert into alerts
select 'Radiation increasing
over security level in combi-
nation with a N-N-W wind' as
alertDescription,
'high' as alertGravity,
radiation.measureLocation as
alertLocalisation, radiation.
measureSEID as alertSEID,
radiation.measureName as com-
ponentName
from
measureEvent(measureUnit='wind
direction'
and measureValue=8
and measureSEID='METEOF-TLS-
NORD').win:time(20 seconds) as
wind,
measureEvent(measureUnit='Nano
sievert per hour'
and measureValue > 110000
```

```
and measureSEID='MFRT-TLS-
BLA').win:time(20 seconds) as
radiation
```

Each pattern triggering inserts an event (with the properties specified into the select clause) into the "alerts" event stream. Through a software connector, Esper is able to send this output event stream as WS-N formatted event. This output event instantiates one of the eight event types previously defined (in the presented example, it instantiates the Alert event type). This approach allows consumer services (like the Agility service or any partner service or any mediation service) to subscribe to the events generated by the CEP.

Among all events received by the MIS (including those generated by the CEP), some events are used to orchestrate activities inside a specific process, some of them are used to choreography the processes, some of them are used to update both field model and expected model. And some of them may assume two or three of these roles. Ideally, any received event will be used to update the models and eventually has a role into the orchestration/choreography function of the MIS. In the previous example, the generated Alert event will be used by the Agility service to update the field model, with the creation of a new risk in the crisis situation model.

To illustrate deeper the concept of such event roles, and their impact on the update of the field model and the expected model in the use case, the following example is presented: the temperature of the reactor of the nuclear plant is significantly increasing. The defined response in this case is to cool the reactor by spreading water on it. In the crisis response workflow in charge of this part, the concerned activity is "cool the reactor with two tanks of water". Once this activity is done, Agility service receives two events (at minima):

- One concerning the status of the activity (the expected model is updated according this event),

- One concerning the real result on the field, provided by a sensor (the field model is updated according this event).

To illustrate the proposal, only the most interesting cases of event combinatory are detailed in the following.

- **First Case:** the status of the activity informs the MIS that the activity is done with success (no technical failure, the two tanks of water were correctly spread, so the reactor is considered as cooled and the temperature as decreasing), and the result on the field of the activity informs the MIS that the temperature is decreasing. These two events lead to the same conclusion: the temperature of the reactor is decreasing. The executed response meets the current crisis situation.
- **Second Case:** the status of the activity informs the MIS that the activity is done with success, and the result on the field of the activity (coming from a sensor) informs that the temperature is absolutely not decreasing, and worse, is still increasing.
- **Third Case:** the status of the activity informs the MIS that the activity execution fails, and the result on the field of the activity informs the MIS that the temperature is absolutely not decreasing, and worse, is still increasing.
- **Fourth Case:** the status of the activity informs the MIS that the activity execution is 50% completed (only one water tank was available), and the result on the field of the activity informs the MIS that the temperature is decreasing.

We can see with these cases that after updating the two models with the received events, the role of the Agility service is to detect the divergence (here, the fact that actually, despite the execution of cooling activity, nothing changes), and also the origin of the divergence (by comparing the models as two sets of points and not in a point-to-point way). The origin may be:

- At the level of the activity itself: was there actually a failure during the execution of the activity? Were the resources enough to reach the objective? (In our example: third case)
- At the level of the activity: is this activity available? Is the activity able to meet our needs? (In our example: second case)
- At the level of the crisis characterization: has the crisis situation evolved so much that the response is not relevant to the current situation? (In our example: second and third case)

The fourth case also shows us that a divergence between the models may not lead to a negative fact. We keep in mind that this example is really simple and only helps to understand the necessity to detect the divergence but also its (multiple) causes in order to decide the strategy concerning the adaptation: (i) re-execute the activity, (ii) modify the response implementation (change a service), (iii) redesign the whole response (from the moment where the response became not relevant). This choice is not trivial, as each option has a cost considering the elapsed time (and so the evolution of the crisis that can lead to an over crisis situation) but also a cost considering the resources (and so the economics). Typically, the fourth case shows that too many resources where asked, and even a lack of 50% of water, the reactor was cooled as expected. We think that the Agility service will not only help to decide what and when we have to modify the crisis response, but also helps to study the crisis response in terms of cost, which is very interesting for the feedback, after the crisis resolution (and so helps to define best possible responses to a crisis situation).

7. PERSPECTIVES

In such a collaborative environment and in such a crisis situation, the quality of the data and its

protection is a critical issue. It is obvious that we must not allow a right of access to any data in any conditions to anybody. During the semantic reconciliation (when the partners' services are chosen to create the executable workflows) and during the event subscriptions, the MISE 2.0 platform will provide governance tools for both services and event sources. Events can only be used if they are trustable: for the moment, several approaches can be considered, like a social filter (using trust criteria on event sources and the relations between the event sources) or a repository of event sources, that gives to the users several criteria to choose the most relevant event sources.

8. CONCLUSION

People, devices, softwares, sensors, 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 inability to cover all possible connections (explosive combinatory) and their relevance at a given time.

Another perspective concerns the crisis situation modeling. For the moment, the Agility

service is only seen as a mean to detect divergences about an established crisis situation and to help to adapt the crisis response to the new situation. But, it can be also used as a detector of anomalies in a nominal system: incoming events would be used to update the model of the current situation and this updated model may be compared with a reference model of the situation (*i.e.* the model of the situation in a nominal state). According to the detected divergences (origin, nature, size) a crisis situation may be identified automatically.

ACKNOWLEDGMENT

REFERENCES

ASN. (2009). Les situations d'urgence - Autorité de Sûreté Nucléaire. Retrieved March 2011, http://www.asn.fr/index.php/S-informer/Dossiers/Les-situations-d-urgence

Atlay, N., & Green, W. III. (2006). OR/MS research in disaster operations management. *European Journal of Operational Research*, *175*(1), 475–493. doi:10.1016/j.ejor.2005.05.016

Badot, O. (1998). *Théorie de l'entreprise agile*. Editions L'Harmattan.

Barros, A., Dumas, M., & Oaks, P. (2005). A critical overview of the Web Services Choreography Description Language (WS-CDL). *BPTrends March 2005*. Retrieved November 10, 2010, from http://news.bptrends.com/publicationfiles/03-05%20WP%20WS-CDL%20Barros%20et%20al.pdf

Beamon, B. M., & Kotleba, S. A. (2004). Inventory modeling for complex emergencies in humanitarian relief operations. *International Journal of Logistics: Research and Applications*, *9*(1), 1–18. doi:10.1080/13675560500453667

Bénaben, F., Boissel-Dallier, N., Lorré, J.-P., & Pingaud, H. (2012). Semantic reconciliation in interoperability management through Model-Driven Approach. *Proceedings of PRO-VE*, *2010*, 705–712.

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*. doi:10.1080/0951192X.2012.684712

Bénaben, F., Hanachi, C., Lauras, M., Couget, P., & Chapurlat, V. (2008). A Metamodel and its Ontology to Guide Crisis Characterization and its Collaborative Management. In FiedrichF.Van de WalleB. (Eds.) *Proceedings of the 5th International ISCRAM Conference (pp.*189-196*)*.

Chandy, M. K., & Roy Schulte, W. (2009). *Event Processing, Designing IT Systems for Agile Companies*. USA: McGraw Hill.

Devlin, E. (2006). *Crisis management planning and execution*. USA: Taylor and Francis Group. doi:10.1201/9780203485897

Etzion, O., & Niblett, P. (2011). *Event Processing in Action*. USA: Manning Publications.

IRSN (2007). CODIR-PA: présentation de deux scénarios d'accident affectant une centrale nucléaire française.

IRSN. (2011). IRSN - Expert in nuclear and radiological risk. Retrieved March 2011, http://www.irsn.fr/EN/Pages/home.aspx

Josuttis, N. M. (2007). *SOA in Practice: the Art of Distributed System Design*. USA: O'Reilly.

Kidd, P. T. (1994). *Agile Manufacturing: Forging New Frontiers (Addison-Wesley.)*. Addison-Wesley.

Konstantas, D., Bourrières, J.-P., Léonard, M., & Boudjilida, N. (2005). Preface of: Interoperability of Enterprise Software and Applications. In D. Konstantas, J.-P. Bourrieres, M. Leonard, & N. Boudjlida (Eds.), *I-ESA'05 Interoperability of Enterprise Software and Applications*. Berlin, Germany: Springer.

Lagadec, P. (1992). *La gestion des crises*. Paris: Ediscience International.

Lindberg, P. (1990). Strategic Manufacturing Management: A Proactive Approach. *International Journal of Operations & Production Management*, *10*(2), 94–106. doi:10.1108/01443579010001221

Luckham, D., & Schulte, R. (2008). Event Processing Glossary – Version 1.1. *Event Processing Technical Society*. Retrieved March, 2011, from http://www.ep-ts.com/component/option,com_docman/task,doc_download/gid,66/Itemid,84/

Mace Ramete, G., Lamothe, J., Lauras, M., & Bénaben, F. (2012). A road crisis management metamodel for an information decision support system. In *Proceedings of the 6th IEEE International Conference on Digital Ecosystems and Technologies*. Campione d'Italia, Italy: IEEE. doi:10.1109/DEST.2012.6227934

Maréchaux, J.-L. (2006). *Combining Service-Oriented Architecture and Event-Driven Architecture using an Enterprise Service Bus (Tech. Rep.)*. Canada: IBM Software Group.

Marterer, R., Matthias, M., & Koch, R. (2012). An architecture for distributed, event)driven systems to collect and analyze data in emergency operations and training exercises. In RothkrantzL.RistvejJ.FrancoZ. (Eds.), *Proceedings of the 9th International ISCRAM Conference*.

McCullen, P., & Christopher, R. S. and M. (2006). The F1 Supply Chain: Adapting The Car To The Circuit-The Supply Chain To The Market. *Supply Chain Forum, 7-1*(12). Retrieved from www.supplychain-forum.com?num=12&art=105

Michelson, B. M. (2006). *Event-Driven Architecture Overview: Event-Driven SOA Is Just Part of the EDA Story (Tech. Rep.)*. Boston, USA: Patricia Seybold Group. doi:10.1571/bda2-2-06cc

Miller, J., & Mukerji, J. (2003). MDA Guide Version 1.0. Retrieved from https://www.assembla.com/spaces/genr/documents/dWOVGkzSir35s9e-Je5cbCb/download/MDA.pdf

Mu, W., Bénaben, F., Pingaud, H., Boissel-Dallier, N., & Lorré, J.-P. (2011) A Model-Driven BPM Approach for SOA Mediation Information System Design in a Collaborative Context. In JacobsenHans-ArnoWangYangHungPatrick (Eds.), *Proceedings of the IEEE 8th International Conference on Services Computing (pp.* 747-748*)*. doi:10.1109/SCC.2011.66

OASIS. (2006). Web Services Base Notification 1.3 OASIS Standard. *OASIS*. Retrieved October, 2012, from: http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf

OASIS. (2006). Web Services Brokered Notification 1.3 OASIS Standard. *OASIS*. Retrieved October, 2012, from: http://docs.oasis-open.org/wsn/wsn-ws_brokered_notification-1.3-spec-os.pdf

OASIS. (2006). Web Services Topics 1.3 OASIS Standard. *OASIS*. Retrieved October, 2012, from: http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-os.pdf

OASIS. (2007). Web Services Business Process Execution Language OASIS Standard Version 2.0. *OASIS*. Retrieved October, 2012, from: http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf

OGC Open Geospatial Consortium. (2012). OGC Sensor Observation Service Interface Standard, retrieved February, 2013, from https://portal.open-geospatial.org/files/?artifact_id=47599

Peltz, C. (2003). Web Services Orchestration and Choreography. *Computer*, *36*(10), 46–52. doi:10.1109/MC.2003.1236471

Pingaud, H. (2009). Prospectives de recherche en interoperabilité: vers un art de la médiation. In *Plenary Lecture, 8th International Congress on Industrial Engineering*. Tarbes, France.

Pingaud, H. (2009). Rationalité du développement de l'interopérabilité dans les organisations. In P.-M. Riccio & D. Bonnet (Eds.), *Management des technologies organisationnelles* (pp. 19–30). Paris, France: Presses de l'Ecole des Mines de Paris.

Rajsiri, V., Lorré, J.-P., Bénaben, F., & Pingaud, H. (2010). Knowledge-based system for collaborative process specification. *Integration and Information in Networked Enterprises*. *Computers in Industry*, *61*(2), 161–175. doi:10.1016/j.compind.2009.10.012

Sharifi, H., & Zhang, Z. (1999). A methodology for achieving agility in manufacturing organisations: An introduction. *International Journal of Production Economics*, *62*(1), 7–22. http://www.sciencedirect.com/science/article/pii/S0925527398002175 doi:10.1016/S0925-5273(98)00217-5

Sheffi, Y. (2004). Demand Variability and Supply Chain Flexibility. Entwicklungspfade und Meilensteine moderner Logistik (pp. 85–117). Gabler Verlag; Retrieved from http://link.springer.com/chapter/10.1007/978-3-322-89044-3_5 doi:10.1007/978-3-322-89044-3_5

Truptil, S., Barthe, A.-M., Bénaben, F., & Stuehmer, R. (2012). Nuclear Crisis Use-Case Management in an Event-Driven Architecture. In F. Daniel, K. Barkaoui, & S. Dustdar (Eds.), *Lecture Notes in Business Information Processing* (pp. 464–472). Berlin, Germany: Springer. doi:10.1007/978-3-642-28108-2_45

Vernadat, F. (2007). Interoperable enterprise systems: Principles, concepts, and methods. *Annual Reviews in Control*, *31*(1), 137–145. doi:10.1016/j.arcontrol.2007.03.004

Yu, B., & Cai, G. (2012). Coordination of Emergency Response Operations via the Event-Based Awareness Mechanism. In RothkrantzL.RistvejJ.FrancoZ. (Eds.), *Proceedings of the 9th International ISCRAM Conference*.

Anne-Marie Barthe-Delanoë is a postdoctoral research fellow in the Interoperability of Organizations (IO) team in the laboratory of Industrial Engineering of the Mines Albi-Carmaux, since the end of 2013 when she received her PhD in Industrial Engineering and IT. She also graduated in 2007 as an engineer and got a M.S. in Automated, IT and Decision-making Systems from the Mines Albi-Carmaux. Her research works deal with the interoperability of organizations and the agility of collaborative processes, especially in a crisis situation. Prior to joining the IO team, she served first as a research engineer at EBM WebSourcing (a French SOA software editor), then as a consultant at Cameleon Software (a French product configurator software editor).

Sabine Carbonnel was a research engineer in the laboratory of Industrial Engineering of the Mines Albi-Carmaux. She graduated as a doctor in IT from INSA Rennes in 2005: she worked on « integration and modellng of linguistic knowledge for online handwriting recognition ». Her research interests focused on artificial intelligence. After six years serving as a research engineer in the domain of industrial engineering, she is now an engineer in the Information System Department of the French Ministry of Education.

Frédérick Bénaben is an Assistant-Professor in the IO team at the laboratory of Industrial Engineering of the Mines Albi-Carmaux, since the end of 2003. After a master's degree in computer science (1998), he worked on a PhD on "model checking for properties verification and validation on heterogeneous optronic systems" with THALES Optronics and University of Montpellier (2001). After a post-doc period at the Ecole des Mines Alès, he joined the laboratory of Industrial Engineering of the Mines Albi-Carmaux in order to work on Information Systems and interoperability of enterprises. Member of several research associations (Pole GSO of InterOp V-Lab, CNRS GDR-MACS) he works particularly on mediation information system engineering to support inter-organizational interoperability. Involved in several national and European projects, he is aiming at joining Business Process Management approaches and model-driven design to achieve interoperability of organizations.

Hervé Pingaud is graduated as engineer in 1983 and as doctor in 1988 from the Institut National Polytechnique de Toulouse (INPT) with a PhD on process dynamic simulation. Then he worked as a full time researcher on process simulation and control engineering in INPT. From 1992 until 1999, he had an increasing interest for industrial engineering. He acted as a project leader in developing new diploma in industrial engineering both in INPT, and in Mines Albi-Carmaux that he joined for a new position of full time Professor in 1999. From 1999 until 2010, in the laboratory of Industrial Engineering of Mines Albi-Carmaux, he supervised PhD students in three main scientific areas: enterprise process modelling, process and risk management, and interoperability in information system design. Since 2010, he is the Director of Champollion University (Albi, France). He is a French representative of IFAC TC 5.3 "Enterprise Networking and Integration", and used to act as one of the CNRS group coordinator on this subject at a national level.