

# Making the Right Decisions for Disaster Prevention

Juan Carlos Augusto Hui Wang Jun Liu

University of Ulster at Jordanstown, Northern Ireland, UK.

## Abstract

We present a framework for decision-making in relation to disaster management. The use of causality reasoning based on the temporal evolution of a scenario provides a natural way to chain meaningful events and possible states of the system. There is usually several ways to analyse a problem and several possible strategies to follow as a solution and it is also often the case that information originating in different sources can be inconsistent. Therefore we allow the specification of possible conflicting situations as they are a typical element in disaster management. A decision procedure to decide on those conflicting situations is explained which not only provides a framework for the assistance of one decision-maker but also how to handle opinions from a hierarchy of decision-makers.

**Keywords:** Disaster prevention, Causal Reasoning, Decision Theory, Artificial Intelligence.

## 1. Introduction

Disaster management, which encompasses monitoring, predicting, preventing, preparing for, responding to, mitigating and recovering disasters, aims to deal with any potential and actual disaster by effective and efficient organization, communication, interaction and utilization of counter-disaster resources. Usually decisions are made by humans but more and more computer-based decision-making support has been accepted and developed. Although human decision-makers (DMs) are usually better than machines to judge complex situations and make thoughtful decisions, computers can provide a stress-free view of the situation, efficiently compile important amounts of knowledge and make objective, calculated decisions.

In this paper we propose a unique AI-based hybrid approach to making decisions in the specific domain of disaster prevention. We use a temporal logic language to represent causal relations of the world, which also assimilates dynamic and heterogeneous information. Decision-making support is performed by evaluating alternative explanations that can be used as predictions of potential future states of a world. A theory on preferences and associated algorithms are developed

for arbitration of different explanations and of different advisers, which gives rise to the optimal decision for the situation.

Here we describe a theoretical framework to capture some key aspects of a decision support system. We focus in this paper on the logical core of it, which uses a simple representation of causal relationships (see Section 2) to build explanations for possible diagnosis of situations (see Section 3). These notions are illustrated by way of a scenario (Section 4) which is later used to explain how decision making can be supported by our system. Different preference criteria can be encoded to rank order explanations based on particular evidence. In the classical preference structures, the decision-maker is supposed to be able to totally compare the explanations. But certain situations, such as lack of information, uncertainty, ambiguity, and conflicting preferences, can lead to partial orders between explanations. This problem is considered in Section 5 where we define a framework through which different preferences can be merged to make a preference criteria and select in between alternative explanations. This process is generalized in Section 6 for the case where a DM have to take decisions based on the advise of other subordinates.

## 2. Causal representation of the system

An essential part of the decision making process deals with connecting current states of the system with potential scenarios for the DM to assess the situation and make appropriate judgments. We represent basic key features of the system and their inter-relation from a causal perspective.

States are partitioned into two classes: *dependent* ( $\mathcal{S}_D$ ) and *independent* ( $\mathcal{S}_I$ ).  $\bar{S}$  is the opposite of  $S$ . An independent state does not depend causally on other states holding at the same time, whereas a dependent state can do so. An independent state can only be initiated by the occurrence of initiating or terminating *events*. A state  $S$  will be *co-independent* if  $\bar{S}$  is independent. Events will represent external influences to the system being modelled and will drive its internal change.

A specification of a system will be interpreted over a sequence of states, starting with the initial state  $S_0$ :  $S_0, S_1, S_2, \dots$  indexed by time. When the clock ticks at time  $t$  the system leaves  $S_t$  and the system is at state  $S_{t+1}$ . This new state is computed from  $S_t$  by first applying any event  $E_i$  such that  $occurs(E_i, t : t + 1)$  and then applying the causal rules. An example of this will be given later when we introduce a practical scenario.

The following technical language, based on that of [1], is used to represent our scenarios. We restrict ourselves to a propositional language where cause-effect relationships can be expressed directly in a simple way.

**Definition (syntax)** ([1] page 3): There are two kinds of *causal rule*,

$$\begin{aligned} \text{Same-time rules: } & S_1 \sqcap S_2 \sqcap \dots \sqcap S_n \rightsquigarrow S \\ \text{Next-time rules: } & S_1 \sqcap S_2 \sqcap \dots \sqcap S_n \rightsquigarrow \bigcirc S \end{aligned}$$

where each  $S_i$  is an atomic state and  $S \in S_D$ .

**Definition (semantics)**: lets assume a set of  $S$  of states  $S_0, S_1, S_2, \dots$  and a time  $t, t \in Nat$ , then by  $S_t$  we represent the formulas which are true in the system at the state it is at time  $t$ :

$$\begin{aligned} (\mathcal{S}, t) \models S & \text{ iff } S \in S_t \\ (\mathcal{S}, t) \models \neg S & \text{ iff } S \notin S_t \\ (\mathcal{S}, t) \models S_1 \sqcap S_2 & \text{ iff } S_1 \in S_t \text{ and } S_2 \in S_t \\ (\mathcal{S}, t) \models S' \rightsquigarrow S & \text{ iff } (\mathcal{S}, t) \models \neg S' \text{ or } (\mathcal{S}, t) \models S \\ (\mathcal{S}, t) \models S' \rightsquigarrow \bigcirc S & \text{ iff } (\mathcal{S}, t) \models \neg S' \text{ or } (\mathcal{S}, t+1) \models S \end{aligned}$$

Same-time rules are required to be *stratified*. This is explained as follows [1]:

1. Stage 1 rules are:  $S_1 \sqcap S_2 \sqcap \dots \sqcap S_n \rightsquigarrow S$ , with  $S_1, \dots, S_n$  all independent. In this case  $S$  is said to be *1-dependent*. (The independent states are called *0-dependent*.)
2. Stage  $k$  rules are:  $S_1 \sqcap S_2 \sqcap \dots \sqcap S_n \rightsquigarrow S$ , with each of  $S_1, \dots, S_n$  at most  $(k-1)$ -dependent, and at least one of them being  $(k-1)$ -dependent. Then  $S$  is said to be *k-dependent*. In this case we also say that  $\bar{S}$  is *co-k-dependent*.
3. A set of same-time rules is stratified so long as for every rule in the set there is a number  $k$  such that the rule is a Stage  $k$  rule.

Causal rules are applied to transition the system from state  $S_t$  to the next state  $S_{t+1}$  in order of  $k$ -dependency, i.e., first 1-dependent, then 2-dependent and so on. See more details on this at [1].

The language presented above can be extended by covering different alternatives (e.g., nonlinear, continuous time, delays, etc). We keep it simple enough to easily allow specification of dynamic systems and a tractable combination with other features of our system to be introduced later.

### 3. Alternative and conflicting explanations

Decision-making is about evaluating alternatives, following a particular sequence of events the decision-maker is presented with a query and it is her/his task to assess the possible options to follow and make a decision. In this section we focus on the process of gathering the explanations available. At a later section we consider how to assess them.

Whenever a query about a particular state  $S$  of the system is passed to the DM our system will assist the analysis by considering the causal structure leading to that particular state at a particular point in time. The assistance provided comes in the form of an explanation, i.e., it provides details on what events  $e_1, e_2, \dots$  are meaningful for that state to be reached and what causal laws  $r_1, r_2, \dots$  governing the system are being exercised when events in the system can cause a state  $S$  to hold. We will call that a *causal explanation*.

There may be more than one possible explanation for how the modelled system can reach  $S$ . These explanations can be consistent with each other in which case there is no conflict. But it may also be, due to the ambiguity or lack of information characteristic in real-time applications, that some of the explanations are contradictory or somehow antagonistic. Analyzing the quality of the explanations and why some of these are contradicting to each other is a difficult and time consuming task, to be avoided when a DM has to react to an imminent hazard. This part of our system is strongly related to previous developments in theory of temporal argumentation ([2, 3, 4, 5]).

We consider possible competing causal explanations  $\langle c_1, s_1 \rangle$  and  $\langle c_2, s_2 \rangle$  at time  $t$ , where  $c_1$  is a causal structure (containing causal rules  $r_i$ ) explaining why the system may reach state  $s_1$  at time  $t$  and  $c_2$  an alternative causal structure (based on a set of causal rules  $r_j$ , disjoint with  $r_i$ ) explaining why the system may reach state  $s_2$  at time  $t$ . These causal explanations can be such that  $s_1$  contradicts  $s_2$  or contradicts the possibility that  $c_2$  may exist. In this paper we focus on the way these possible causal explanations can be found and also on the potential scenarios that can cause two possible explanations to be mutually contradictory or undermining (similarly to *rebuttal* or *undercutting* arguments [6]).

### 4. Airport security

Hazards can arise in many contexts and for many reasons. The following scenario describe situations involving hazards and possible explanations for them, each explanation will be suggesting different courses of action.

Suppose an airport gathers information that a bomb may have been placed there. To react to this specific event and avoid potential disaster, the system contacts national intelligence departments (NIDs) to collect the latest information on airport related terrorist threats. It also checks all surveillance information within the airport, such as video footage, and any reported suspected incidents. Lets assume no feedback has indicated attacks either being planned or underway. Facing with the bomb threat the airport authority has to make a hard decision on what to do next. While keeping the airport operating normally is critical, protecting human life from a bomb attack is certainly of paramount importance.

Lets assume we have the following simplified description of the decision making process. States are: *bombAlert* (a bomb alert has been issued), *assesmRisk* (assessment of real risk is needed), *nidReq* (NID is requested information on current threats), *checkLI* (check current security related reports at the airport), *nidRepOK* (NID reports everything is normal), *localInfoOK* (security related information from the airport is normal), *source1* (one source of information), *source2* (another source of information), *emergency* (state of emergency is declared), *suspectArrived* (suspect is within the airport), *passedC* (suspect passed security checks), *resultALowR* (report indicates low risk), *resultAHiMeR* (report indicates medium or high risk).

Independent states are: *bombAlert*, *resultALowR*, *resultAHiMeR*, *-resultALowR*, and *-resultAHiMeR*. *nidRepOK* and *localInfoOK* are known to be *true* initially and all the rest *false*. The following events occur during the development of activities: initially *occurs(ingr(bombAlert))* at instant 1:2, *occurs(ingr(source1))* and *occurs(ingr(-source2))* at 3:4, *occurs(ingr(suspectArrived))* at 5:6, *occurs(ingr(passedC))* at 6:7, *occurs(ingr(resultALowR))* and *occurs(ingr(-resultAHiMeR))* at 8:9. The following causal rules captures the basic aspects of the scenario:

$$\begin{aligned} bombAlert &\rightsquigarrow assesmRisk \\ bombAlert &\rightsquigarrow nidReq \\ bombAlert &\rightsquigarrow checkLI \end{aligned}$$

$$\begin{aligned} -nidRepOK &\rightsquigarrow \bigcirc emergency \\ -localInfoOK &\rightsquigarrow \bigcirc emergency \\ nidRepOK \sqcap localInfoOK \sqcap reliableSource &\rightsquigarrow \bigcirc emergency \\ nidRepOK \sqcap localInfoOK \sqcap unrelSource &\rightsquigarrow \bigcirc -emergency \end{aligned}$$

$$\begin{aligned} suspectArrived &\rightsquigarrow -localInfoOK \\ suspectArrived \sqcap passedC &\rightsquigarrow localInfoOK \end{aligned}$$

$$\begin{aligned} nidReq \sqcap resultALowR &\rightsquigarrow nidRepOK \\ nidReq \sqcap resultAHiMeR &\rightsquigarrow -nidRepOK \end{aligned}$$

Based on the above rules, our Prolog-based prototype can extract the two possible competing explanations (*A1* and *A2*) in relation to the declaration of emergencies:

$$\begin{aligned} &[\{nidRepOK \sqcap localInfoOK \sqcap reliableSource \rightsquigarrow \\ &\quad \bigcirc emergency\}, emergency] \\ &[\{nidRepOK \sqcap localInfoOK \sqcap unrelSource \rightsquigarrow \\ &\quad \bigcirc -emergency\}, -emergency] \end{aligned}$$

## 5. Dynamic POSet generation for explanation preference handling

Each time two or more explanations differ in the conclusion they have to be analyzed to decide which of them, if any, can be considered the most credible explanation(s). This section gives a more precise description on the process to compare and select competing explanations.

Typically the competing explanations can be arranged in a partial order and our approach will take that as the departing point aiming to generate a POSet out of the set of explanations being considered. In occasions it will be possible to order the options into a total order if there are a few options. Notice that we cannot ensure the POSet will be a lattice as we leave open the type of order relationships that can be used (provided the user gives an algorithm on how to use them) and therefore we cannot ensure the POSet will have a greatest lower bound or a lowest upper bound. The process can be generically depicted as in Figure 1. The possible explanations are given to a specific module of the system which evaluates them based on domain specific and also general criteria to assess which is their relative strength.

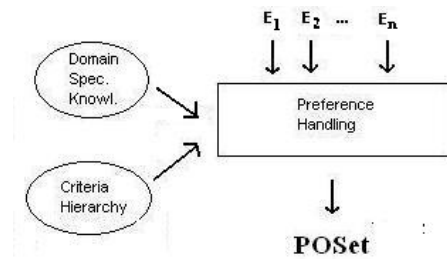


Fig. 1: Preference POSet Generation

A structure to evaluate the preference POSet of explanations can be syntactically constructed as follows:  $\langle A, B, C, D, Z, (1, 0) \rangle$  where *A* is a set of alternative explanations;  $B \in Z$  is a validity benchmark

(what is the minimum value for an explanation to be valid/useful?);  $C$  a criteria to compare explanations and build a POSet and  $D$  is the domain specific information. Here  $C$  is also a complex structure  $\langle C_1, C_2, C_3 \rangle$  where  $C_1$  is a set of criteria  $\{CR_1, CR_2, \dots, CR_m\}$ ;  $C_2$  is a set of order meta-relationships defined over different criteria  $\{M_1, M_2, \dots, M_n\}$  and  $C_3$  is a meta-criterion defined over elements of  $C_1$  using  $C_2$ .  $Z$  is a range of possible values used to rank the strength of the explanations. Symbol “1” represents the top element and “0” represents the bottom element,  $\{“1”, “0”\} \subseteq Z$ . For example, lets assume we have a structure  $\langle A, B, C, D, Z, (1, 0) \rangle$  defined over two possible explanations  $a_1$  and  $a_2$ :  $\langle \{a_1, a_2\}, “acceptable”, \langle \{S, T, EP\}, \{ \prec \}, \{S \prec T \prec EP\} \rangle, \{“excellent”, “verygood”, “good”, “acceptable”, “bad”\}, (“excellent”, “bad”) \rangle$ .

Lets assume  $S$  stands for Specificity ([7], [5]),  $T$  for Trust and  $EP$  for Event Probability.  $\prec$  is the meta-relation ‘precedence’ which in the case of  $S \prec T \prec EP$  indicates that our procedure will first apply  $S$  and if that is not enough to decide then will apply  $T$  and if that is not enough it will apply  $EP$ . If this is not enough then the explanations have equal strength. It may also be that one or both explanations  $a_i$  and  $a_j$  cannot be judged using some of this criteria so  $a_i$  and  $a_j$  will be said ‘incomparable’. For the purpose of evaluating and comparing different explanations we will use a function  $V(a_i)$  such that  $V : A \times D \rightarrow Z$  which will measure the strength value of an explanation  $a_i \in A$ , possibly using information from the domain  $D$ , into a value of  $Z$ .

*Airport scenario revisited:* when a bomb alert is triggered several reassuring mechanisms are started as a consequence. Gathering information from different sources to confirm or discard the threat is essential. Information is gathered locally and also from specialized units (e.g., NID).

For the case of local information gathered we have two competing explanations:  $\langle S_1, -localInfoOK \rangle$  and  $\langle S_2, localInfoOK \rangle$  where

$$S_1 = [suspectArrived \rightsquigarrow -localInfoOK]$$

$$S_2 = [suspectArrived \sqcap passedC \rightsquigarrow localInfoOK]$$

Lets call these explanations  $S_1$  and  $S_2$  respectively. As the set of causes related to explanation  $S_1$  is entirely contained in  $S_2$  and furthermore  $S_2$  complements and extends  $S_1$  (is a “richer” explanation) then  $S_2$  should be preferred to  $S_1$ . So the conclusion of the system will be that the alarm is off at 11 and the explanation is  $S_2$ . This conflict was decided purely on syntactical basis, i.e., the structure of the explanation allows us to take a decision based on the structure of the competing explanations.  $S_2$  takes into account all that  $S_1$  has to offer and

also brings extra information on what happens after the presence of a suspect is detected. It states not only that the suspect has been detected but later on all the controls carried out over the individual has been negative in terms of identifying a cause of concern. Given that  $S_2$  is preferable to  $S_1$  by being more specific (i.e., according to criteria  $S$ :  $S_2 >_S S_1$ ) the preference order is in this particular case a total order (left to right meaning going from up to down):  $[1, V(S_2, D), V(S_1, D), 0]$ .

NID information is gathered as a risk estimation. For this particular problem it only matters if it is low enough for the threat to be considered. If it is high or even medium then preventive measures have to be taken which is highlighted by the rules describing the scenario. We have two competing explanations:  $\langle N_1, nidRepOK \rangle$  and  $\langle N_2, -nidRepOK \rangle$  where

$$N_1 = [nidReq \sqcap resultALowR \rightsquigarrow nidRepOK]$$

$$N_2 = [nidReq \sqcap resultAHiMeR \rightsquigarrow -nidRepOK]$$

$N_1$  and  $N_2$  are equally specific and the sources of evidence are equally strong (the same official organism) so according to the meta-criteria we consider next criteria  $EP$  which (again this is domain dependent information) is based on the statistical information pointing out to the risk of attack being more likely the case  $N_2$  by the criteria  $EP$  and the resulting preference order is in this particular case a total order:  $[1, V(N_1, D), V(N_2, D), 0]$ .

Although confirmation from NIDs officials and ‘in situ’ evidence of danger suggest to ignore the call and even when the intuition of the DM tells everything is safe, information arrives that the source of the bomb alert can be trusted. If the DM is trying to collate all the supporting evidence from the system in terms of labelling a the situation as an emergency or not the system will consider the two possible contending explanations  $A_1$  and  $A_2$  for declaring an emergency or ignoring the threat. But at time 9 there is evidence for both, however the system will prefer explanations based on more reliable sources of information and hence  $A_1$  will be preferred. Suppose in this context we have two competing explanations:  $\langle A_1, emergency \rangle$  and  $\langle A_2, -emergency \rangle$  where

$$A_1 = [nidRepOK \sqcap localInfoOK \sqcap reliableSource \rightsquigarrow \bigcirc emergency]$$

$$A_2 = [nidRepOK \sqcap localInfoOK \sqcap unrelSource \rightsquigarrow \bigcirc -emergency]$$

$A_1$  and  $A_2$  are equally specific but the domain information in  $D$  says the source of information informing about the possible threat is a reliable source of evidence. So according to the meta-criteria we consider next criteria  $T$  and  $A_1$  is preferable to  $A_2$  by the criteria  $T$  and

---



---

<b>Input:</b>	$\langle A, B, C, D, Z, (1, 0) \rangle$ and $V : A \times D \rightarrow Z$ as defined in Section 5
<b>Output:</b>	$S \subseteq A$ such that it ranks the highest according to $C$
<b>1 repeat</b>	
2	Evaluate all elements of $A$ applying the first criteria as specified in $C_3$
3	Make $P$ the current POSet and $S$ the set of elements with highest $z \in Z$ associated such that for all $a \in S$ , if $V(a, D)$ is its strength then $V(a, D) > B$
4	<b>until</b> ( $S$ is a singleton) or (all criteria on $C_3$ have been exhausted);
5	<b>case of</b>
6	$S = \{\}$ : there is no acceptable explanation
7	$S$ is a singleton, $S = \{a\}$ : “ $a$ ” is the best explanation
8	$S = \{a_1, \dots, a_n\}$ : elements of $S$ are
9	equally strong explanations
10	<b>end case</b>

---

Fig. 2: Weighing explanations with a hybrid criteria.

the resulting preference order is in this particular case a total order:  $[1, V(A_1, D), V(A_2, D), 0]$ .

Figure 2 gives a basic description of the algorithm that is used to implement the dynamic POSet generation in the preference handling kernel of our system.

The process above only considers what the outcome of only one DM. There can be occasions where more than one DM produces a decision and then another DM above them in the hierarchy will decide which advice is better. For example, DM1 takes:  $\langle A, B, C^1, D, Z, (1, 0) \rangle$  to produce a POSet  $P_1$  and DM2 uses:  $\langle A, B, C^2, D, Z, (1, 0) \rangle$  (with  $C^1 \neq C^2$ ) producing a POSet  $P_2$ . One important problem to consider then is to sensibly merge  $P_1$  and  $P_2$  into a final  $P$ . This problem is addressed in the next section.

## 6. Combination of preferences

The decision making process involving multiple decision makers (MDM) can be formulated as follows. We have a set of decision makers (voters)  $\mathcal{V} = \{v_1, v_2, v_3, \dots, v_n\}$ , a set of alternative explanations  $A = \{a_1, a_2, a_3, \dots, a_m\}$ , and a set  $I$  of (individual) preferences where each  $\alpha \in I$  is a partial order over a subset of  $A$ .

The task of MDM is to find a total order that agrees maximally with all preferences in  $I$ . The preference re-

lation considered here is qualitative, therefore we cannot use the standard Borda score based solution [8, 9]. We present a probabilistic approach instead.

We treat a preference as a sequence, and consider the probability that one alternative is before another, based on individual preferences. We then apply the well known ORDERBYPREFERENCE algorithm [10] which takes as input such probabilities and generate an approximately optimal total order (i.e., approximately maximally agrees with the preferences). In the next section we focus on how to obtain such probabilities from given preferences.

## 6.1. Contextual probability

Let  $\Omega$  be a set,  $\mathcal{F}$  be a  $\sigma$ -field over  $\Omega$ . For example,  $\mathcal{F} = 2^\Omega$ . The elements of  $\mathcal{F}$  are called *neighborhoods*, and we can embed  $\Omega$  in  $\mathcal{F}$  so  $\Omega \subseteq \mathcal{F}$ . If there is a probability function  $\mathcal{P}$  over  $\mathcal{F}$  we have a probability space  $\langle \Omega, \mathcal{F}, \mathcal{P} \rangle$ . The *contextual probability* [11] is  $G : \mathcal{F} \rightarrow [0, 1]$  such that, for  $X \in \mathcal{F}$

$$G(X) = \sum_{E \in \mathcal{F}} \mathcal{P}(E) f(X \cap E) / K$$

where  $K = \sum_{E \in \mathcal{F}} \mathcal{P}(E) f(E)$  is normalization factor and  $f(X)$  is a non-negative measure of  $X$  satisfying  $f(X_1 \cup X_2) = f(X_1) + f(X_2)$  if  $X_1 \cap X_2 = \emptyset$ , e.g.,  $f(X) = |X|$ . For  $t \in \Omega$  and  $h \in \mathcal{F}$ , if  $t \in h$  we say  $t$  is *covered* by  $h$ . Let  $I$  be a finite sample of  $\Omega$ , where  $N = |I|$ . It has been shown that  $G$  can be estimated based on  $I$  as follows:

**Lemma 1 ([11])** For  $t \in \Omega$ ,

$$\hat{G}(t) = \frac{1}{N \times K} \sum_{x \in I} cov(t, x)$$

where  $cov(t, x) = |\{E \in \mathcal{F} : t \in E, x \in E\}|$  is the number of all neighborhoods that cover both  $t$  and  $x$ .

## 6.2. Calculating cover for sequences

Let  $\mathcal{A}$  be a finite set of symbols (explanations). A sequence  $\alpha$  is  $\{s_i\}_0^n = \{s_0, s_1, \dots, s_n\}$ , or simply,  $s_0 s_1 \dots s_n$ . We consider a data space,  $\Omega$ , of all sequences without repetition. Obviously  $\Omega$  is finite.

Consider a sequence  $\alpha$  and  $a, b \in \alpha$ . If  $a$  comes before  $b$  (from left to right) then  $a \leq_\alpha b$ . Consider two sequences  $\alpha$  and  $\beta$ . If  $\alpha$  can be obtained by removing some symbols from  $\beta$ ,  $\alpha$  is a *subsequence* of  $\beta$ , written by  $\alpha \preceq \beta$ . A *neighborhood* is a set of all sequences that share a single subsequence  $\gamma$ . We then say it is a *neighborhood with respect to  $\gamma$* . Thus a neighborhood corresponds to a sequence. We write  $\mathcal{F}$  for the set of all

---



---

**Input:** Sequence  $\alpha$  and  $\beta$   
**Output:** A list,  $N(\alpha, \beta)$ , of all common subsequences of  $\alpha$  and  $\beta$ .

```

1 Init:  $Seq = \{\epsilon\}$ ,  $ind(Seq) = 0$ ,  $L = \{\{\epsilon\}\}$ ;
2 for ( $i = 0$ ;  $i < |\alpha|$ ;  $i++$ ) do
3    $x = \alpha[i]$ 
4   if  $\beta[j] = x$  then
5      $Seq' = \emptyset$ 
6     for every  $l \in L$  do
7       if  $j \geq ind(l)$  then  $Seq' = Seq' \cup l$ 
8     end
9      $Seq = \{yx : \forall y \in Seq'\}$ 
10     $ind(Seq) = j$ 
11     $L = L \cup \{Seq\}$ 
12  end
13 end
14  $N(\alpha, \beta) = \cup_{x \in L} x$ .
```

---

Fig. 3: Gathering advise from multiple experts.

neighborhoods, or sequences. We can show that  $\mathcal{F}$  is a  $\sigma$ -field. Incidentally the data space and the  $\sigma$ -field are the same.

A neighborhood of sequence  $\alpha$  is a neighborhood with respect to a subsequence  $\gamma$  of  $\alpha$ . Therefore the set of all neighborhoods of  $\alpha$  is the set of all subsequences of  $\alpha$ , written by  $N(\alpha) = \{\gamma : \gamma \text{ is a subsequence of } \alpha\}$ . Furthermore we write  $N(\alpha, \beta)$  for the set of neighborhoods covering both  $\alpha$  and  $\beta$ , which is in fact the set of all common subsequences of  $\alpha$  and  $\beta$ . The algorithm in Figure 3 is designed to find all common subsequences.

### 6.3. The probability that one explanation is before another

For any two explanations  $a, b \in A$  we consider a sequence  $ab$ .  $a$  and  $b$  can be regarded as singleton sequences.

According to Lemma 1 we have

$$\hat{G}(ab) = \frac{1}{NK} \sum_{x \in I} cov(ab, x)$$

Recall that  $K$  is a normalization factor independent of  $ab$ ,  $N$  is the number of sequences in  $I$ , and  $cov(ab, x)$  is the number of common subsequences  $ab$  and  $x$ . Similarly

$$\hat{G}(b) = \frac{1}{NK} \sum_{x \in I} cov(b, x)$$

The probability that  $a$  is before  $b$  is then modeled by

$$\hat{G}(a|b) = \frac{\hat{G}(ab)}{\hat{G}(b)}$$

## 6.4. An example

Suppose there are three decision makers (DMs):  $dm_1, dm_2, dm_3$ , and there are five alternatives  $a, b, c, d, e$ . The preferences by individual DMs are  $dm_1 : abcde$ ,  $dm_2 : ab, aced$ ,  $dm_3 : bc, aed$ . Assuming equal weighting of the decision makers we have a preference set  $I = \{abcde, ab, aced, bc, aed\}$ .

Consider a query sequence  $q = ac$ . To calculate  $G(q)$  we need to calculate  $cov(q, x)$  for every  $x \in I$ , which is the number of all common subsequences of  $q$  and  $x$ .

Consider  $x = abcde$ . The set of all subsequences of  $x$  is  $\{\emptyset, a, b, c, d, e, ab, ac, ad, ae, bc, bd, be, cd, ce, de, abc, abd, abe, acd, ace, ade, bcd, bce, bde, cde, abcd, abce, abde, acde, bcde, abcde\}$ .

The set of all subsequences of  $ac$  is  $\{\emptyset, a, c, ac\}$ . Therefore  $cov(ac, abcde) = 4$ . Similarly we have  $cov(ac, ab) = 2$ ,  $cov(ac, aced) = 4$ ,  $cov(ac, bc) = 2$ , and  $cov(ac, aed) = 2$ . Therefore  $G(ac) = \frac{1}{NK}(4+2+4+2+2) = \frac{14}{NK}$ .

Similarly  $G(ab) = \frac{14}{NK}$ ,  $G(ad) = \frac{15}{NK}$ ,  $G(ae) = \frac{15}{NK}$ ,  $G(bc) = \frac{13}{NK}$ ,  $G(bd) = \frac{12}{NK}$ ,  $G(be) = \frac{12}{NK}$ ,  $G(cd) = \frac{13}{NK}$ ,  $G(ce) = \frac{13}{NK}$ , and  $G(de) = \frac{12}{NK}$ . Applying the ORDERBYPREFERENCE algorithm we get the approximately optimal total order:  $a, b, c, e, d$ , which agrees maximally with the individual preferences by all three DMs. A decision can then be made on the basis of this total order.

## 7. Related Work

Some related systems like [5] and [4] focused already on temporal argumentation systems but both are based on one DM and one fixed preference criteria. The former uses specificity whilst the second only counts the number of arguments in favour and opposed to a claim. We believe different contexts and different DMs will impose different criteria and therefore for a system to be of practical use that component of the system has to allow more flexibility.

Research and endeavor towards disaster management have been undertaken and exercised in various initiatives under different banners. For example, the United Nation started the International Decade for Natural Disaster Reduction (IDNDR) programme in late 1980s and the follow-up International Strategy for Disaster Reduction (ISDR) programme in late 1990s [12].

The application of information technologies to enhance disaster management [13] has been a constant centre of research, which explores the critical and evolving role of IT and its infrastructure, in particular, the communication and information sharing. Recently using agent technologies for disaster management has got momentum with a number of agent-based disaster management simulation systems being developed [14, 15]. These systems provide disaster management experience for decision makers through training and experiments; and technically they concentrate on the co-ordination of agents. Instead we focus here on real-time support for DMs. Our framework assist DMs to evaluate situations and make decisions as a threat is unfolding. The system can be used to detect particular events and warn on possible threats as well as to predict a possible outcome and give an explanation associated with the conclusion. Advice to the DMs is based on decision criteria which can be adapted to different domains and the opinion of a team of advisers can be gathered to combine their views on a specific set of decisions to be made.

## 8. Conclusions

We provide a system which combines basic important features to the decision-making process. The use of causality reasoning based on the temporal evolution of a scenario provides a natural way to chain meaningful events and possible states of the system. Allowing the specification of possible conflicting situations also brings a very typical element in these domain as there is usually several ways to analyze a problem and several possible competing strategies to follow and it is often the case that information originating in different sources can be inconsistent. A decision procedure to decide on those conflicting situations has been explained which serves as a decision making framework involving not only one DM but also a hierarchy of DMs.

## Acknowledgements

We would like to thank Dr. Liming Chen from the University of Ulster at Jordanstown who provided valuable feedback at early stages of this paper. The work has been partially supported by the Research Project TIN2006-02121.

## References

- [1] A. Galton and J.C. Augusto. Two approaches to event definition. In A. Hameurlain, R. Cicchetti, and R. Traummüller, editors, *Proceedings of DEXA*

- 2002, pp.547–556. LNCS Vol. 2453 - Springer Verlag, 2002.
- [2] G. Ferguson and J. Allen. Trips: An integrated intelligent problem-solving assistant. In *Proceedings of 15th National Conference on AI*, pp.567–573, 1998.
- [3] C. Chesñevar, A. Maguitman, and Ronald Loui. Logical models of argument. *ACM Computing Surveys*, 32(4):337–383, 2000.
- [4] J. Fox and S. Das. *Safe and Sound, Artificial Intelligence in Hazardous Applications*. AAAI Press/MIT Press, 2000.
- [5] J. C. Augusto and G. R. Simari. Temporal defeasible reasoning. *Knowledge and Information Systems*, 3(3):287–318, 2001.
- [6] J.L. Pollock. *Cognitive Carpentry. A Blueprint for How to Build a Person*. MIT Press, 1995.
- [7] F. Stolzenburg, A. Javier García, C. Chesñevar, and G. Simari. Computing generalized specificity. *Journal of Applied Non-Classical Logics*, 13(1):87–113, 2003.
- [8] Kenneth J. Arrow. *Social Choice and Individual Values*. John Wiley and Sons, New York, 1963.
- [9] T.N. Tideman. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4:185–206, 1987.
- [10] W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999.
- [11] H. Wang and W. Dubitzky. A flexible and robust similarity measure based on contextual probability. In *Proceedings of IJCAI’05*, pp.27–32, 2005.
- [12] ISDR. International Strategy for Disaster Reduction (ISDR), 2006. URL: [www.fire.uni-freiburg.de/programmes/un/idndr/idndr.html](http://www.fire.uni-freiburg.de/programmes/un/idndr/idndr.html).
- [13] CSTB, editor. *Proceedings of “Workshop on the use of Information Technologies to enhance disaster Management”*, The Keck Center of the National Academies, Washington, D.C., June 22-23 2005. URL: [www7.nationalacademies.org/CSTB/project\\_fema\\_workshop.html](http://www7.nationalacademies.org/CSTB/project_fema_workshop.html).
- [14] N. Schurr, J. Marecki, P. Scerri, J. P. Lewis, , and M. Tambe. *The defacto system: Training tool for incident commanders*, volume Proceedings of IAAI’05. AAAI Press, 2005.
- [15] J. Buford, G. Jakobson, L. Lewis, N. Parameswaran, and P. Ray. *D-AESOP: a situation aware BDI agent system for disaster situation*, volume Proceedings of the First International Workshop on Agent Technology for Disaster Management. Japan, 2006.