Process Recommendation using Context in Crisis Management: Application to Flood Management

Hanane Ariouat, Eric Andonoff and Chihab Hanachi

IRIT, University of Toulouse 1, 2 rue du Doyen Gabriel Marty, 31042 Toulouse Cedex, France

Keywords: Crisis Management, Process Recommendation, Context, Flood.

Abstract: This paper addresses process recommendation in crisis management from relevant facts observed in the field and business knowledge of actors involved in crisis resolution. Facts observed correspond to damage or risk while business knowledge of crisis actors, i.e. actors involved in crisis resolution, corresponds to actions these actors can perform in the field to reduce the crisis and to strategies for using these actions according to the context. The approach recommended in the paper filters facts observed with strategies modelled taking into account the current situation and dynamically builds process models dealing with these facts. Built process models, represented as BPMN diagrams, define actions crisis actors have to perform in the field along with the coordination of actions. As several strategies are possible to deal with facts, several process models are recommended, each being labelled with its adequacy with the current situation. This paper presents the meta-model for facts and business knowledge modelling along with the recommended approach for process recommendation. Flood of the Loire serves as a case study for process recommendation illustration.

1 INTRODUCTION

In France, crisis management is under the responsibility of a command and control centre, called crisis cell. A crisis cell is headed either by a *prefect* or by the *interior minister*, depending on the crisis scale and it is composed of the representatives of different public organisations involved in its resolution. These participating actors are collectively responsible for the crisis resolution: they are responsible for actions undertaken in the field to mitigate risk or deal with damage and also for coordination of these actions, which has to be as efficient as possible. In a crisis cell, crisis resolution is modelled as a process, called Crisis Resolution Process –CRP– (Bénaben et al, 2015) (Andonoff et al., 2015): actions and actors in the field correspond to CRP activities and roles performing these activities, while coordination of actions is explicitly modelled in the CRP using coordination patterns such as sequence, alternative, or parallelism.

This paper addresses crisis resolution process modelling, which is an important issue for crisis cells. More precisely, the paper focuses on process recommendation from facts (risk or damage) observed in the field and considering the current situation. Each recommended process corresponds to a suitable response strategy for copying with a fact. The GéNéPi project serves as a support for process recommendation illustration. Indeed, in this project, we collaborate with crisis cell of county 45 in France with the aim to define a tool which recommends the most appropriate strategies to deal with facts taking into account the current situation. We mainly deal with flood crisis management, as county 45 is often impacted by floods of the Loire, which is one of the main French rivers.

Recommendation has already been investigated in business process management and even in crisis management. Some contributions (*e.g.*, (Schonenberg et al., 2008), (Negre, 2013), (Maamar et al., 2016)) addressed activity recommendation by suggesting the next activity to perform in a given situation. However these contributions did not deal with process recommendation, i.e. recommendation of coordinated activities, which is very useful for a crisis cell to have a comprehensive view of the resolution process. Other contributions (*e.g.*, (Macé-Ramette et al., 2013), (Ribeiro et al., 2014), (Ariouat et al., 2018)) addressed process recommendation. For instance, in (Macé-Ramette et al., 2013), the recommended solution deduces the crisis resolution

111

process to be deployed in the field according to the situation observed. The drawback of these solutions is that they indicate what has to be done and not what can be done. Yet interviews with crisis cell members in the context of GéNéPi have highlighted the need for knowing the possible options (strategies) to deal with the situation observed: crisis cell members want to assess these possible options and decide by themselves which one to perform in the field in accordance with their available resources for instance. As a consequence, existing contributions have to be revisited and improved to allow crisis cells for choosing among possible crisis resolution processes the most appropriate one according to the current situation.

This paper addresses process recommendation issue in crisis management field. Its contribution is threefold. First it introduces a meta-model supporting the modelling of both facts observed in the field and business knowledge required to deal with these facts. More precisely, business knowledge modelling includes (i) the modelling of strategies to deal with facts along with their use context, (ii) the modelling of services (i.e., actions in the field) offered by crisis actors and that the strategies need and, (iii) the modelling of use rules for these services. Note that processes implementing strategies are not directly modelled but rather built dynamically from relations existing between services that these strategies need. Second the paper presents the recommended approach for process recommendation. More precisely it presents the filtering step which matches facts and strategies comparing context of strategies with the current situation. The result of this filtering is, for each considered fact, the best strategies to deal with it, ordered by their similarity with the current situation. The paper also introduces the building process step which deduces BPMN processes from services needing for implementing chosen strategies. Third the paper reports on the experiment of our solution considering a real case study from GéNéPi.

Accordingly, the paper is organised as follows. Section 2 focuses on related work about process recommendation and compares our approach w.r.t existing contributions. Section 3 presents the recommended meta-model for facts and business knowledge modelling. Section 4 is dedicated to the recommendation of CRPs. First, it introduces our approach for filtering strategies using context and second it introduces the recommended solution for building processes corresponding to strategies. Section 5 illustrates facts and knowledge modelling and process recommendation, considering the case

study of GéNéPi, namely last important flood of the Loire in June 2016. Finally Section 6 concludes the paper and gives some directions for future work.

2 RELATED WORK

We have examined related work addressing recommendation in business process management. We distinguish those recommending activities from those recommending processes, i.e. set of coordinated activities.

Activity recommendation is particularly useful at run-time, i.e. when executing process. Some contributions addressed activity recommendation to suggest the next activity to perform in a given situation. For instance, the solution described in (Schonenberg et al., 2008) analyses log files to suggest to the user the next activity to be peformed in a given situation, which is featured by the already executed activities. Both contributions described in (Negre, 2013) and (Maamar et al., 2016) addressed activity recommendation in case of unforeseen situation. (Maamar et al., 2016) defined social relations between different components of a process (activities, machines, actors) which serve as a support for recommending corrective actions (activities) in response to an unforeseen situation such as unavailability of actors, machines or activities. (Negre, 2013) also recommended activities to deal with unforeseen situations affecting the stability of the real world. This contrbution uses knowledge from past experiences along with a similarity algorithm comparing the current situation with the situations of these past experiences for recommending corrective actions. On the other side, (Rangiha et al., 2016) described a recommender task system that uses social tagging to collect relevant information from discussions between process actors during process execution. Analysis of these tags allows the system for recommending new tasks when the same process must be executed again. Finally, (Deng et al., 2016) introduced a mechanism identifying patterns in process models and compares the process model being designed with the identified patterns to recommend activities that can be added to the process model being designed. These related work are interesting but they have the same drawbacks: (i) all of them only recommend an activity and thus they do not provide users with a comprehensive view of the set of activities (and their coordination) to perform to face the unforeseen situation and (ii) most of them –all except (Schonenberg et al., 2008), do not highlight the

possible activities in a given situation and they does not leave it to the user to decide which one he prefers to perform.

Regarding process recommendation, we can mention the following contribution: (Hornung et al., 2007), (Macé-Ramette et al., 2013), (Ribeiro et al., 2014) and (Ariouat et al., 2018). Note that process recommendation is rather useful at design-time, i.e. before process execution. Both (Macé-Ramette et al., 2013) and (Ariouat et al., 2018) have the same objective: the recommendation of a process model to deal with observed facts during a crisis. Both use business knowledge of crisis actors to deduce the crisis resolution process to be performed for reducing the crisis in the field. Unfortunately, none of them highlight the possible options to deal with observed facts and leave it to the user to decide which one he prefers to perform. In (Hornung et al., 2007), recommendation for designing processes is based on process reuse. Indeed, in this work, there is an ontology-based comparison between a process model being designed, expressed as a Petri net, and a set of already existing process models, also expressed as Petri net. The result of this comparison is the process model closest syntactically to the one being designed. Finally (Ribeiro et al., 2014) describes a recommender system that help users in choosing the best discovery algorithm for their data. This system uses as input a log file (data) and the different process discovery techniques. Measurements such as fitness and generalisation are used for the evaluation of the performance and the quality of these techniques. The system recommends process discovery techniques according to the best measures.

This work, led in the context of the GéNéPi project in collaboration with crisis cell of county 45 in France, focuses on process recommendation, i.e. recommendation of a set of coordinated activities, to deal with observed facts in the field. Indeed, crisis cells have two mains needs. First they need to have a comprehensive view of the set of activities to be performed (and their coordination) to cope with each observed fact (*e.g.*, to be able to evaluate the resource required to carry out all the activities and ask for help to other countys if necessary). Second, as they are responsible for the response in the field, crisis cells want to know the possible options (strategies) to deal with each fact observed, to assess these strategies and decide by themselves the ones that are the most suitable. Our solution meets these two needs as it recommends strategies and processes implementing them to cope with each facts observed. It orders these strategies (and processes)

comparing the context of the current situation and the context of possible strategies. Our solution differs from the existing ones cited above. The closet ones are (Negre, 2013) and (Ariouat et al., 2018): the first one recommends activities to be performed when unexpected situations occur during crisis while the other one describes a solution to deduce from facts observed in the field the crisis resolution process to be performed to cope with these facts. However, (Negre, 2013) does not meet the first need. Moreover, it advocates a context-based comparison but conditions featuring contexts are only equalities, which is really a drawback as illustrated in Section 5. As for (Ariouat et al., 2018), it does not meet the second need as it does not offer any options to crisis cells for their response in the field.

3 FACTS AND BUSINESS KNOWLEDGE MODELLING

This section introduces the recommended meta-model for facts and business knowledge modelling. This meta-model is given in Figure 1 as an UML class diagram.

Regarding facts modelling, the main concept is *Observed Risk/Damage*. This concept corresponds to an observed fact in the field, which can either be risk or damage. Damage is a negative situation affecting for instance population (*e.g.*, flooded house with people inside), building (*e.g.*, flooded school), road (*e.g.*, cut-off road)…, while a risk is the potential for damage. For each risk or damage, we store whether if it is risk or damage, and whether if it is known or unknown. When it is known, we refer to the knowledge base and more particularly we refer to *Intrinsic Risk/Damage* (relationship *correspond*), which corresponds to the known solution to the considered damage or risk. When it is unknown, the crisis cell has to specify how to deal with the new risk or damage, indicating which services to be deployed in the field. In addition, for each risk or damage, we store a specific property indicating if the risk or damage takes priority or not. A priority risk or damage has to be considered in the recommendation process, while a non-priority risk or damage will be taken into account later, when another recommendation is made. Crisis cell members may change the value of this property according to the urgency of risk or damage.
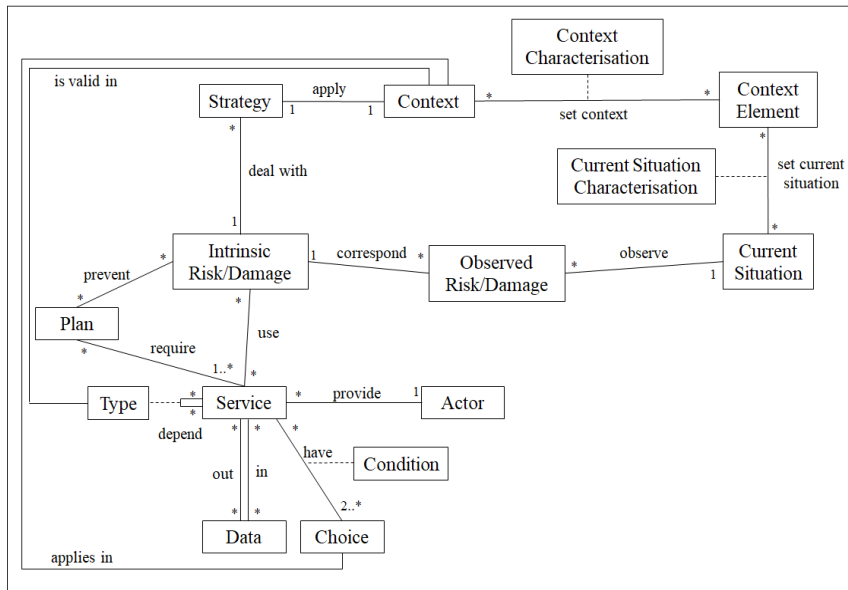
Figure 1: Facts and Business Knowledge meta-model for process recommendation.

Finally, risks and damages are observed in a specific situation, namely the *Current Situation*. We characterise a current situation by a set of conditions involving context elements. These conditions are defined as a triplet (context element, operator, value).

Regarding business knowledge modelling, we distinguish services offered by crisis actors from knowledge required to deal with observed facts. Services offered by crisis actors are modelled using the following concepts: *Service*, *Actor*, *Data*, *Choice*, *Condition* and *Type*. A service is an operational action that can be executed in the field by an actor. For each service, we store data consumed and produced (relationships *in* and *out*). Moreover, we specify use rules of these services. These use rules are expressed as relations between services (relationship *depend*), and whose type may be require, cause, or follow. Types *require* and *cause* define a strong relation among considered services, indicating that both services have to be executed one after the other: require indicates there is a precedence relation among them while cause indicates that there is a succession relation among them. Opposite, type *follow* defines a weak relation among considered services, indicating that a service will obviously be performed after another, but not necessarily right after. In addition, we also have introduced another use rule for services, namely the *choice* use rule. The idea is to support alternative modelling, each alternative being a solution to deal with an issue. A condition defines when using this alternative.

Knowledge required to deal with observed facts is modelled using the following concepts: *Intrinsic Risk/Damage*, *Plan*, *Strategy*, *Context*, *Context Element* and *Context Characterisation*. The notion of Intrinsic Risk/Damage is central. First it defines how to deal with an already observed risk specifying the required services, possibly as part of a plan, which corresponds to an already specified set of actions to be undertaken to address a particular issue. Second it defines the different strategies to deal with observed risk or damage along with the context in which to use these strategies. A *Context* is featured by a set of conditions involving context elements. As for current situation characterisation, these context conditions are defined as triplet (context element, operator, value). Note that relations between services depend on the context (relationships *applies in* and *is valid in*).

Section 5 illustrates the modelling of facts and business knowledge as instance of this meta-model considering the Loire case study.

4 PROCESS RECOMMENDATION

Our approach for process recommendation is a two-step approach. In a first step, for each fact observed, we filter the possible strategies to deal with the considered fact and order them according to context. In a second step, after user has chosen, for each fact observed, the strategy he prefers to implement in the

field, we build the process corresponding to each chosen strategy. The following sections detail these two steps.

4.1 Filtering Strategies using Context

We discuss below filtering strategies using context first introducing the recommended approach for filtering and second detailing the approach giving some of the algorithms implementing it.

4.1.1 Recommended Approach

The objective of filtering strategies is the recommendation of a set of strategies suitable for each fact observed in the field. Our approach for this filtering is given in Figure 2 as a BPMN diagram.

The process is composed of a single activity, *Strategy Filtering*, modelled as a sub-process in the BPMN diagram, and repeated for each fact observed (hence the cycle in the sub-process). In addition, for each observed fact, we process the five following steps. In a first step, we identify the current situation in terms of context elements and values for these context elements: classes *Current Situation*, *Current Situation Characterisation* and *Context Element* of the meta-model are required for this identification. The second step is dedicated to the intrinsic fact identification, i.e. the identification of the intrinsic risk or damage corresponding to the risk or damage observed: class *Intrinsic Risk/Damage* of the meta-model is required for this identification. The third step deduces the possible strategies and their context for dealing with the considered intrinsic fact: classes *Strategy*, *Context*, *Context Element* and *Context Characterisation* are required for this deduction. In the fourth step, for each strategy (hence the cycle in the activity *Similarity Calculation*), we calculate the similarity between the current situation context and the context of the considered strategy, and finally, in the fifth step, we recommend/order the strategies according to the similarity.

4.1.2 Detailing the Approach

We mainly detail below the algorithm implementing similarity calculation step. Indeed, as we have implemented the meta-model in a database management system (namely MySQL), the other steps are implemented as queries. Thus even if some of them were hard to write due the complexity of the meta-model, the main challenge for strategy filtering is the context-based similarity calculation.

As explained before, in this calculation, there is a comparison between the context of the current situation and the use context of a strategy, each involving context elements. More precisely, the context of the current situation is a set of conditions involving context elements and their corresponding values, measured in the field: for instance, *water level = 1.80*, where water level is a context element and 1.80 is the measured water level, in meters. On the other hand, the context of a strategy is a set of conditions involving context elements. These conditions define the use conditions of the strategy. For instance, a strategy may involve the context element water level and may be used when the condition *water level < 2.50* is checked. The algorithm implementing similarity calculation has these two sets of conditions as input and it returns a similarity value corresponding to the number of conditions of the strategy that are checked according to the values of the current situation divided by the total number of conditions of the strategy. This algorithm uses the following set of functions supporting the handling of set of both context elements and conditions:

- determineContextElements(s) returns the set of context elements involved in the set of conditions *s*,
- checkCondition(c,s) returns *true* if the condition *c* is checked in the set of conditions *s*, otherwise *false*,
- cardinality(s) returns the number of elements in the set *s*.

The algorithm implementing similarity calculation is the following.

```
SimilarityCalculation(csc,sc:
                  Set(Condition)):real
Local similarity: real, c: Condition,
  CEinCSC, CEinSC: Set(ContextElement)

Begin
  CEinCSC = determineContextElement(csc)
  CEinCS = determineContextElement(cs)
  similarity = 0
  If CEinCSC ⊆ CEinSC Then
    For Each c in csc Loop
      If checkSimilarity(c,sc) Then
        Similarity = similarity + 1
      End If
    End Loop
    similarity = similarity /
              cardinality(sc)
  End If
  Return similarity
End
```
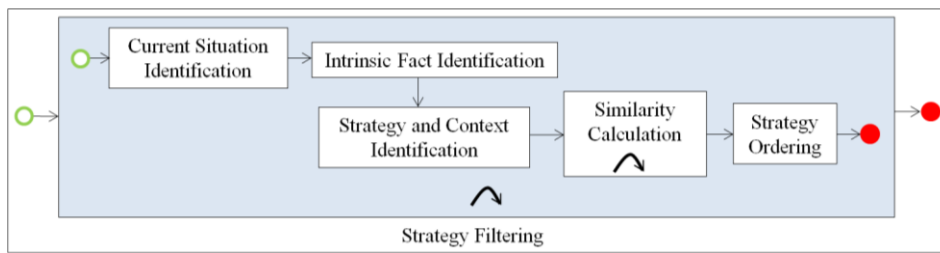
Figure 2: Filtering Strategies Approach.

Note that this algorithm returns 0 when the set of context elements featuring the current situation is not included in the set of context elements featuring the considered strategy. That means that, for a strategy to be considered suitable, each context element must exist in both the current situation and the considered strategy

4.2 Dynamic Process Building

We discuss below dynamic process building first introducing the recommended approach for process building and second detailing the approach giving some of the algorithms implementing it.

4.2.1 Recommended Approach

Our approach for building processes of chosen strategies is given in Figure 3 as a BPMN diagram. This BPMN diagram defines three main steps.

The first step is the *Service Identification* step, which selects services required to implement the considered strategy in the field. The resulting set of services is then expanded in the *Service Expansion* step. To do this, we exploit use rules between services, and more particularly require, cause and choice use rules, to identify additional services to be deployed. The result of this expansion step is the set of services to be ordered in the corresponding crisis resolution process. Finally, the *Service Ordering* step is responsible for ordering services w.r.t. their relation. It is visualised as a sub-process in Figure 3. First, we build a matrix describing dependences existing between considered services from relations existing between them. As in (Aalst, 2016), we consider three types of dependences:

- causal dependence: a causal dependence between services a and b, denoted a $\rightarrow$ b, indicates that service a has to be executed just before service b,
- parallelism dependence: a parallelism dependence between services a and b, denoted a $\|$ b, indicates that services a and b are executed in any order,

- unrelated dependence: an unrelated dependence between services a and b, denoted a # b, indicates that services a and b are completely independent one from another, that is it does not exist any causal or parallelism dependence between them.

Note that, opposite to process mining algorithms (Aalst, 2016), we do not exploit log files but only business knowledge from crisis actors: services offered by these actors and use rules between these services.

Then, from this dependence matrix, we build the corresponding Petri Net from which we derive the corresponding BPMN diagram. The Petri net serves as a support for crisis resolution process simulation, validation and analysis (this can be very useful for crisis cells), while the BPMN serves as a support for crisis resolution process execution.

Note that the Petri net formalism has been chosen as it provides formal and executable specifications to analyse, simulate, check and validate the process built (Aalst, 1998) while BPMN has been chosen as it is the language of the process engine that we use in GéNéPi. This process engine, namely Iterop, is built on top of Activiti. It is provided by our GéNéPi partner InteropSys.

4.2.2 Detailing the Approach

Different algorithms have been written to implement process building. Due to space limitation, we give below the main ones. We first introduce the *ServiceExpansion* algorithm. This algorithm uses the following set of functions supporting the handling of relations between services:

- require(s) returns set of services required by the service *s*,
- cause(s) returns set of services caused by the service *s*,
- choice(s) returns set of services in choice with *s*.
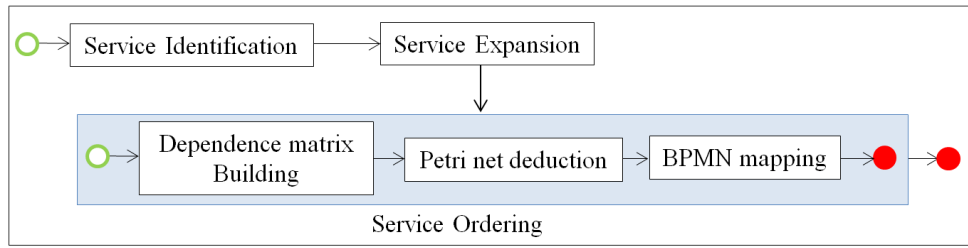
Figure 3: Process Building Approach.

This algorithm is the following.

```
ServiceExpansion(s:Set(Service)):
                    Set(Service)
Local x: Service, Expanded,
            tobeExpanded: Set(Service)
Begin
  tobeExpanded = s
  Expanded = ∅
  While tobeExpanded <> ∅ Loop
    x = Select(tobeExpanded, Expanded)
    tobeExpanded = tobeExpanded - {x} +
        require(x) + cause(x) + choice(x)
    Expanded = Expanded + {x}
  End Loop
  Return Expanded
End
```

The idea is to add services which are required to, consequence of, or alternative to each service obtained after the initial service identification. For that, we use two sets of services, namely *tobeExpanded*, whose initial value is the set of services obtained after service identification, and *Expended*, which is the resulting set and whose initial value is empty. The algorithm adds to *Expended* both a service x from *tobeExpanded* and services connected to x by require, cause or choice relations. Note that we take into account the relationships *is valid in* and *applies in* of the meta model (cf. Figure 1) to only consider require, cause and choices relations that holds for the context of the chosen strategy.

Regarding dependence matrix building, we do not give the underlying algorithm that implements this step. However, we give some hints to understand what the algorithm is doing. To get into details, from the set of services obtained after service expansion, the algorithm produces causal dependences according to require, cause and follow relations. It also analyses use conditions of services to eventually define new services which correspond to choices and produces unrelated dependences according to choice relations. Finally, parallelism dependences are deduced using following rules:

(1) If a → b and a → c and not (b # c) Then b ∥ c

(2) If a ∥ b and a → c then b ∥ c

We also introduce the algorithm *PetriNetCalculation*, which returns deduced crisis resolution process as a Petri net diagram. Petri net formalism supports process description in terms of places, transitions, corresponding to actions to be executed, and arcs, connecting places and transitions.

This algorithm differs from the process-mining algorithm Alpha (Aalst, 2004). While Alpha analyses log files to identify direct succession dependences between activities, from which it builds the matrix, we derive them from the meta-model. Second, our construction of the Petri net from the matrix is fairly similar to the Alpha's one, but we add specific places and transitions to build processes possibly starting with parallelism or alternative. More precisely, as Alpha, we identify initial and final services, which are services to be executed respectively at the beginning and at the end of the crisis resolution process. Then, the novelty is to define two virtual transitions: Start and End. Start is connected to each initial service so that they could be performed after Start. Also, each final service is connected to the End transition, so that the End transition merges the results of the final services. Another important difference with Alpha is that we are able to deduce alternatives involving empty activities. Thus we overcome some limitations of Alpha (e.g., (Wen et al., 2007)).

Finally, the part of the algorithm inspired by Alpha is (i) the determination of X, the minimum set of couples (Servicesa, Servicesb) for which, each sa in Servicesa has a causal dependence with each sb in Servicesb as well as sa and sb are unrelated and (ii) the aggregation of the final Petri net (T,P,A).

This algorithm uses the following functions for the handling of dependences between services:

- determineCausalDep(m) returns the set of causal dependences in the matrix *m*,
- determineParallelDep(m) returns the set of parallel dependences in the matrix *m*,
- determineUnrelatedDep(m) returns the set of

unrelated dependences in the matrix m,

- determineServices(m) returns the set of services in the matrix m,
- determineLeftSideService(m) returns the set of services in the matrix m which are not left-hand side of any dependence,
- determineRightSideService(m) returns the set of services in the matrix m which are not right-hand side of any dependence,

This algorithm is the following.

```
PetriNetCalculation(m:Matrix): PetriNet
Local ca, pd, ud: Set(Dependence),
     T: Set(Transition), P: Set(Place),
     A: Set(Arc), PN: PetriNet
Begin
 ca=determineCausalDep(m)/*perform →
 pd=determineParallelDep(m)/*perform //
 ud=determineChoiceDep(m)/*perform #
 Ts = {Start}
 Te = {End}
 T = DetermineServices(m) + Ts + Te
 Ss = determineRightSideService(m)
 Se = determineLeftSideService(m)

 X = {(A,B) / A⊆T ∧ B⊆T  ∧ ∀a∈A ∀b∈B,
a→b ∧ ∀a1,a2∈A, a1#a2 ∧ ∀b1,b2∈B, b1#b2}

 Y = {(A,B)∈X / ∀(A',B')∈X A⊆A' ∧ B⊆B'
⇒ (A,B)=(A',B')} + {(Ts,W) / W⊆T ∧
(∃Z⊆T/(W,Z)∈X) ∧ (∀Z'⊆T (Z',W)∉X) ∧
(∃w∈W/w∈Ss)} + {(W,Te) / W⊆T ∧
(∃Z⊆T/(Z,W)∈X) ∧ (∀Z'⊆T (W,Z')∉X) ∧
(∃w∈W/w∈Se)}

 P = {P(A,B) / (A,B)∈Y} + {P(∅,Ts),P(Te, ∅)}
 A = {(a,P(A,B))/(A,B)∈Y ∧ a∈A} +
{(P(A,B),b)/(A,B)∈Y ∧ b∈B} +
{P(∅,Ts),Start} + {End,P(Te, ∅)}
 PN = (T,P,A)
 Return PN
End
```

The resulting Petri net is then mapped with the *Mapping* algorithm into a BPMN diagram. We do not detail this algorithm in the paper as this mapping is after all quite classic (*e.g.*, a specific plug-in in PROM supports mapping to BPMN from Petri net), but we explain the specificities of GéNéPi mapping. Indeed, in GéNéPi, BPMN is not only a notation for crisis resolution process visualisation but also the executable process language of Iterop, the process engine that supports crisis resolution process execution. Thus to obtain a fully executable specification, we have mapped flowing conditions, i.e. conditions attached to sequence flow flowing from open exclusive gateways to activities (*i.e.*, services), in the BPMN diagram. More precisely, if

use conditions of services are defined in the meta-model then these use conditions are the flowing conditions. Otherwise, the algorithm automatically adds an out data to the activity preceding an open exclusive gateway, and defines for each sequence flow flowing from this open exclusive gateway a condition in which this out data is involved. Another interesting aspect in this mapping is the labelling of services with the facts they deal with. Thereby the algorithm labels each service with the facts justifying the selection of the service in the crisis resolution process, making it possible to determine whether or not all activities related to a fact are carried out or not. Thus it is possible to modify crisis situation deleting facts processed from the list of facts to be taken into account. Finally, we simplify the crisis resolution process in removing Start and End services, which were introduced for consistency reasons when building the Petri net, but which are no more useful in the BPMN. We also remove added services in the Petri net for syntactic reasons but useless in the BPMN.

5 CASE STUDY

Orléans, main city and prefecture of county 45 in France is often deeply affected by Loire's floods and the mastering of these floods is of utmost importance for the city. Thus we have conducted an experiment in collaboration with the crisis cell of Orléans, considering the last important flood. Members of the crisis cell were the Prefect, head of county 45 prefecture, the COD, which is the operational committee set up within the crisis cell and finally the representatives of the different actors acting in the field (*e.g.*, DDT that are responsible for dykes supervision, CPZCR that are responsible for motorways supervision ARS that are responsible for health-related matters…).

The experiment has focused on the simulation of several days of the last important flood of the Loire in June 2016. We report below part of the experiment which illustrates both modelling of business knowledge and facts and the recommendation in terms of retrieved strategies and corresponding built processes. For this illustration, we mainly focus on two specific facts observed during the flood, which are:

- risk of dyke failure in Saint Pryvé Saint Mesmin: municipality of Saint Pryvé Saint Mesmin, next to Orléans, could be flooded and some districts of the municipality could be evacuated,

Table 1: Strategies for Risk of Dyke Failure.

Strategy	water level	impacted area	probability	evacuationType
RDF.1	<2.0	="urbanised"	>=0.5	
RDF.2	>=2.0 and <3	="urbanised"	>=0.7	
RDF.3	>=3	="urbanised"	>=0.7	<=0.5
RDF.4	>=3	="urbanised"	>=0.7	>=0.5

▪ risk of flooding impacting both nursing home Saint Pryvé Lake (the nursing home has possibly to be evacuated) and motorway A71 (the motorway has to be partly cut off).

5.1 Copying with Risk of Dyke Failure

The risk of dyke failure has been observed during several days (the level of the Loire has risen regularly from day 3 to day 8 of the crisis), notably for the dyke in Saint Pryvé Saint Mesmin.

To cope with this risk, we have modelled, in collaboration with crisis cells members, the possible response strategies according to the context. Context elements needed for this modelling are: *water level*, *impacted area*, which features the size of the population potentially impacted (Saint Mesmin Saint Privé is an urbanised area), *probability*, which corresponds to the potential for dyke failure (and thus flooding), and *evacuationType*, which indicates the effort require for the evacuation. For instance on day 7, the following context elements and values feature the current situation: *water level = 3.20* and *impacted area = "urbanised"* and *probability = 0.7*. On the other hand no value is specified for the context element evacuationType. That means that at the time of the situation assessment, the crisis cell has not been able to value this element.

The different modelled strategies for the intrinsic risk dyke failure, which corresponds to the observed risk disk failure in Saint Privé Saint Mesmin are summarised in Table 1, along with their use context. The different conditions defined in use context of strategies are connected to each other by the logical operator *and*.

Note that the context element *evacuationType* differentiates RDF.3 from RDF.4. When the value of this context element is lower than 0.5 then RDF.3 must be chosen as the effort required for the evacuation is not very important. On the other hand, when the value of this context element is greater than 0.5 then RDF.4 must be chosen. Finally when the value of this context element is equal to 0.5 then both strategies can be chosen. In this case, it is up to the crisis cell to decide which strategy to apply.

In light of current situation on day 7 of the crisis, for which we have *water level = 3.20* and *impacted*

area = "urbanised" and *probability = 0.*7, the execution of the filtering step orders the strategies according to their similarity with the current situation as illustrated in Table 2.

Table 2: Similarity Calculation Results.

Strategy	Similarity
RDF.3	1
RDF.4	1
RDF.2	0.66
RDF.1	0.33

Similarity for *RDF.3* and *RDF.4* is equal to 1 as all their conditions are checked: the values observed for the context elements in the current situation matches with the conditions of both strategies. In contrast similarity for RDF.2 is equal to 0.66 (the condition related to the context element water level is not verified) and similarity for RDF.1 is equal to 0.33 (conditions related to water level and to probability are not verified).

Then members of the crisis cell have to choose among the recommended strategies, which one they prefer to use. Once selected, the second step of the recommendation process, i.e. the dynamic building of the process implementing the chosen strategy is performed. Let us suppose that *RDF.3* is chosen by crisis members. Knowledge required for building the process corresponding to this strategy is stored in the meta-model. Table 3 lists services stored while Table 4 shows relations between them.

Table 3: Services required for RDF.3.

ID	Name	Actor
0	Prepare for dyke supervision	COD
1	Dyke supervision	DDT
2	Report on dyke supervision	DDT
3	Decision-making for evacuation	COD
4	Issue evacuation order	Prefect
5	Inform population	Prefect
6	Door knocking	Mayor
7	Evacuation supervision	COD
8	Encouraging evacuation	Gendarmerie

Note that services whose Id is 100, 101 and 102 are automatically added to the list of services (there are not shown in Table 3 since they were not initially modelled) as the algorithm identifies choices (cf. Section 4.2.2). For each of them, a condition involving an out data from activity

preceding choice is added. Data and added conditions are shown in Figure 5. In addition we store in the meta-model the intrinsic risk *dyke failure*. This intrinsic risk is linked to RDF.3 (relationship *deal with* in the meta-model) and it is linked to the service whose Id is 1 (relationship *use* in the meta-model).

Table 4: Relations between Services Required.

ID1	relationType	ID2
1	require	0
1	cause	2
2	cause	3
2	cause	100
3	choice	100
3	cause	4
3	cause	101
4	choice	101
4	cause	5
5	cause	6
6	cause	7
7	cause	8
7	cause	102
8	choice	102

Then we build the corresponding dependence matrix from which first the Petri net and second the corresponding BPMN diagram are deduced. The matrix built for RDF.3 is given in Table 5 while the deduced BPMN diagram is given in Figure 4. In this BPMN diagram, out data from activity are indicated in BPMN comments (*e.g.*, out data for activity *Evacuation supervision* is *evacuation speed*) and these out data can be involved in sequence flow conditions after exclusive gateways (*e.g.*, *evacuation speed = "slow"*). In addition, some data are modelled as BPMN data objects. For instance, *supervision report* is an out data object for activity *Report on dyke supervision* and an in data for activity *Decision-making for evacuation*.

Table 5: Dependence Matrix for RDF.3 process.

	0	1	2	3	4	5	6	7	8	100	101	102
0	■	→										
1		■	→							→		
2			■	→						→		
3				■	→					#	→	
4					■	→					#	
5						■	→					
6							■	→				
7								■	→			→
8									■			#
100										■		
101											■	
102												■

Note that the built process includes activities implementing crisis cell decision making.

5.2 Copying with Risk of Flooding

A very high risk of flooding was observed from day 7 to day 8 of the crisis, involving several components close to Orleans. In the following, we report on the risk of flooding on the nursing home Saint Privé Lake and on the motorway A71.

Both observed risks correspond to the intrinsic risk *flooding*, for which we have modelled different response strategies according to the type of impacted component. For each of these strategies, we have also modelled the services required for their implementation along with existing relations between these services. Due to space limitation, we do not detail the modelling of this knowledge (as we did in section 5.1 for the risk of dyke failure) but we give in Table 6 the modelled strategies and their use context. Table 6 indicates that we have defined two strategies to cope with the risk of nursing home flooding and four strategies to deal with the risk of road flooding.

Table 6: Strategies for Risk of Flooding.

Strategy	probability	component-Type
RF.1	>=0.4 and <0.7	="nursing home"
RF.2	>=0.7	="nursing home"
RF.3	>=0.7	="motorway"
RF.4	>=0.7	="main road"
RF.5	>=0.7	="county road"
RF.6	<0.7	="road"

The two first strategies define what to do for risk of flooding on nursing home. The first one addresses the preparation of the nursing home evacuation while the second one corresponds to its effective evacuation. The context element probability enables the choice between the two. The four last strategies define what to do for risk of flooding on roads according to the size of the road (see component-Type value): the three first ones define how to cut-of the road (there is only one strategy per road size) while the fourth one define how to alert motorists to the risk of flooding.

On day 7 of the crisis, two main context elements feature the current situation and have the following values:

- regarding risk of flooding on Saint Privé Lake, we have: *probability = 0.8* and *component-Type="nursing home"*,
- regarding risk of flooding on A71, we have: *probability=0.8* and *component-Type="motorway"*.
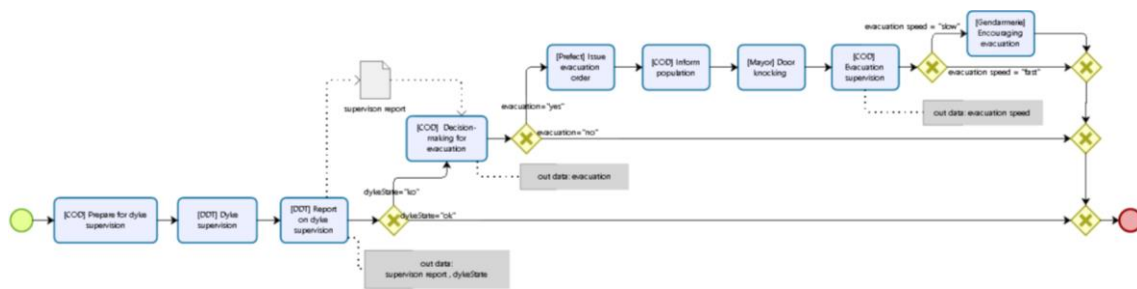
Figure 4: Process Recommendation for Risk of Dyke Failure.

From these field data, we proceed to the filtering step first identifying the corresponding intrinsic risks (flooding for both observed risks in the field), second identifying their corresponding strategies along with their context and third calculating, for each of these strategies, the similarity between strategy context and current situation context. The final result is strategy RF.2 to cope with risk of flooding on Saint Privé Lake, and strategy RF.3 to cope with risk of flooding on A71.

In the building process step, we dynamically build the process implementing selected strategies RF.2 and RF.3 in the same BPMN diagram. The BPMN diagram obtained is given in Figure 5. Note that the built process includes activities implementing hierarchical communication towards the interior ministry to which crisis cell is accountable. In addition this process is complex as it models parallel activities implementing response to each observed risk. Moreover, these activities are labelled with the observed risk they deal with (*e.g.*, risk of flooding in nursing home Saint Privé lake labels activities of the upper branch of the BPMN diagram.

6 CONCLUSION

This paper has addressed process recommendation in crisis management field. The Process recommendation solution advocated in this paper uses data observed in the field, i.e. risk and damage of the crisis, along with business knowledge of actors involved in crisis resolution in order to (i) filter and recommend the different strategies copying with observed facts according to the context and (ii) dynamically build processes corresponding to chosen strategies. Recommendation is a key step in GéNéPi, and more generally in process-driven crisis management, as it provides crisis cells with guidelines for crisis reduction. These guidelines are consistent with facts observed, context in which

these facts are observed and crisis actors' knowledge. Moreover, crisis cells, to which the process recommendation solution is intended for, and which is responsible for the response in the field, can know the possible options (strategies) to deal with each observed fact, to assess these strategies and decide on its own the ones that are the most suitable.

The recommended solution includes (i) a meta-model supporting facts and knowledge modelling and (ii) a set of algorithms implementing crisis resolution process recommendation. Our knowledge-based solution extends existing contributions, and notably (Negre, 2013), (Macé-Ramette et al., 2013), (Ariouat et al., 2018), which are the most interesting solutions in crisis management field. The two last contributions deduce the process that must be performed in the field. These contributions do not left any choice to crisis cells as these latter do not know the possible options to cope with facts observed. This is a major drawback of these two contributions. Our solution also extends the one described in (Negre, 2013) for the following reasons. First (Negre, 2013) recommends only activities, which is a drawback for crisis cells that need to know the full resolution process to be performed in the field in order to assess the resources needed for its implementation. Our solution recommends processes and thus overcomes this first drawback. Second (Negre, 2013) advocates a context-based similarity calculation in the filtering step of process recommendation, as we do in our solution. However, in (Negre, 2013), there are the following limitations: conditions defining contexts involve only equal operator, the algorithm supporting the filtering is not given, and no convincing examples are provided. In our solution, both we consider conditions involving any comparison operators and we give the algorithm implementing the filtering of strategies. In addition, we have tested our solution with a real case study, in the context of the GéNéPi project. Crisis cell of county 45 helped us in defining knowledge of business actors, involved in the field for crisis
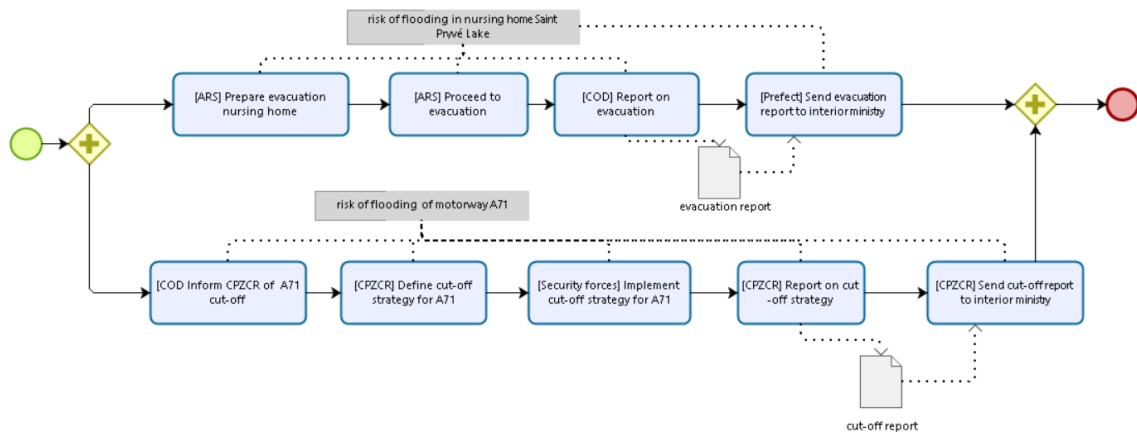
Figure 5: Process Recommendation for Risk of Flooding.

resolution, and provided us with real field facts, from the last important flood of the Loire in June 2016.

We really believe our contribution is a step forward to address process recommendation in the crisis management field. However, we have identified two main improvements. The first one is related to consistency of modelled knowledge, and more precisely consistency of relation between services. We did not investigate this point and have planned to do it shortly. The second one is related to the integration of social dimension into crisis resolution processes for improving recommendation. We will investigate this point the next future.

REFERENCES

van der Aalst W., 2016. *Process Mining – Data Science in Action*. Springer.

van der Aalst W., 1998. The application of Petri Nets to Workflow Management. *Circuits, Systems and Computers*, 8(1), pp. 21–66.

van der Aalst W., Weijters T., Maruster L., 2004. Workflow Mining: Discovering process models from event logs. *Transactions on Knowledge and Data Engineering*, 16(9), pp. 1128–1142.

Andonoff E., Hanachi C., Le Nguyen Tuan T., Sibertin-Blanc C., 2015. Interaction Protocols for Human-Driven Crisis Resolution Processes. *IFIP Working Conference on Virtual Enterprises*, Albi, France, pp. 63–76.

Ariouat H., Andonoff E., Hanachi C., 2018. From Declarative Business Knowledge to Process Driving Crisis Resolution. Toulouse 1 - Capitole University technical report.

Bénaben F., Truptil S., Lauras M., Salatgé N., 2015. Management of Collaborative Behavior through a Service-Oriented Mediation System: the case of Crisis

Management. *Int. Conference on Service Computing*, New-York, NY, USA, pp. 554–561.

Deng S., Wang D., Li Y., Cao B., Wu Z. Zhou M., 2016. A Recommendation System to Facilitate Business Process Modelling. *IEEE Transactions on Cybernetics*, 47(6), pp. 1380–1394.

Hornung T., Koschmider A,. Oberweis A., 2007. A Recommender System for Business Process Models. *Int. Workshop on Information Technology and Systems*, Montreal, Canada, pp. 127–132.

Maamar Z., Facib N., Sakrc F., Boukhebouzed M., Barnawie A., 2016. Network-based Social Coordination of Business Processes. *Information Systems*, 58, pp. 58–74.

Macé-Ramète G., Lauras M., Steffan L., Lamothe J., Bénaben F., Barthe-Delanoë AM., Dolidon H., Lilas L., 2013. Towards a Predictive Model for Decision Support in Road Crisis Management. *Int. Conference on Digital Ecosystems and Technologies*, Menlo Park, CA, USA, pp. 137–140.

Negre E., 2013. Towards a Knowledge (experience)-based Recommender System for Crisis Management. *Int. Conference on P2P, Parallel, Grid Cloud, and Internet Computing*, Compiègne, France, pp. 713–718.

Rangiha M., Comuzzi M., Karakostas B., 2016. A Framework to Capture and Reuse Process Knowledge in Business Process Design and Execution using Social Tagging. *Business Process Management*, 22(4), pp. 835–859.

Ribeiro J., Carmona J., Misir M., Sebag M., 2014. A Recommender System for Process Discovery. Int. Conference on Business Process Management, Eindhoven, The Nederlands, pp. 67–83.

Schonenberg H., Weber B., van Dongen B., van der Aalst W., 2008. Supporting Flexible, Process through Recommendations based on History. *Int. Conference on Business Process Management*, Milan, Italy, pp. 51–66.

Wen L., van der Aalst W., Wang J., Sun J., 2007. Mining process models with non-free-choice constructs. *Data Mining and Knowledge Discovery,* 15(2), pp. 145–180.