



HAL
open science

Towards the agility of collaborative workflows through an event driven approach

Anne-Marie Barthe-Delanoë, Aurelie Montarnal, Sébastien Truptil, Frederick Benaben, Hervé Pingaud

► **To cite this version:**

Anne-Marie Barthe-Delanoë, Aurelie Montarnal, Sébastien Truptil, Frederick Benaben, Hervé Pingaud. Towards the agility of collaborative workflows through an event driven approach. International Journal of Disaster Risk Reduction, 2018, 28, p.214-224. 10.1016/j.ijdr.2018.02.029 . hal-01724575

HAL Id: hal-01724575

<https://imt-mines-albi.hal.science/hal-01724575v1>

Submitted on 10 Jul 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards the agility of collaborative workflows through an event driven approach – Application to crisis management

Anne-Marie Barthe-Delanoë^{a,*}, Aurélie Montarnal^a, Sébastien Truptil^a, Frédérick Bénaben^a, Hervé Pingaud^b

^a Mines Albi - Université de Toulouse, France, Route de Teillet, Campus Jarlard, 81013 Albi CT Cedex 09, France

^b University Jean-François Champollion, France, Place de Verdun, 81000 Albi, France

ABSTRACT

Organizations are more and more involved into collaborations to face globalization, crisis situations or market demand. Consequently, they are strongly dependent on their collaborative processes and matching workflows, which have to be relevant and efficient through time. As collaborations and their context are now fast evolving, agility of collaborative workflows is now an unavoidable requirement. Several solutions have been developed to deal with this issue, but they often focus their monitoring on a single aspect of the collaboration and/or impose a single adaptation strategy to face the detected evolutions. This article describes a platform to support decision-making about the adaptation strategy to lead by: (i) taking into account and exploit relevant data gathered from the collaboration itself and its environment (context awareness); (ii) proposing relevant adaptation strategies to correct or improve the collaborative behaviour. Based on the solutions proposed by the platform, decision-makers can choose the most appropriate adaptation tool to perform the selected adaptation strategy. As crisis responses are specific cases of collaboration in an unstable context of execution (short time to react, highly evolving environment), a proof of concept based on the Fukushima Daiichi nuclear disaster is developed to illustrate the use of the developed platform.

Keywords:

Agility
Adaptation
Collaborative workflow
Complex Event Processing
Crisis response
Service Oriented Architecture

1. Introduction

It is now acknowledged that organizations (companies, institutions, administrations or other) take part into collaborative situations to face globalization, crisis situation, market demand [1] or any other unstable environment. As a consequence, they are strongly dependent on their collaborative processes (and their matching workflows): they have to be efficient and relevant through time. Considering the instability of the collaborative environment (strengths and weaknesses evolve, goals change, partners leave or join the collaboration), agility of collaborative workflows becomes an unavoidable requirement. This agility requirement for collaborative workflows can be expressed as follows: 1) how to detect the instant where the collaborative workflows do not match with the current situation (and what are the causes); 2) how to redefine the best possible collaborative behaviour (at a concrete level, the best workflows and their orchestration and choreography) at time t , in a very short time (close to real time for specific contexts like crisis situations), depending of the current situation, the collaborative workflows progress, the state of the involved resources and collaborative

partners.

In the remainder of this article, crisis response management domain will be the common theme for the presented examples. Crisis situation response is a specific case of collaboration among organizations in an unstable environment. Their context of execution is moving, unsure and highly sensitive to their evolving environment (a tsunami after an earthquake, pyroclastic flows after an eruption, pandemic into a crowded refugee camp, etc.). In addition, the network of stakeholders involved into the response is also evolving (arrival/departure of partners, denial of service, etc.) and the execution of activities can fail. Plus, the time allowed to react on changes is very short by the nature of the objectives of such collaborations. Moreover, it is necessary to underline that the human factor can also dramatically slow down the process in analysing, combining or even exchanging data during disasters [2] underline that in the Fukushima disaster management, the main issue, far beyond the technical problems caused by both the earthquake and the consequent tsunami, are the information avoidance behaviours that are related to personal and organizational cultures. This phenomenon was also observed in many major natural or industrial disasters

* Corresponding author. Present address: Laboratoire de Génie Chimique, Université de Toulouse, CNRS, INPT, UPS, 4 allée Emile Monso, 31030 Toulouse, France.

E-mail addresses: anne-marie.barthe@mines-albi.fr, annemarie.barthe@ensiacet.fr (A.-M. Barthe-Delanoë), aurelie.montarnal@mines-albi.fr (A. Montarnal), sebastien.truptil@mines-albi.fr (S. Truptil), frederick.benaben@mines-albi.fr (F. Bénaben), herve.pingaud@univ-jfc.fr (H. Pingaud).

according to [3].

In the official report of the Fukushima Nuclear Accident Independent Investigation Commission [4], it is underlined many times that the lack of information sharing (the right information to the right people at the right time), and as a consequence, of a model of the current situation, led to misunderstandings between the operator company and the government, despite past trainings: “In terms of accident response this time, as shown above, it was problematic that there was confusion in the chain of command for the station and that it ended up being an impractical response organization where persons, who did not understand field conditions, were making decisions from places that did not have that information” [5]. This information failure led to late and inconsistent evacuation orders [6].

Our proposal here intends to automate the situational awareness during a crisis response in order to propose relevant adaptations of the ongoing response processes.

The presented research works aspire to solve these issues at the Information System level, by (i) recalling a definition of collaborative workflow agility, (ii) presenting a brief overview of existing solutions to support the choice of a relevant adaptation, and (iii) proposing a methodology to support the choice of a relevant adaptation solution. Then, (iv) the implementation of the presented methodologies based on Event-Driven Architecture principles is presented, illustrated by a use case based on the Fukushima disaster, before concluding.

2. Agility as a requirement for collaborative situations

Being able to efficiently and fluently participate into opportunities of collaborations is nowadays a strong requirement for organizations. This simple statement actually involves a lot of correlated expectations: organizations must be able to design collaborative networks, to join existing ones, and to adapt their behaviour (on a business point of view but on a technological point of view as well). Considering these elements from the collaborative network point of view, the unavoidable requirement for organizations to relevantly and appropriately take part into collaborative situations must be compared with collaborative network lifecycle.

From this perspective, the research approach of this article, based on (and presented in) [7], aims at covering this lifecycle through three main objectives underlying this necessity of collaboration and directly covering the steps of collaborative situations lifecycle:

- **Definition:** Any collaborative network first needs to be designed and settled up. This implies several subtasks such as *identification of collaborative objectives, selection of relevant partners, definition of tasks and planification*, etc. This first objective is supposed to be reached by the *creation* step of the collaborative network lifecycle.
- **Realization:** Once the collaborative network is defined and structured (on both static and dynamic points of view), it is crucial to support its operational phase. Actually, all previous steps such as the selection of partners or the definition of tasks are not sufficient to warranty the adequate performing of the emerging collaborative situation. This objective implies to proactively orchestrate and drive the collaborative situation according to the results of the definition level. This second level is supposed to be reached by the *operation* step of the collaborative network lifecycle.
- **Maintaining:** Obviously, a collaborative situation may evolve. As presented in [8] there are three kinds of sources of adaptation: (i) the motivations of the collaboration (context, and/or objectives) are no longer the same and require a new understanding of the collaborative situation, (ii) the collaborative network of partners is no longer the same (arrival or departure of partners or even unavailability of capabilities) and requires a redefinition of the collaborative behaviour, and (iii) there might be a dysfunction of some tasks (even if the collaborative network and the associated behaviour are pretty relevant) requiring to execute them again or find palliative operation.

The phases related to the definition and realization of the collaborative behaviour have already been studied, in particular through the works led by Mu et al. [9] and Boissel-Dallier et al. [10] respectively. However maintaining it by adapting it to the evolving situation still remains a challenge that we aim to tackle in these works. It strongly requires from the collaborative network to be able to react adequately and to perform agility.

In this sense, agility is a crucial concept in a collaborative situation. Rao et al. [11], Kidd [12], Lindberg [13] and Sharifi and Zhang [14] introduce the boundaries between this concept and adaptability, flexibility and reactivity. In logistics, flexibility is seen by Sheffi [15] as “the ability to meet short-term changes”. McCullen and Christopher [16] differentiate flexibility from adaptation over time in response to a change. The concept of agility is based on three concepts: the system must be able to change its structure (flexibility) according to a relevant analysis of the situation and its requirements (efficiency) and this should be done in a hurry (reactiveness). In the context of this article (and as presented in Benaben et al. [7]), these three aspects of agility have been regarded according to two orders: the first order represents the functional level of agility while the second order concerns the non-functional level of agility. Consequently, agility has been defined, on first order, as the capacity of a system to (i) detect any evolution that requires changes and (ii) adapt effectively the system. Regarding the second order, two other attributes may be considered: first, the dynamicity of agility (i.e. reactiveness) might be crucial and second, the relevance of the detection and adaptation (i.e. efficiency) may also be critical. Consequently as presented in Benaben et al. [7], this vision may be simply and roughly formulated as:

$$Agility = (detection + adaptation) \times (reactiveness + efficiency)$$

This qualitative formula is a structuring scheme that allows the study of agility to be partitioned according to these four properties. Detection and adaptation may be considered as the main attributes of agility while reactiveness and efficiency are second order attributes (related to detection and adaptation). In the presented research works, it has to be noted that the agility of the collaborative workflows should be led on-the-fly, during the collaborative workflows execution, and the adaptation step can focus on either realization level (re-execution of part/whole collaborative workflows) or definition level (redefinition of part/whole collaborative processes) as detailed in Section 4. Following the taxonomy of the workflow flexibility approaches defined by Schonenberg et al. [17], the approach of workflow agility we presented here relies on mixing flexibility by deviation and flexibility by change (as explained in Barthe-Delanoë et al. [18]).

Over the two first order attributes, the theory about the detection part of agility has already strongly been presented and discussed in Barthe-Delanoë et al. [18]. For this specific reason, the remainder of this article will be focusing on an adaptation feature of collaborative networks, within the frame of the overall approach presented in Benaben et al. [7] and the implementation of the agility (at the technical level) through a proof of concept. The following Fig. 1 illustrates this overall framework:

The aim in the presented works is to automatically determine what kind of adaptation to perform (and why) rather than how to perform it. Finally, considering crisis management, it is also crucial to notice

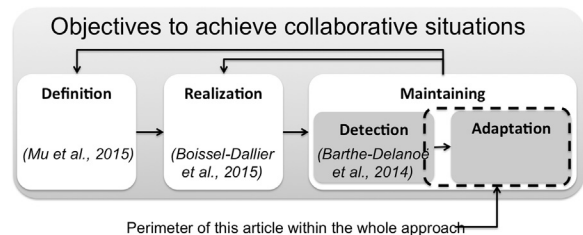


Fig. 1. Positioning of the current article in the research framework.

that this application domain drastically embeds the agility requirement. It is a matter of fact that crisis contexts are by nature unstable (over-crisis, unknown and unexpected events, continuous evolution, strong disruptions, etc.) and require agility.

3. Literature review: choosing adaptation strategy for collaborative workflows

3.1. Existing approaches

Many approaches are discussed in the literature about agility of collaborative workflows, such as uncertainty management in workflows systems, in particular, adaptable IOW (Inter Organizational Workflows) [19], the ADEPTflex approach [20], and the research works of van der Aalst [21].

Loose IOW aim to define collaboration and to find candidates (organizations) to collaborate (candidates are not necessary known). The partners can change during IOW runtime. Andonoff et al. [22] present an implementation of such IOW in a collaborative environment. Multi agents approach is in charge of agility and use two ontologies to search and select the candidate organizations.

The ADEPT project developed a management system of adaptable workflows [23]. It allows drifting pre-defined workflow: a user can modify instances of this workflow on the fly, by adding or removing activities or defining a new sequence of activities. A first commercial solution of ADEPT (and of its 2.0 version ADEPTflex), called AristaFlow BPM Suite was released in 2008 [24].

The ATAPIS Project is focused on time-aware processes and proposes to adapt the workflows during run-time to unforeseen events under specific time constraints [25]. It inherits from the previously cited AristaFlow, and provides a study of change operations directly conducted on the workflow and how to still keep the temporal constraints of the process according to the changes' effects.

The ProSit Method is proposed by [26] and proposes situation-aware mechanisms to adapt process execution, when deployed in a dynamic environment. The resulting situation-aware workflow can then adapt itself to specific and expected changes in the environment. The method basically implements a translation from situation-unaware to situation-aware workflows. For this, it is based on the recognition of pre-defined situation-aware fragment models inside a workflow, which are in turn substituted to the corresponding situation-unaware parts of this workflow.

Another approach is based on Worklets [27]. A Worklet is an atomic and autonomous workflow that manages one specific task inside an activity. Worklets are available into a repository and are chosen depending on the context of execution of the workflow. This approach is a late binding one, according Schonenberg's taxonomy [17]: the process structure is defined once for all (definition step) but its implementation as a workflow is done during the runtime (realization step). This approach has been realized into the YAWL (Yet Another Workflow Language) environment [28].

Case Handling is another approach, developed by van der Aalst et al. [29] that manages activities by focusing on their exchanged data instead of focusing on the sequence of activities (i.e. the process itself). In this approach, activities to perform are chosen according the available data at a time t and not depending on the last performed activity [30]. The CHS (Case Handling System) gives the user all available data at a time t , and orchestrates activity runtime in accordance with these data. CHS has been implemented into FLOWer [31], since renamed as BPM|One as a commercial software solution.

Solutions that aim to cover any possible deviation of workflows by preparing alternative branches are excluded. They are not realistic into the crisis management context for obvious reasons (combinatorial explosion) as crisis situation are particular kinds of unstable environments.

On the implementation side, a few commercial products and

research projects attempt to provide agility to collaborative workflows. On the one hand, BonitaSoft (a suite of tools to design, execute and monitor processes) and the Architecture of Integrated Information Systems (ARIS) tools are the major commercial products. ARIS' approach manages workflow adaptation only in a deterministic manner [32]. On the other hand, several research projects like the WORKPAD project [33] and the CRISIS project [34,35] set the focus more on recovering the disconnecting nodes through specific tasks (WORKPAD project) or supporting collaboration into crisis situation and on exploring decision-making under conditions of uncertainty (CRISIS project). The European project PRONTO allows to fuse data from various sources and analyse it to extract useful information [36]. If it can support detection of new risks as a way to provide context awareness, and data share among stakeholders, it does not focus on the workflow management part. The European project PLAY proposes an adaptation recommender service [37]: it allows adapting the on-going processes on pre-determined milestones, through the addition of relevant pieces of pre-existing processes (extracted from a knowledge database) [38] proposes a Situation-Aware Dynamic Service Coordination that detects and adapts processes in the domain of surgery, using an Event-Driven Architecture to cope with the amount of data generated by the Internet of Things. But the proposed adaptation is strongly domain dependent regarding the detection conditions and is based on pre-determined process alternatives. Despite interesting features and approaches to solve the workflow agility issue, none of them allows dynamically and automatically to both detect a mismatch in a continuous way and adapt the collaborative processes according the mismatch.

We can see that, to our knowledge, there are plenty of research works and commercial solutions to conduct adaptation at a pre-determined moment and/or at a specific level (change a partner, change an activity, etc.) as summed up into Table 1. But there is no solution allowing to automatically detect a deviation and support the decision makers in their choice of the kind of adaptation to perform, depending on the detected deviation of the workflow or of its environment and independently from a specific adaptation tool.

3.2. Information system architectures to support agility

On the technical side, we also investigate the way to implement agility. As explained previously, collaborative workflows are subject to evolutions due to the context where they are executed or the effects of their execution itself. Thereby it is crucial to identify and react on such evolutions: on the one hand, the collaboration has to be aware of the changes of its execution context and on the other hand, it has to monitor the collaborative workflows in progress. Thus, any evolution, any change, any data, any information that could challenge the relevance of the collaborative workflows have to be managed. In a few words, there is a need for context awareness. This need underlines the necessity of bridging the gap between the Information System (IS) and the things that happened in the physical world, as explained by Zang et al. [39].

The notion of *event* defined by Chandy and Schulte [40], Etzion and Niblett [41] and Luckham and Schulte [42] as an occurrence, and any particular embedded data, and of the associated Event-Driven Architecture (EDA), can provide a support to automatically retrieve real-time and historical data about both workflows and their context of execution. Events are produced by the people and the devices on the collaboration field (as shown on Fig. 2), and also by the services invoked by the collaborative workflows.

The choice of the architecture is also led by the need for adaptability. Service Oriented Architecture (SOA) principles are used to implement the collaboration itself, through a Mediation Information System (MIS) in charge of the interoperability of the partners' ISs (as introduced by Verginadis et al. [43]). The combination of both EDA principles and SOA principles into an ED-SOA has been widely discussed in the literature. As stated by Josuttis [44], Luckham [45],

Table 1
Overview of existing solutions to provide agility to collaborative workflows.

Product/Project	Detection	Adaptation	Reactivity	Effectiveness
AristaFlow	No	Yes (manually done)	No	Yes (relevancy rules)
ATAPIS	No	Yes (based on change operations, at workflow level only)	Yes	N/A
ProSit Method	No	Yes (automated on pre-determined alternatives, in design-time)	N/A	N/A
ARIS	Yes (automated, event-driven)	Yes (automated and pre-determined alternatives)	N/A	N/A
BPM One	No	Yes (manually done)	No	No
BonitaSoft (Community Edition) (Cheng et al. 2017)	No	No	No	No
	Yes (pre-determined conditions)	Yes (pre-planned conditions and actions)	Yes	Yes (as both conditions and actions are planned)
Loose IOW	No	Yes (at the software protocol level only)	N/A	N/A
TIBCO	Yes (manually done)	Yes (manually done)	No	Yes (performance indicators)
YAWL	No	Yes (pre-planned alternatives)	Yes	N/A
CRISIS	Yes	Yes (partial adaptation)	Yes	N/A
PRONTO	N/A	No	Yes	No
PLAY	Yes (pre-planned milestones)	Yes (pre-planned alternatives)	Yes	No
WORKPAD	Yes	No	Yes	No

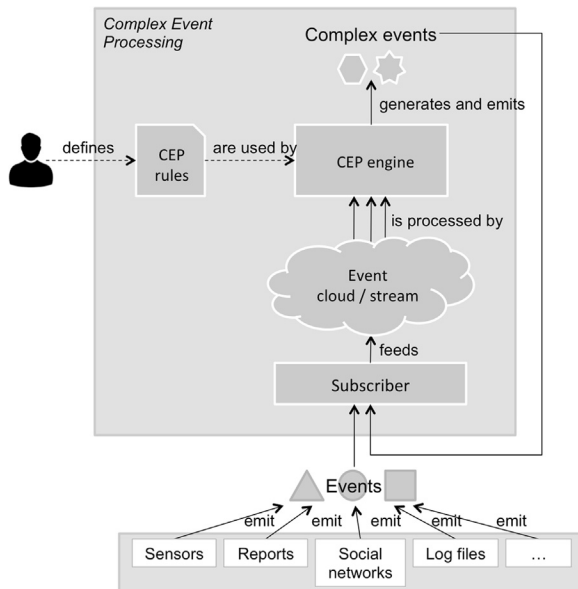


Fig. 2. Principles of Complex Event Processing.

Maréchaux [46] and Michelson [47], we agree that EDA should not be seen as a competitor to SOA but as additional principles to complement SOA principles. Adding an EDA layer to the existing SOA allows the system to gather knowledge about the collaborative situation, the collaborative environment and the collaborative workflows in almost real-time by taking events into account, thus making the IS context aware.

But EDA is not only about managing event exchanges between workflows, people and machines: it also concerns the business level by filtering and applying business rules to detect relevant events or combinations of events. Two (or more) events (called simple events), which are not seen as risks or opportunities when viewed separately, may have a different meaning if they are considered together (and so they create a complex event). For example, the combination of events embedding a temperature measure of three locations (quite nearby) can lead to emit a fire alert event. This event aggregation and derivation is called Complex Event Processing (CEP) (as shown in Fig. 2), which is technically carried out by a CEP engine [41]. It is interesting to note that the CEP is also able to detect and analyse events resulting from a combination of simple events. These so-called complex events are detected through the permanent execution of pre-defined business rules (called CEP rules) among simple events coming from data streams that are feeding the event cloud (Fig. 2). As a result combined EDA and CEP can

ensure real-time and permanent information availability. Plus, the time dimension is also taken into account with CEP: business rules can include time window, meaning only events during a given time frame will be taken into account. For example, a time window can be used to calculate the average value of a temperature on a specific location during the last ten minutes.

One may argue that the CEP is “yet another expert system”. But CEP is not about decision making but about creating knowledge from events in real-time, as underlined by Fülöp et al. [48], and making this knowledge available for people and systems in charge of decision making through the emission of complex events.

Moreover, the amount of emitted events is growing as our world tends to be more and more connected to become an Internet of Things (IoT) [49]. Analysis of such an amount of various events is a very complex task if made by solely human beings, in terms of time criteria and available human resources, and bridging the gap between IoT and context awareness is still one major gap as identified by [50]. For example, [38] use EDA to support context awareness by processing events into complex events that are meaningful regarding their application domain (surgery). The ability of the system to take this analysis in charge is also an important point to support the agility of the workflows, considering the time dimension of the given agility definition.

The other interesting characteristic of EDA is its ability to provide very loose coupling between applications (web services of the partners and of the MIS here) through the publish/subscribe mechanism as described into the Web Services Base Notification Standard of OASIS [51]. Applications subscribe to a certain type of event and not to a specific source providing this type of event. In fact, the focus is set on the subject of interest to retrieve events. We can imagine that two web services provide the same kind of information, through the publication of the same event type A. Other web services subscribe to the event type A. If a third provider enters the network and publishes events of event type A, the subscribers will receive them without any technical modification or interface creation. This ability fits completely with our need for an agile structure, especially for an application to crisis management.

The main characteristics of EDA are summed up in Table 2.

The use of an event-driven approach combined with SOA allows our system to monitor the changes as they happen, whether during the execution of the collaborative workflows or in the environment of the collaboration, in a real-time perspective.

4. Proposed adaptation approach

In this section, we will focus on the adaptation part of the developed Agility Service platform, where strategies to adapt the workflows (by

Table 2
Main characteristics of EDA.

Properties	Description
Loose coupling	Producer has no knowledge of the event's subsequent processing, or the interested parties
n-to-m communication	Proactive publish/subscribe mechanism where one produced event may have impact on several consumers
Event triggering	Control is done by consumer on the base of the received events
Asynchronous	Asynchronous operations possible through events

taking into account the detected differences) are proposed to the decision makers. First, we will describe the inputs coming from the detection step, then available adaptation solutions and finally the mechanisms that led to propose one or several solutions to the decision makers.

4.1. Inputs from detection step

Considering that the detection step was already described and discussed [18], a brief reminder is done here about the results of this step.

The collaborative situation is represented with a meta model inspired by Macé-Ramète's road crisis situation meta model [52] as it fits most of our needs to represent the studied system (using concepts of goods, people, civilian society, natural sites, characteristics, risks, consequences, etc.), and the treatment system (actors and their services). Two models of the same collaborative situation are so realized using the same meta model:

- The field model that represents the collaborative situation from a field point of view (only data incoming from the field are used to feed the model),
- The expected model that represents the collaborative situation from a workflow monitoring point of view. The progress of workflows and states of activities feed the model.

Then these models are continuously compared: a risk was added (it exists only into the field model), an activity failed during its execution, a partner left the collaborative network (it exists in the expected model but not in the field model), etc. The whole difference δ is the sum of all the detected atomic differences δ_i between the field model and the expected model. If δ value is over a threshold (defined by the decision makers) the adaptation step is triggered.

4.2. Various adaptation solutions

The choice of the kind of adaptation to perform is highly bounded to the nature of the collaborative situation evolution, as presented by Pingaud [53] and Camarinha-Matos and Afsarmanesh [8]:

- Evolution of the context of the collaboration.
- Evolution of the network of partners (including their capabilities and resources).
- Failure during execution or non-compliant results.

These evolutions can be associated to the three levels of the collaborative workflow design [7] and can lead to:

- Redo the characterization of the situation (on which the collaborative processes are defined) based on the field model (users do not start from scratch at this step): as the situation has evolved a lot, it is necessary to go back to the very first step of the process design by defining the studied system, threats and opportunities, involved actors (as described into Mu's research works [9]).
- Redesign collaborative network: based on the defined process, it is

necessary to find other actors and/or activities to obtain the matching workflows (based on semantic reconciliation system, as developed by Boissel-Dallier [10]).

- Re-execute one (or more) activity of the running workflows.
- The evolution of both the collaborative network and the crisis situation itself (leaving/arrival of partners, removal of a risk, addition of resources, etc.) questions the choices made to define, design and execute the collaborative processes. Evolutions concerning partners and their capabilities can have negative or on the contrary positive impacts on the studied system:

- Negative impacts: typically an activity invoked into the collaborative workflows is missing (the partner does not offer it anymore). Thus, it is necessary to find a surrogate activity (or set of activities) that can achieve the pursued goals (e.g.: reduce a risk, prevent a danger).
- Positive impacts: the arrival of a new partner into the collaboration implies the offer of new activities. One (or more) of these new activities may be more relevant regarding the collaboration objectives than the ones called in the running workflows. It could be interesting to switch these activities in order to fulfil the collaborative goals with a higher efficiency.

In addition, the evolution of the collaboration itself can also challenge the relevancy of the deployed processes. Sometimes, it is possible to call activities that are now useless or to ask for disproportionate resources. It covers cases like sending a team to rescue wounded people who do not eventually exist, call a team of fire fighters specialized in forest fire to extinguish a fire that is ultimately banal, ask for a demolition unit to bring down a building that collapsed in the meantime.

Moreover, the Agility Service has to propose several solutions of adaptation, considering the inputs from detection step and the impacts of the detected evolutions on the running workflows. As a decision support system, its role is to underline the actions to perform and to help the decision makers, not to make the decision.

These facts lead us to consider the adaptation solutions as a set of six ranks of adaptation (Table 3). First, the adaptation has to be done in order to correct (rank 1.x) or to improve the collaborative processes (rank 2.x). Secondly, given by decreasing priority order of collaborative workflow adaptation design actions, adaptation concerns the characterization of the collaboration (rank x.1), the collaborative network (rank x.2) or the execution of the collaborative workflows (rank x.3).

When adaptation solutions are suggested to the decision makers, they are sorted by priority: corrective rank takes precedence over improvement rank, as characterization rank takes precedence over collaborative network and execution ranks. For example, a proposal of adaptation "correct by re execution" (rank 1.3) takes precedence over a proposal of adaptation "improve the collaborative network" (rank 2.2).

When an adaptation is suggested, complementary information is given: for instance, in the case of a re execution of a set of activities, the unique ID numbers of the concerned ones are given in order to support the decision making and to allow the realization of the adaptation itself through dedicated tools.

Table 3
Adaptation ranks.

Adaptation level	Purpose of adaptation	
	Correction	Improvement
Characterization	1.1	2.1
Collaborative network	1.2	2.2
Re execution	1.3	2.3

4.3. Offering adaptation solutions

Knowing the available solutions of adaptation is not sufficient. Indeed, the main challenge is now to fill the gap between the calculated difference δ and the above adaptation ranks, i.e. how to offer one or more adaptation solutions based on the analysis of the calculated δ .

It is important to note that the recommendation is domain dependent, due to the strong bounds with the collaborative situation meta-model. Concepts and relations among concepts are not necessarily the same for an industrial partnership, a crisis situation, a nuclear crisis situation, a road crisis situation, etc. As the need of adaptation is detected by comparing the two models of the collaborative situation, the detected differences are indeed meta-model dependent and so domain dependent. In this paper, the algorithm defined to offer adaptation solutions is related to the civil crisis management domain.

As a reminder, δ is the sum of all detected atomic differences δ_i between the field model and the expected model. A difference δ_i is composed of two variables: cost and importance. The cost variable represents the cost of the operation (addition, removal, change) made on an instance of a concept of the field model (e.g. removal of a risk) to be isomorphic with the expected model. The importance variable adds a weight to the instances of each concept of the model, giving them more or less impact on the global calculated difference δ . In brief, only the cost variable gives information about the origin and the concept impacted for the detected difference δ_i . Considering that adaptation solutions are based on the nature of the difference δ , i.e. (i) evolution of the collaboration itself, (ii) evolution of the collaborative network, (iii) failure during execution, it can be underlined that the focus is then set on the involved concepts (People, Natural Site, Goods, Civilian Society, Gravity Factor, Complexity Factor, Risk, Consequence, Actor, Service [52]) and operations rather than on the involved instances themselves to determine the most relevant kind of collaborative situation evolution.

It can be tempting to describe all the existing combinations of {concept, operation} tuples and to match them with the six possible ranks of adaptation. But the combinations are far too numerous, even for our crisis situation meta model which contains only ten classes (People, Natural Site, Goods, Civilian Society, Gravity Factor, Complexity Factor, Risk, Consequence, Actor, Service) and for the three operations (addition, removal, change). This leads to at least 59,049 existing {concept, operation} tuple combinations.

In order to reduce this explosive combination, we have studied the relationships among the classes of the crisis situation meta model (formalized as a UML class diagram). Goods, People, Civilian Society and Natural Site inherit from the Sub Component class of the Studied System package. The same applies to Actor_On_Site (which also inherits from Actor) but this class is part of the Treatment package: Actor and Actor_On_Site have a different behaviour compared to the Goods', People's, Civilian Society's and Natural Site's one. A Sub Component may have a Characteristic, from which Complexity Factor and Gravity Factor inherit. A Characteristic may be related to a Risk. A Risk impacts a Sub Component and is realized through a Consequence, which has an effect on a Sub Component. An Actor provides a Service that respond to a Risk and/or a Consequence. A Characteristic forbids the use of a Service.

Considering this meta model, we can highlight some meta classes or High Level Classes (HLC) (a kind of meta model of the considered meta model), as shown in Fig. 3:

- HLC "Sub Component" of the studied system, containing Goods, People, Civilian Society, and Natural Site.
- HLC "Disruption", that impacts the HLC "Sub Component". It contains Characteristic, Risk, Consequence, Complexity Factor and Gravity Factor.
- HLC "Service" that includes only the Service class. This is due to the particular type of this class because a Service must be re-executed if

need be. Its behaviour in the system is different from the Actor's one: they cannot be in the same HLC even if they share strong bounds (e.g. the removal of an Actor leads to the removal of the Service he/she provides). HLC "Service" responds to the HLC "Disruption".

- HLC "Partner" responds to the HLC "Disruption" and provides HLC "Service".

The three editing operations (addition, removal, change) have been applied to these four HLC. The resulting impacts on the collaborative situation have been studied (in terms of evolution of the collaborative context, evolution of the collaborative network or failure during the execution of the activities). Moreover, the study of the links among the HLC ("impacts", "provides", etc.) enabled to find the minimum requirements to propose an adaptation solution.

So we have determined seven generic families of combinations of {HLC, operation} tuples that lead to one of the six ranks of adaptation, as shown in the following algorithm:

AdaptationChoice(delta, field_model)

This algorithm proposes adaptation solutions to the decision-maker.

Parameters:

delta, the global difference calculated by comparing field model and expected model (containing all δ_i elements)
field_model, the file containing the field model

Variables:

choice, the list containing the suggested adaptation solutions as a 3-tuple: adaptationLevel (string), adaptationPurpose (string), adaptationArgs (object)

Begin

choice \leftarrow {}

If (delta contains an added or changed HLC Disruption element) **or** (delta contains a changed HLC SubComponent element **and** its linked HLC Disruption elements are removed) **Then**

choice \leftarrow add("characterization", "correct", field_model)

If (delta contains a removed HLC Disruption element **and** delta contains a removed HLC SubComponent element **and** this involved HLC Disruption element impacts this HLC SubComponent element) **or** (delta contains an added HLC SubComponent element without being impacted by any HLC Disruption element) **Then**

choice \leftarrow add("characterization", "improve", field_model)

If (delta contains a removed HLC Disruption element) **or** (delta contains an added HLC Actor element) **or** (delta contains an added HLC Service element) **Then**

choice \leftarrow add("network", "improve", field_model)

If (delta contains a removed HLC Service element) **Then**

If (delta contains a removed HLC Disruption element) **and** (this removed HLC Service element responds to this removed HLC Disruption element) **Then**

choice \leftarrow add("network", "improve", field_model)

Else

choice \leftarrow add("network", "correct", field_model)

Endif

If (delta contains a removed HLC Actor element) **Then**

If (delta contains a removed HLC Disruption element) **and** (delta contains a removed HLC Service element) **and** (the removed HLC Service element is provided by the removed HLC Actor element) **and** (the removed HLC Service element responds to the removed HLC Disruption element) **Then**

choice \leftarrow add("network", "improve", field_model)

Else

choice \leftarrow add("network", "correct", field_model)

Endif

If (delta contains a changed HLC Actor element) **Then**

choice \leftarrow add("network", "correct", field_model)

AdaptationChoice(delta, field_model)

```

If (delta contains a changed HLC Service element) and (delta contains
an added or changed HLC Disruption element) and (the changed
HLC Service element responds to the added or changed HLC
Disruption element) Then
choice ← add(“execution”, “correct”, get(changed HLC_Service
element ID))
Else
choice ← add(“characterization”, “improve”, field_model)
EndIf
Return choice
End

```

Once δ is analysed through the AdaptationChoice algorithm, the adaptation solutions contained into the *choice* list are sorted by decreasing priority (following the adaptation ranks described in Table 3). This sorted list is then proposed to the decision makers. Depending on the chosen adaptation solution, dedicated technical tool will be opened to help the decision maker:

- To redesign the collaborative characterization (based on the data contained into the current field model).
- To redesign the collaborative network (based on the data contained into the current field model).
- To re-execute the identified activities (based on the retrieved ID numbers).

The aim of such algorithm is to help the decision maker to identify the best level of adaptation to perform by analysing all the evolutions that happened into the collaborative situation and highlighting the most appropriate solutions.

5. Technical platform implementation and results

In this part, the focus is set on the technical realization of the proposal, the platform called Agility Service. One can note that detection technical realization is also presented here (as only theory was exposed previously [18]). This is motivated by the fact that detection and

adaptation are quite symbiotic parts: adaptation cannot be achieved without detection; detection makes sense only if it takes into account up-to-date and adapted elements. A nuclear crisis example, which has been realized on the platform, is used to illustrate the technical implementation and obtained results.

5.1. EDA to allow context awareness

5.1.1. Event types

As explained above, the context awareness is assumed by ED-SOA. For this purpose, a typology of eight event types covering all the possible kinds of events that can happen in the studied system and its environment is defined. Like the meta model used to characterize the collaborative situation, the proposed typology is domain dependent: here, a crisis situation event typology is presented. Fig. 4 presents four meta types of events (Situation, Resources, Activity and Consequence), all depending on one common ancestor called Event and containing two attributes (TimeStamp and UniqueID, meeting the minimal requirements about event characteristics [41]). Each event type (Measure, Alert, Demand, Offer, Resources_Status, Activity_Status, Instruction and Report) owns attributes and inherits a part of them from its ancestors.

These event types are used to subscribe to event sources such as sensors, devices, social networks or workflow monitoring, and to process the collected events through the CEP engine.

5.1.2. ED-SOA technical choices

Luckham and Schulte [42] recommend defining the event types through a language like XML Schema or Java. Keeping in mind that the architecture is ED-SOA, the choice of the technical format for event exchange is naturally guided to use the Web Services-Notification (WSN) standard (OASIS [50,54,55]), which is based on XML formalism, to describe the event occurrences sent by the sensors and received by their subscribers (including the CEP engine instance(s) dedicated to the considered crisis situation).

The chosen open source CEP is Esper, developed by the American software editor EsperTech [56], as it natively proposes to manage XML formatted events.

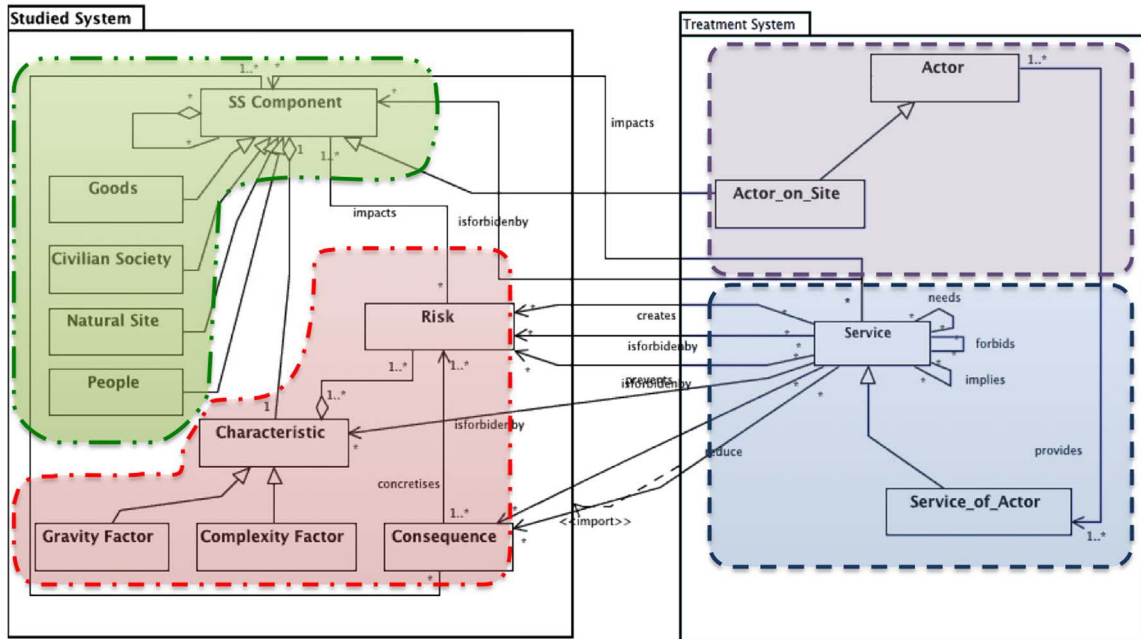


Fig. 3. High Level Classes: Subcomponent (up left), Disruption (down left), Partner (up right), Service (down right).

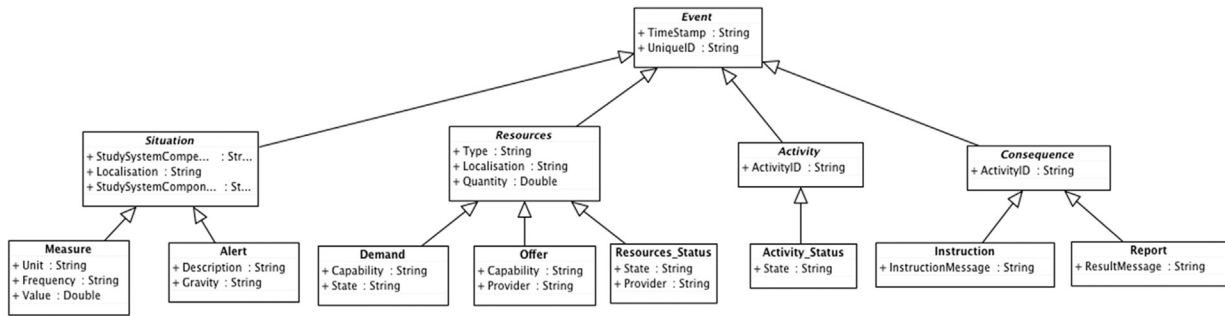


Fig. 4. Class diagram of the eight event types for crisis situation management.

5.2. Implementation of the proposal

5.2.1. Fukushima Daiichi disaster based example

As a proof of concept, an example based on the Fukushima Daiichi nuclear plant accident was developed. The data emitted by the nuclear plant's sensors (wind speed, wind direction, γ -ray and neutron radiation) were retrieved in a report available on the TEPCO's website [57] (TEPCO is the nuclear plant owner). These sensors, located in several places and measure points (MP-1, MP-8, Main Gate, etc.) of the nuclear plant, are simulated by Web Services we have developed. The data types and values emitted by these Web Services are the same as the real data emitted by the sensors of the plant during the nuclear accident. One example of such data, emitted as a Measure Event simulated by our system and received into the implemented CEP can be seen in Fig. 5.

The use-case takes place during the first hours of March 12th, 2011, less than 24 h after the tsunami and the nuclear crisis breakdown.

5.2.2. Automated models update (detection step)

Regarding the comparison method described into Barthe-Delanoé et al. [18], events are used to track the changes inside the collaborative situation model as follows:

- The expected model is fed by updating the initial model with complex events generated on the base of monitoring events. The timeline of the real performed crisis response is used to simulate the response process and to feed the expected model.
- The field model is obtained by updating the initial model with complex events generated on the base of events emitted by the field.

The CEP engine subscribes to various sources of events and listen/analyse incoming events thanks to a set of rules. These rules, defined by the decision makers (based on their past experience, feedbacks and emergency plans) allow to react on incoming events and to create new events that, once gathered by the Agility Service through a publish/subscribe mechanism, are used to automatically update both situation and field models.

For instance, the following sample business rule (Fig. 6) generates

```
<?xml version="1.0" encoding="UTF-8"?>
<measureEvent xmlns="http://www.mines-albi.fr/nuclearcrisisevent"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.mines-albi.fr/nuclearcrisisevent
  file:eventGlobal.xsd">
  <uid>465143465244444313</uid>
  <timestamp>2011-03-11T21:00:00+09:00</timestamp>
  <unit>km/h</unit>
  <frequency></frequency>
  <value>0,4</value>
  <situation>
    <localisation>Main entrance</localisation>
    <componentSEID>Wind_speed_0009</componentSEID>
    <componentName>Wind speed sensor</componentName>
  </situation>
</measureEvent>
```

Fig. 5. Example of an event emitted by a wind speed sensor, during the Fukushima Daiichi disaster.

an alert event about a new risk as the radiation in ambient air is increasing too fast or beyond a threshold (combined with wind speed) is impacting the 20-km radius area from the measure locations.

It is interesting to note that in such rule, the sensors used to gather the measures of radiation rate have not to be named identified in the business rule. The only requirement is to identify the relevant topics and pieces of information (i.e. wind speed during the last two minutes, rates of radiation in ambient air in various locations of the nuclear plant during the last minute) to get them through the events emitted by the sensors the CEP engine has subscribed. The sensor itself does not matter (as long it is supposed that the CEP engine subscribes to relevant and trustable event producers): the information is sensor-independent but topic-dependent. This also means that sensors can be added in the monitoring network (e.g. through mobile units) without requiring changing the CEP business rules to take them into account.

This CEP rule is triggered if the radiation level measured in several places on the nuclear plant is over 4 μ Sv in the last two minutes, combined with light windy conditions. In this case, the Esper CEP engine emits an alert event. The Agility Service then gathers this alert in order to update the field model thanks matching rules. The data emitted by the sensors (i.e. events) are filtered by the CEP engine. In our example, around 7:00 a.m., incoming events (i.e. real sensor data simulated by the Web Services) trigger the previously detailed business rule. As a consequence, Esper CEP engine creates a "global contamination 20-km radius area" event. The type of this event is the alert type. The Agility Service (which subscribes to the alert event type) gathers it and uses it to automatically update the field model of the crisis situation. Therefore, a new risk is added into the field model (as shown on Fig. 7), after having checked that this risk (and its links to other instances) does not exist yet into the model. The alert is maintained (and so the added risk) around 10:00 a.m. by new events coming from the field of the crisis situation.

5.2.3. Field model and expected model comparison (detection step)

Concerning the implementation of the models comparison, as our models are written in XML (Fig. 8), the retained methodology is the use of algorithms for XML tree comparison (like the ones presented in Demaine et al. [58], Pawlik and Augsten [59]).

But these algorithms do not really meet our requirements that are: (i) looking for similarity (the order of sibling nodes does not matter in our comparison) and (ii) a full usable report of the detected difference (s) between both models (for each detected difference: its type –removal, addition, change- and the concerned element). Finally, we have adapted a tool used to check the quality of XML transformations called XMLUnit [60] as it fits the needs described previously. XMLUnit is a software tool designed for XML unit testing in Java programs. It allows comparing XML documents against expected outcomes. For our use, XMLUnit is called to compare the field model against the expected model. This means when a difference is detected by the XMLUnit difference engine and identified as a:

- Removal operation: the concerned element exists into the expected

```
String combinedEventStreams7 =
    "insert into alerts " +
    "select 'Global contamination increasing.' as alertDescription, " +
    "'high' as alertGravity, 'Daiichi plant' as alertLocalisation, 'Contamination_20_km' as alertSEID, " +
    "'Contamination population' as alertName, 'risk' as alertType, 'people_habitants' as alertImpactedSEID from " +
    "measureEvent(measureUnit='m/s' and measureLocation='MainEntrance').win:time(120 seconds) as wind," +
    "measureEvent(measureUnit='microSv/h' and measureLocation='MainEntrance').win:time(60 seconds) as airMain, " +
    "measureEvent(measureUnit='microSv/h' and measureLocation='MP-8').win:time(60 seconds) as airMP8 " +
    "having avg(airMP8.measureValue)>4 and avg(airMain.measureValue)>4 and avg(wind.measureValue)>1.9";
```

Fig. 6. Sample of a business rule implemented into Esper CEP engine. (screenshot from the Java class implementation).

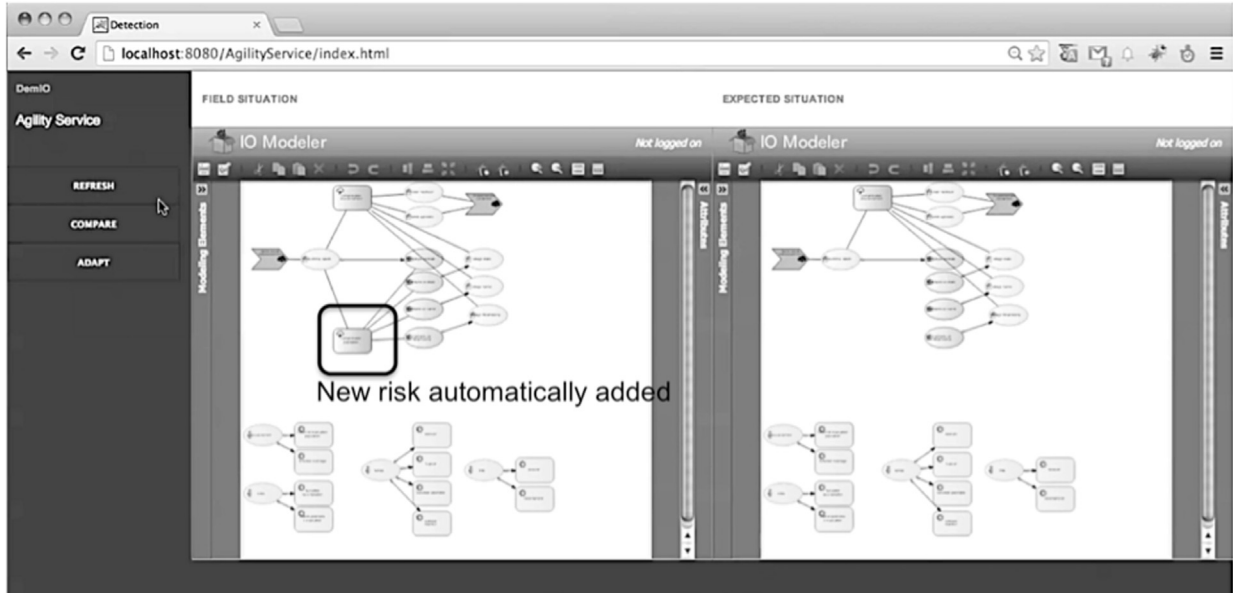


Fig. 7. Screenshot of the crisis situation modeller (thin client). Left side: field model; right side: expected model.

```
<situationModel>
  <risks>
    <risk id="R_000126" description="Temperature increasing" gravity="HIGH" localisation="Plant 1"/>
  </risks>
  <services>
    <service id="S_000467" description="Cool reactor" state="WAITING"/>
  </services>
  <linkService_Risks>
    <linkService_Risk id="LSR_128" serviceId="S_000467" riskId="R_000126"/>
  </linkService_Risks>
</situationModel>
```

Fig. 8. Short sample of situation model XML file (field model here).

COMPARISON RESULTS

Calculated delta (=7.0) is over the threshold (=1.0).

Type	Nb. of instances	Weight (%)
risk	1	28.571428571428573
link	5	71.42857142857143

Details for "risk"

Operation	Details
added	risk Contamination_20_km has been added

Fig. 9. Screenshot of the detection step result of the Agility Service: a new risk has been added into the field model and the difference between field and expected models has been detected.

- model but not into the field model.
- Addition operation: the concerned element exists into the field model but not into the expected model.
- Change operation: the value of one or more attributes of an element (existing in both models) has been modified.

However, XMLUnit algorithm does not take into account the specific nature of attributes typed with the standard type ID. As a reminder, such an attribute has a value that is a unique identifier: it cannot appear more than once in the XML document. In our case, each element of our models (partner, service, risk, etc.) is annotated with such ID typed attribute. But XMLUnit does not interpret the value of an attribute assigned with the ID type as a unique one. This implies the detection of meaningless differences between our models and to finally lead to a wrong analysis of the existing distance between our models. To solve it, we have completed XMLUnit diff engine behaviour with an additional software layer. Once XMLUnit has retrieved all the differences, this additional layer searches for differences typed as "change operation" concerning the ID typed attribute named *id* into the difference report. If such a difference is found between two elements (one with *id*="A0" from expected model, one with *id*="B0", from field model), the following actions are triggered to clean and correct the difference report:

- A difference identified as a removal operation concerning element with *id*="A0" is created and added to the difference report.
- A difference identified as an adding operation concerning element

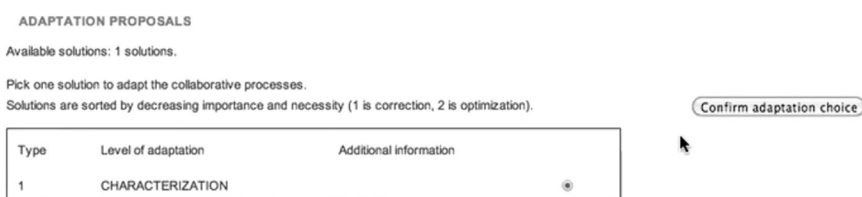


Fig. 10. Screenshot of the adaptation step result of the Agility

Service (thin client).

with id = "B0" is created and added to the difference report.

- The difference identified as a change operation is removed from the difference report.

Then, once the report of the detected differences is cleaned and corrected, the analysis of the detected differences can be run. It is interesting to note that the comparison step is domain independent. Regardless the considered situation (B2B collaboration, nuclear crisis management, road crisis management, industrial plant survey, etc.), the algorithm remains accurate.

After the comparison step of both expected and field models (Fig. 9), it appears that the contamination risk only exists into the field model. A difference is detected, and δ is important enough to exceed the threshold.

5.2.4. Adaptation step

When the detection of differences is exceeding the threshold, an adaptation proposal is triggered through the algorithm defined in the previous section. This algorithm is implemented using Java language.

In our example, the Agility Service advises the decision makers to redefine the collaborative response (i.e. redefine the collaborative processes) as no activity of the current workflows can respond to this risk (Fig. 10). The Agility Service redirects the decision makers automatically to the tools they use to design the collaborative processes, i.e. the crisis response.

As a result of this automated addition of a new risk into the situation model and the consequent proposed solution to adapt the response by redefinition of the collaborative process (helped with the collaborative process editor proposed by [9] or any other software tool assuming this feature). Based on this information, the stakeholders could decide to adapt the on-going crisis response process by adding an activity to evacuate people living within a 20-km radius area from the nuclear plant at the end of the morning of March 12th, 2011.

This simulation based on real disaster data and timeline seems to be quite trivial, as automatically gathering and aggregating data through a simple business rule to automatically add a risk in the crisis model appears as an easy task to perform. But, while facing the amount of emitted data, the number of stakeholders and the human factors to cope with in such situations, being able to automatically filter, gather, infer and share information about both the crisis situation and the crisis response can help stakeholders to get an accurate view of the crisis situation and support them to take decisions to adapt their response processes.

6. Conclusion

It is now acknowledged that the collaboration between heterogeneous organizations is a crucial issue for their survival into an unstable environment, especially in a crisis management context. The proposal within these research works assumes that their collaboration is based on the interoperability of their Information Systems, conducted through a Mediation Information System. It supports the execution of the collaborative processes between the collaboration partners. However, the unstable nature of crisis situation and of its environment may compromise the relevance of the crisis response to achieve the stakeholders' goals and the proper execution of the corresponding workflows.

Taking into account the emergence of new constraints during the runtime of the workflows led us to focus on the collaborative process agility. Two major steps are necessary to ensure the agility of collaborative workflows: (i) the detection of an evolution that compromises the pursuit of the collaborative goals through the deployed workflows; (ii) the deduction of the behaviour to adopt to overcome the detected evolution and achieve objectives whatever happens, within a very short time. Combining the principles and strengths of both Event-Driven Architecture and Service Oriented Architecture not only allow us to ensure an agile architecture but also provide the required context awareness to the Agility Service.

One can note that the adaptation step is strongly domain dependent. For each domain layer of the meta model, new High Level Classes have to be determined and matches between these HLC and the six ranks of adaptation have to be done. A way to infer knowledge about HLC and their relationships to the adaptation ranks should be studied, in order to facilitate the use of the defined proposal for new business domains.

Considering that overall proposal, two reproaches might be done: (i) what about the robustness of such a system (considering how fragile are communication networks are, especially in crisis context)? And (ii) how to deal with the constraint of data trust (event content, event sources)? Concerning the first point, such architecture is just as vulnerable as another one: it is as strong as the physical system is strong. Considering the software part, the second point is the subject of some studies towards trust [61] and confidence into data and information sharing [62], especially into the domain of Big Data and Internet of Things. Performance and scalability tests of the Agility Service are also planned in future works.

Acknowledgements

These research works have been partially funded by the French Research Agency (ANR) regarding the research project SocEDA (SOCial Event Driven Architecture) [Grant ANR-10-SEGI-013] and the European Commission under Seventh Framework Program (FP7) regarding the research project PLAY (Pushing dynamic and ubiquitous interaction between services Leveraged in the Future Internet by Applying complex event processing) [Grant FP7-258659].

SocEDA aims to provide dynamic and adaptive workflows to collaborative situations through EDA and CEP. PLAY aims to develop and validate an elastic and reliable architecture for dynamic and complex, event-driven interaction in large highly distributed and heterogeneous service systems.

The authors would like to thank the project partners for their advice and comments regarding this work.

References

- [1] S.E. DeGroot, T.G. Marx, The impact of IT on supply chain agility and firm performance: an empirical investigation, *Int. J. Inf. Manag.* 33 (6) (2013) 909–916.
- [2] A. Thatcher, A.C. Vasconcelos, D. Ellis, An investigation into the impact of information behaviour on information failure: the Fukushima Daiichi nuclear power disaster, *Int. J. Inf. Manag.* 35 (1) (2015) 57–63.
- [3] D. Chernov, D. Sornette, *Man-made Catastrophes and Risk Information Concealment: case Studies of Major Disasters and Human Fallibility*, Springer, 2015.
- [4] The National Diet of Japan, The official report of The Fukushima Nuclear Accident Independent Investigation Commission, The National Diet of Japan, Jul. 2012.
- [5] TEPCO, Fukushima Nuclear Accident Analysis Report, Tokyo Electric Power Company, Inc., Japan, 2012.

- [6] T. Hatanaka, S. Yoshida, M. Ojino, M. Ishii, The communication of information such as evacuation orders at the time of a nuclear power station accident, *Jpn. Med Assoc. J.* 57 (5–6) (. 2014) 293–319.
- [7] F. Benaben, W. Mu, N. Boissel-Dallier, A.-M. Barthe-Delanoë, S. Zribi, H. Pingaud, Supporting interoperability of collaborative networks through engineering of a service-based Mediation information system (mise 2.0), *Enterp. Inf. Syst.* 9 (5–6) (2015) 556–582.
- [8] L.M. Camarinha-Matos, H. Afsarmanesh, Collaborative networks: a new scientific discipline, *J. Intell. Manuf.* 16 (4–5) (2005) 439–452.
- [9] W. Mu, F. Bénaben, H. Pingaud, A methodology proposal for collaborative business process elaboration using a model-driven approach, *Enterp. Inf. Syst.* 9 (4) (2015) 349–383.
- [10] N. Boissel-Dallier, F. Benaben, J.-P. Lorré, H. Pingaud, Mediation information system engineering based on hybrid service composition mechanism, *J. Syst. Softw.* 108 (2015) 39–59.
- [11] L. Rao, G. Mansingh, K.-M. Osei-Bryson, Building ontology based knowledge maps to assist business process re-engineering, *Decis. Support Syst.* 52 (3) (2012) 577–589.
- [12] P.T. Kidd, *Agile Manufacturing: Forging New Frontiers*, Addison-Wesley Longman Publishing Co., Inc., 1995.
- [13] P. Lindberg, Strategic manufacturing management: a proactive approach, *Int. J. Oper. Prod. Manag.* 10 (2) (1990) 94–106.
- [14] H. Sharifi, F. Zhang, A methodology for achieving agility in manufacturing organisations: an introduction, *Int. J. Prod. Econ.* 62 (1–2) (1999) 7–22.
- [15] Y. Sheffi, Demand variability and supply chain flexibility, *Entwickl. Meile. Mod. Logist.* (2004) 85–117.
- [16] P. McCullen, R. Saw, M. Christopher, D. Towill, The F1 supply chain: adapting the car to the circuit—the supply chain to the market, *Supply Chain Forum.: Int. J.* 7 (2006) 14–23.
- [17] H. Schonenberg, R. Mans, N. Russell, N. Mulyar, W. van der Aalst, *Process Flexibility: A Survey of Contemporary Approaches*, *Advances in Enterprise Engineering I*, Springer, 2008, pp. 16–30.
- [18] A.-M. Barthe-Delanoë, S. Truptil, F. Bénaben, H. Pingaud, Event-driven agility of interoperability during the Run-time of collaborative processes, *Decis. Support Syst.* 59 (2014) 171–179.
- [19] E. Andonoff, W. Bouaziz, C. Hanachi, Protocol management systems as a middleware for inter-organizational workflow coordination, *IJCSA* 4 (2) (2007) 23–41.
- [20] M. Reichert, T. Bauer, and P. Dadam, Enterprise-wide and cross-enterprise workflow management: Challenges and research issues for adaptive workflows, 1999.
- [21] W. M. van der Aalst, T. Basten, H. M. W. Verbeek, P. A. Verkoulen, and M. Voorhoeve, Adaptive workflow: On the interplay between flexibility and support, in: *Proceedings of the first International Conference on Enterprise Information Systems*, Setúbal, Portugal, 1998, pp. 353–360.
- [22] E. Andonoff, W. Bouaziz, C. Hanachi, L. Bouzguenda, An agent-based model for autonomic coordination of inter-organizational business processes, *Informatica* 20 (3) (2009) 323–342.
- [23] M. Reichert, P. Dadam, ADEPT flex—supporting dynamic changes of workflows without losing control, *J. Intell. Inf. Syst.* 10 (2) (1998) 93–129.
- [24] P. Dadam et al., From ADEPT to AristaFlow BPM suite: a research vision has become reality, in: *Proceedings of the International Conference on Business Process Management*, 2009, pp. 529–531.
- [25] A. Lanz, *Adaptive Time- and Process-Aware Information Systems* (Ph.D. Thesis), Ulm University, 2017.
- [26] K. Képes, U. Breitenbücher, S. G. Sáez, J. Guth, F. Leymann, and M. Wieland, Situation-aware execution and dynamic adaptation of traditional workflow models, in: *Proceedings of the European Conference on Service-Oriented and Cloud Computing*, 2016, pp. 69–83.
- [27] M. Adams, A. H. Ter Hofstede, D. Edmond, and W. M. Van Der Aalst, Worklets: A service-oriented implementation of dynamic flexibility in workflows, in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, 2006, pp. 291–308.
- [28] W.M. Van Der Aalst, A.H. Ter Hofstede, YAWL: yet another workflow language, *Inf. Syst.* 30 (4) (2005) 245–275.
- [29] W.M. Van der Aalst, M. Weske, D. Grünbauer, Case handling: a new paradigm for business process support, *Data Knowl. Eng.* 53 (2) (2005) 129–162.
- [30] C. W. Guenther, M. Reichert, and W. M. van der Aalst, Supporting flexible processes with adaptive work? Ow and case handling, in: *Proceedings of the Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2008. WETICE'08. IEEE 17th, 2008, pp. 229–234.
- [31] H. M. Verbeek, A. Hirsenschall, and W. AALST VAN DER, XRL/Flower: Supporting inter-organizational workflows using XML/Petri-net technology, in *Web Services, E-Business, and the Semantic Web*, CAiSE 2002 International Workshop, WES 2002, 2002, pp. 93–109.
- [32] A.-W. Scheer, H. Kruppke, W. Jost, H. Kindermann, *Agility by ARIS Business Process Management: Yearbook Business Process Excellence 2006/2007*, 243 Springer Science & Business Media, 2006.
- [33] T. Catarci, et al., WORKPAD: process management and geo-collaboration help disaster response, *Int. J. Inf. Syst. Crisis Response Manag. (IJISCRAM)* 3 (1) (2011) 32–49.
- [34] R. Kovordanyi et al., Real-time Support for Exercise Managers' Situation Assessment and Decision Making, in *ISCRAM2012*, 2012.
- [35] C. Rooney, Crisis Project Website, Crisis Proj. (2011) (Online). (Available), <<http://idc.mdx.ac.uk/projects/crisis/>>.
- [36] A. Artikis, R. Marterer, J. Pottebaum, and G. Paliouras, Event processing for intelligent resource management, in: *Proceedings of the 20th European Conference on Artificial Intelligence*, 2012, pp. 943–948.
- [37] Y. Verginadis, I. Patiniotakis, N. Papageorgiou, and R. Stuehmer, Service adaptation recommender in the event marketplace: conceptual view, in: *The Semantic Web: ESWC 2011 Workshops*, 2011, pp. 194–201.
- [38] B. Cheng, M. Wang, S. Zhao, Z. Zhai, D. Zhu, J. Chen, Situation-aware Dynamic service coordination in an IoT environment, *IEEE/ACM Trans. Netw.* 25 (4) (. 2017) 2082–2095.
- [39] C. Zang, Y. Fan, R. Liu, Architecture, implementation and application of complex event processing in enterprise information systems based on RFID, *Inf. Syst. Front.* 10 (5) (. 2008) 543–553.
- [40] K. Chandy, W. Schulte, *Event Processing: Designing IT Systems for Agile Companies*, McGraw-Hill, Inc, 2009.
- [41] O. Etzion, P. Niblett, and D. C. Luckham, *Event processing in action*. Manning Greenwich, 2011.
- [42] D. Luckham and W. R. Schulte, Event Processing Glossary - Version 1.1, 2008 [Online]. Available: <http://complexevents.com/wp-content/uploads/2008/08/epts-glossary-v11.pdf>.
- [43] Y. Verginadis, D. Apostolou, A.-M. Barthe-Delanoë, and F. Benaben, Addressing agility in collaborative processes: A comparative study, in: *2013 Proceedings of the 7th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, 2013, pp. 120–125.
- [44] N.M. Josuttis, *SOA in Practice: The Art of Distributed System Design*, O'Reilly Media, Inc, 2007.
- [45] D. Luckham, SOA, EDA, BPM and CEP are all complementary. Part I. 2007.
- [46] J.-L. Maréchaux, Combining service-oriented architecture and event-driven architecture using an enterprise service bus, *IBM Dev. Works* (2006) 1269–1275.
- [47] B. M. Michelson, Event-Driven Architecture Overview: Event-Driven SOA Is Just Part of the EDA Story, Customers.com, 2006. [Online]. Available: <<http://www.customers.com/articles/event-driven-architecture-overview/>>. (Accessed 13-Jan-2018).
- [48] L. J. Füllöp, Á. Beszédes, G. Tóth, H. Demeter, L. Vidács, and L. Farkas, Predictive complex event processing: a conceptual framework for combining complex event processing and predictive analytics, in: *Proceedings of the Fifth Balkan Conference in Informatics*, 2012, pp. 26–31.
- [49] S. Li, L. Da Xu, S. Zhao, "The internet of things: a survey, *Inf. Syst. Front.* 17 (2) (2015) 243–259.
- [50] C. Perera, A. Zaslavsky, P. Christen, D. Georgakopoulos, Context aware computing for the internet of things: a survey, *IEEE Commun. Surv. Tutor.* 16 (1) (2014) 414–454.
- [51] OASIS, Web Services Base Notification 1.3 OASIS Standard, 2006. [Online]. Available: <http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf>. (accessed 13-Jan-2018).
- [52] G. M. Ramète, J. Lamothe, M. Lauras, and F. Benaben, A road crisis management metamodel for an information decision support system, in: *Proceedings of the 6th IEEE International Conference on Digital Ecosystems Technologies (DEST)*, 2012, pp. 1–5.
- [53] H. Pingaud, Rationalité du développement de l'interopérabilité dans les organisations, *Manag. Des. Technol. Organ.* (2009) 19–30.
- [54] OASIS, Web Services Brokered Notification 1.3 OASIS Standard, 2006. [Online]. Available: <http://docs.oasis-open.org/wsn/wsn-ws_brokered_notification-1.3-spec-os.pdf>. (Accessed 13-Jan-2018).
- [55] OASIS, Web Services Topics 1.3 OASIS Standard, 2006. [Online]. Available: <http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-os.pdf>. (Accessed 13-Jan-2018).
- [56] Esper, EsperTech, *EsperTech*. [Online]. Available: <<http://www.espertech.com/>>. (Accessed 13-Jan-2018).
- [57] TEPCO, Monitoring the situation at the Fukushima Daiichi nuclear power plant, Japan, 2011. [Online]. Available: <http://www.tepco.co.jp/cc/press/betu11_j/images/110528d.pdf>. (Accessed 16-May-2013).
- [58] E.D. Demaine, S. Mozes, B. Rossman, O. Weimann, An optimal decomposition algorithm for tree edit distance, *ACM Trans. Algorithms (TALG)* 6 (1) (2009) 2.
- [59] M. Pawlik, N. Augsten, RTED: a robust algorithm for the tree edit distance, *Proc. VLDB Endow.* 5 (4) (2011) 334–345.
- [60] < XmlUnit/ > . [Online]. Available: <<http://xmlunit.sourceforge.net/>>. (Accessed 13-Jan-2018).
- [61] Y.-H. Chen, T.-P. Lin, D.C. Yen, How to facilitate inter-organizational knowledge sharing: the impact of trust, *Inf. Manag.* 51 (5) (2014) 568–578.
- [62] M. L. Itria, A. Daidone, and A. Ceccarelli, A Complex Event Processing Approach for Crisis-Management Systems, arXiv:1404.7551 [cs], Apr. 2014.