Causal interaction: from a high-level representation to an operational event-based representation

Irène Grosclaude, Marie-Odile Cordier and René Quiniou
IRISA, Campus de Beaulieu, 35042 Rennes Cedex FRANCE
igroscla,cordier,quiniou@irisa.fr

Abstract

We propose to extend the temporal causal graph formalisms used in model-based diagnosis in order to deal with non trivial interactions like (partial) cancellation of fault effects. A high-level causal language is defined in which properties such as the persistence of effects and the triggering or sustaining properties of causes can be expressed. Various interaction phenomena are associated with these features. Instead of proposing an ad hoc reasoning mechanism to process this extended formalism, the specifications in this language are automatically translated into an event calculus based language having well-established semantics. Our approach improves the way fault interaction and intermittent faults are coped with in temporal abductive diagnosis.

1 Introduction

In the field of model-based diagnosis, which aims at explaining abnormal observations, *causal graphs* [Console *et al.*, 1989] have been widely used to formulate the "deep knowledge" specifying how a faulty system may evolve. Such a graph models the causal chains leading from initial faults to observable symptoms. The causal graph is used abductively in order to compute the set of initial faults explaining the observations (for example [Brusoni *et al.*, 1995]).

In most diagnostic applications, the temporal aspect appears to be crucial, either because some effects depend on the duration of fault occurrences or on their order, or because different faults leading to similar effects can be discriminated by the order in which symptoms appear or by the delays separating symptom occurrences. Causal graphs have been extended to take time into account [Brusoni *et al.*, 1995]: the nodes represent time dependent propositions (fluents [Sandewall, 1994]); the edges correspond to formulas:

$$C_1 \ \ldots \ C_n \ cause \ E \ \ \{Temporal\ Constraints\} \qquad (1)$$

Such a causal relation can be informally interpreted as: the conjunction of facts $C_1 \ \ldots \ C_n$ leads to the occurrence of the effect E. Usually, the temporal constraints indicate the delay before the effect occurs. But they can also impose a minimal duration for the cause to produce the effect, or state that the effect has a maximal duration.

Though one of the claimed advantages of this kind of deep knowledge is the ability to represent fault interaction, paradoxically, existing diagnosis abductive algorithms [Brusoni *et al.*, 1997; Gamper and Nejdl, 1997] deal with limited forms of interaction. The language is restricted to positive effects and the interaction of faults reduced to additional effects. Moreover, unique fault and absence of uncertainty are currently hypothesized, which limits the interaction problem.

In real cases however, interaction can be much more complex: the effects of a fault can prevent, delay or accelerate the appearance of the effects due to other faults. In the medical domain for instance, the patient is often under treatment when the diagnosis is performed: the beneficial effects of drugs - interacting with the disease effects - must be taken into account as well as their secondary effects [Long, 1996]. Another kind of interaction is illustrated by the conjunction of two diseases leading to the absence of symptoms that are characteristic of one of the diseases. Existing approaches dealing with interaction in causal models [Gazza and Torasso, 1995; Long, 1996] propose to extend the causal language. First of all, negative effects ($not(E)$) are allowed in causal relations in order to express preventing effects. Secondly, the causal ontology is enriched by introducing knowledge on the causes and the effects. Thirdly, priorities between causal relations can be asserted. The advantage of this method is a relatively easy incremental knowledge acquisition. The weakness of existing approaches is the use of ad hoc reasoning mechanisms, able to manage the different interaction schemes based on the particular features of the causal relations, but dependent on the chosen causal ontology.

The problem of interacting faults is strongly related to the problem of concurrent actions representation encountered in the domain of action modeling. Deducing indirect effects of actions (the so-called ramification problem) requires to take the way they interact into account. In the theories of action and change, the problem has first been viewed as how to integrate the treatment of interaction in existing formalisms, and has been partially solved by introducing state constraints and conditional effects into action rules [Boutilier and Brafman, 1997; Miller and Shanahan, 1999]. A now quite common way to tackle the ramification problem is to associate to the set of rules describing the direct effects of actions a set of causal rules modeling their indirect effects [Giunchiglia and Lifschitz, 1998;

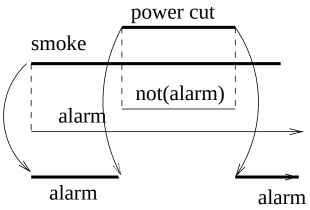

Figure 1: Example 1

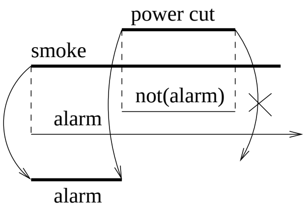

Figure 2: Example 2

Denecker *et al.*, 1998]. An interesting method consists in using the notion of *influence* [Karlsson *et al.*, 1998]: actions have no direct effects but influences. Separate rules describe the effects resulting from different conjunctions of influences. The main drawback of these approaches is the difficulty of knowledge representation.

This paper proposes to combine the advantages of both preceding approaches by using a high-level causal language dedicated to the interaction problem. This language is powerful enough to express various forms of interaction. An automatic translation into an event calculus variant [Kowalski and Sergot, 1986] makes explicit the implicit knowledge embedded in causal relations and provides clear semantics to the language. Moreover, this translation step makes the reasoning mechanisms no longer dependent on the ontology of causality as it produces an intermediate representation which can be exploited by dedicated algorithms. This work was motivated by an industrial application in the field of diagnosis. An efficient abductive diagnosis algorithm, making use of heuristics and specialized temporal constraint managers, has been developed [Grosclaude and Quiniou, 2000].

Section 2 introduces interaction leading to masked or delayed effects on several motivating examples. In section 3, we describe the formalism of the extended causal graph and explain in section 4 how it can be translated into the event calculus. In section 5, we evoke the application of the method to diagnosis. Our proposition is compared to related work in section 6 and we conclude in section 7.

2 Motivating examples

Example 1 A fire alarm starts ringing immediately after smoke has been detected and keeps ringing as long as smoke remains. The alarm is electrical, so may be subject to power cut. In a causal graph the situation could be described by:

$smoke\ cause\ alarm\ \ \{I_a\ begun\ by\ I_s\}$

$power\_cut\ cause\ not(alarm)\ \ \{I_{na}\ equals\ I_{pc}\}$

where I_a, I_s, I_{na} and I_{pc} are the temporal extents associated to the basic propositions $alarm$, $smoke$ and $power\_cut$. *begun by* and *equals* correspond to Allen temporal relations.

Let us suppose that smoke is observed and that a power cut occurs, as illustrated by fig. 1. To derive the common sense consequences of these interacting events, some extra knowledge is necessary: the power cut takes precedence over the smoke, the effect of the power cut is not persistent after power is back; the smoke sustains the alarm and triggers it again after cancellation.

Here follows a way to represent the situation by expliciting the relations between the start/end instants of the phenomena:

a) - the start of smoke causes the alarm to ring if power is on;

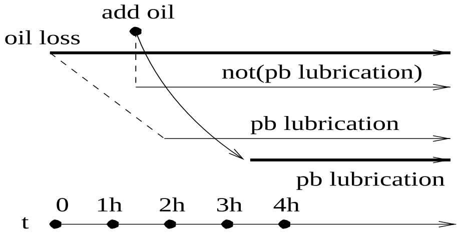

Figure 3: Example 3

b) - the start of the power cut provokes the end of the alarm if the alarm is ringing;

c) - at the end of the power cut the alarm starts again if some smoke is still present.

Example 2 Figure 2 differs from the previous one by the fact that the fire alarm is only triggered when a particular smoke concentration is reached. The difference is that the "smoke" cause is instantaneous and triggers the alarm at inception instead of sustaining it from beginning to end. In this case, the alarm will not reappear at the end of the power cut (the smoke concentration is already above the triggering threshold). The a) and b) relations are valid but the c) relation is not.

Example 3 The example represented in fig. 3, taken from [Gazza and Torasso, 1995], involves on the one hand, an oil loss from a motor leading to a lubrication problem if it lasts more than 2 hours and, on the other hand, an addition of oil. The effects of the two interacting phenomena correspond to the following relations:

d) - the oil loss provokes a lack of lubrication 2 hours after its start if no oil is added during this time;

e) - the oil addition stops a lubrication problem;

f) - if oil is added during oil loss the lubrication problem is delayed to 2 hours later (if no oil is added meanwhile).

In the following, we propose a high-level language in which properties of the causal relations influencing interaction phenomena can be expressed. We show how these relations inducing constraints on the starts/ends of causes and effects can be translated into axioms in an event calculus formalism.

3 The high-level language ECG

The previous examples show that the interaction outcomes can be derived from extra knowledge on the causal relations. In this section we propose a high-level representation language which corresponds to an extended causal graph (ECG).

The entities we consider are propositional descriptions of situations, events or properties and are referred to as fluents ([Sandewall, 1994]). Since we are dealing with time, the truth value of an entity is associated to time intervals. The ECG is composed of a set of causal relations having the following form:

$$[\mathit{TYP}_{C_1}]C_1 \wedge \cdots \wedge [\mathit{TYP}_{C_n}]C_n\ \ \{MD\}\ \ causal\_link\ E\ \ \{D\}$$

where the causes C_i and the effect E are in a positive or negative form. The set of causes is referred as the Cause. The time interval associated to the Cause is the maximal time interval where all the causes are simultaneously true[1]. Under the con-

---

[1] This is a simplifying choice and nothing prevents from extend-

Example 1:	(1.a)	$[C] smoke\ cause\ alarm\ \ \{0\}$
	(1.b)	$[CN] power\_cut\ impose\ [non\_pers]\ not(alarm)\ \ \{0\}$
Example 3:	(1.c)	$[C] oil\_loss \wedge [C] not(add\_oil)\ \{2h\}\ cause\ pb\_lubrication\ \ \{2h, \infty\}$
	(1.d)	$[OS] add\_oil\ impose\ not(pb\_lubrication)\ \{0\}$
Example 4:	(1.e)	$[C] sparks \wedge [CN] gasoline\_loss\ cause\ [non\_pers]\ flames\ \ \{0\}$

Figure 4: Examples of causal rules in the ECG formalism

dition that this interval is non empty, the effect occurs and its validity interval satisfies the temporal conditions[2]. The temporal features of a causal relation are given by the delay D and the minimal duration of the Cause *MD*. The delay $D = [d_{min},\ d_{max}]$ represents the time units between the start of the Cause and the start of the effect. The minimal duration of the Cause *MD* indicates how long the Cause must last to produce the effect.

This language is particularly suited to the expression of interaction patterns from which the precise effects of interacting causes will be deduce. The causal ontology has been extended in order to express priorities between causal relations, to specify two forms of effect persistence and different kind of causal relations (as trigger or sustain).

- the **causal-link** can be one of {cause, impose, may_cause, may_impose}. $cause$ and $impose$ express the strength of the causal relation, *cause* being weaker than *impose*. When two causes have opposite effects, the causal link with the lowest priority produces its effect only if no causal link with a higher priority is enabled[3]. In example 1, represented in the ECG in fig. 4, smoke causes alarm and power-cut imposes not(alarm). may_cause and may_impose express the uncertainty of the causal relation as in [Console *et al.*, 1989].

- the **type of effect persistence** indicates whether an effect lasts even when its inducing cause has disappeared or if it does not survive it. If qualified by *non_pers* effects are non persistent (*non_pers(d)* means that the effect disappears after a delay d), otherwise they are persistent by default.

- the **type of causality** TYP_{C_i} is one of {OS, C, CN} and indicates for each cause whether it triggers or sustains the effect. When a OS (One-$shot$) cause comes true it *triggers* the effect. In fig. 2, reaching a given level of smokiness triggers the alarm; the alarm keeps on ringing until some other fact stops it. A C ($Continuous$) cause *sustains* the effect: the effect is "continuously" triggered by the cause. The effect can be temporarily masked but reappears later on if the cause is still present. In example 1, the smoke sustains the alarm which, even interrupted by a power-cut, can reappear. A *CN* (for *Continuously-Necessary*) cause is needed for a non-persistent effect to persist. In example 4 (fig. 4), gasoline is a *CN* cause: flames disappear when there is no more gasoline.

Integrity constraints complete the description of the causal graph and allow to specify the maximal duration of a state.

Note that, contrary to [Gazza and Torasso, 1995], the type of causality is assigned to each individual cause, as each one can play a different role. See example 4 (fig. 4) where the sparks are only necessary to trigger flames, whereas gasoline loss must be continuously present.

4 From the high-level causal language to event calculus

The high-level language presented above makes easier the representation of temporal causal relations. Sentences in this language are then translated into formulas of an intermediate language, a variant of the Event Calculus [Kowalski and Sergot, 1986]. These formulas express the temporal relations existing between the starts or ends of causes and the starts or ends of the produced effects. Reasoning tasks, such as abductive explanation for diagnosis, rely upon this language. This approach has several benefits: on the one hand, a clear semantics is provided to our causal formalism; on the other hand, the causal ontology for interaction can be easily completed or changed by updating the set of translation rules without modifying the reasoning tasks. First, we present the event calculus formulation that we have used and then the rules translating (extended) causal relations into event calculus sentences.

4.1 Event calculus

The event calculus is a logic-based formalism for representing actions or events and their effects. We use a classical version of the event calculus based on i) a set of time points isomorphic to the non-negative integers, ii) a set of time-varying properties (fluents) and iii) a set of events related to the start and end of fluents and named $start(P)$ and $end(P)$. The classical event calculus axioms are adapted:

$$\begin{aligned} &holdsAt(P, T_2) \leftarrow happens(start(P), T_1) \wedge \\ &\qquad T_1 < T_2 \wedge \neg clipped(T_1, P, T_2) \\ &clipped(T_1, P, T_2) \leftarrow happens(end(P), T) \wedge \\ &\qquad T_1 < T \wedge T < T_2 \end{aligned}$$

We need to express the maximal validity interval of propositions. For this purpose, we introduce the predicate $holdsOnMax(P, T_1, T_2)$ which means that the interval bounded by T_1 and T_2 is a maximal validity interval for the fluent P: P is true throughout this interval and is false just before and just after. The constant $infinity$ is introduced to represent the latest instant in time.

$$\begin{aligned} &initiated(P, T) \leftarrow happens(start(P), T) \wedge \\ &\qquad \neg\ holdsAt(P, T) \\ &terminated(P, T) \leftarrow happens(end(P), T) \wedge \\ &\qquad holdsAt(P, T) \\ &holdsOnMax(P, T_1, infinity) \leftarrow initiated(P, T_1) \wedge \\ &\qquad \neg\ clipped(T_1, P, infinity) \\ &holdsOnMax(P, T_1, T_2) \leftarrow initiated(P, T_1) \wedge \\ &\qquad \neg\ clipped(T_1, P, T_2) \wedge happens(end(P), T_2) \end{aligned}$$

---

ing the formalism to set other constraints between the causes C_i.

[2] The unique effect is not restrictive as multiple effects can be expressed by several causal relations with the same Cause.

[3] The two degrees of priority can clearly be extended to an ordered set of priorities, as in [Gazza and Torasso, 1995].

$$\begin{aligned}
&\text{(2.a)}\ \forall T_C \exists ! T_E\ happens(start(alarm), T_E) \leftarrow \\
&\qquad T_E = T_C \wedge initiated(smoke, T_C) \wedge \neg(\exists T_1, T_2\ holdsOnMax(power\_cut, T_1, T_2) \wedge T_1 \le T_C \wedge T_C \le T_2) \\
&\text{(2.b)}\ \forall T_C \exists ! T_E\ happens(end(alarm), T_E) \leftarrow T_E = T_C \wedge initiated(power\_cut, T_C) \\
&\text{(2.c)}\ \forall T_C \exists ! T_E\ happens(start(alarm), T_E) \leftarrow \\
&\qquad T_E = T_C \wedge terminated(power\_cut, T_C) \wedge (\exists T_1, T_2\ holdsOnMax(smoke, T_1, T_2) \wedge T_1 \le T_C \wedge T_C \le T_2)
\end{aligned}$$

Figure 5: Event calculus domain axioms related to example 1 (fig. 4)

The basic event calculus version is extended to take into account indirect effects of events (the ramification problem) and uncertain delays. The domain axioms are computed from causal relations expressed in the ECG. The translation rules are described in section 4.2.

In our representation, positive information has been emphasized as this is common practice when representing expert causal knowledge. Negative information, such as $\neg holdsAt(P, T)$ or $\neg clipped(T_1, P, T_2)$, can be deduced by non-monotonic reasoning (negation as failure as in logic programming) or transformation (completion or circumscription as in [Miller and Shanahan, 1999]). The negation symbol $\neg$ in event calculus sentences must be interpreted as non-monotonic negation.

4.2 Translation rules

The principle underlying the translation of the causal graph is to make explicit the links between the starts or ends of causes and the starts or ends of effects. The causal relations leading to contradictory effects must be treated globally. They are gathered and translated into several domain axioms of the form (examples are given in fig. 5):

$event2 \leftarrow$ *Temporal_constraints* $\wedge\, event1 \wedge \neg Prev \wedge Nec$

The *Temporal constraints* relate the occurrences of $event1$ and $event2$. Nec represents the preconditions necessary to the realization of the relation whereas $Prev$ are the preconditions preventing the relation. If the causal relation is uncertain (may_cause or may_impose) an abstract precondition α_i is added in the body of the axiom [Console *et al.*, 1989]. The translation rules are given below. In a first step, we assume that the causal relation antecedents only contain a single cause. Later on, we explain how a conjunction of causes can be treated as a single cause. The typefaces of cause symbols (boldface or underline) will be used to explain the treatment of conjunctive causes.

Rule 1: strong positive causation - causal link $impose$; positive effect.

$$\frac{\mathbf{C}\ impose\ E\ \{D\}}{\begin{aligned}&\forall T_C \exists ! T_E\ happens(start(E), T_E) \leftarrow \\ &\qquad T_E = T_C + D \wedge initiated(\mathbf{C}, T_C)\end{aligned}}$$

Rule 2: weak positive causation - causal link $cause$; positive effect.

$$\frac{\mathbf{C}\ cause\ E\ \{D\}\ \{R'_i \mid \underline{\mathbf{C}}'_i\ impose\ not(E)\ \{D'\}\}}{\begin{aligned}&\forall T_C \exists ! T_E\ happens(start(E), T_E) \leftarrow \\ &\qquad T_E = T_C + D \wedge initiated(\mathbf{C}, T_C) \wedge \neg Prev\end{aligned}}$$

The rules R'_i leading to $not(E)$ are taken into account by inserting the precondition $\neg$ *Prev*. This ensures that $start(E)$ does not happen during an interval where $not(E)$ is imposed by a relation R'_i. $Prev$ is the expression:

$$\bigvee_i \exists T_{i1}, T_{i2}\ holdsOnMax(\underline{\mathbf{C}}'_i, T_{i1}, T_{i2}) \wedge TC_i(T_{i1}, T_{i2}, T_{\mathbf{C}})$$

With regard to the delays, two cases are considered:

a) the delays are precise: $D = [d, d]$ and $D' = [d', d']$.
The temporal constraints TC_i are: $T_C + d \ge T_{i1} + d'$.
When the effect $not(E)$ is not persistent and disappears at the end of $\underline{\mathbf{C}}'_i$ after a delay D"=[d",d"] the constraint $T_C + d \le T_{i2} + d''$ is added.
As an illustration, Rule 2 is used to translate the causal rule 1.a (fig. 4) into the axiom 2.a (fig. 5).

b) the delays are imprecise. Then, constraints on the relative temporal position of the interacting causes are not sufficient to ensure that $start(E)$ does not happen during an interval where $not(E)$ is imposed. We solve the problem by introducing intermediate effects representing the positive or negative influences of interacting causes. A causal relation *[TYP$_C$] C causal_relation E {D}* is replaced by the two relations (1) *[TYP$_C$] C cause InflE {D}* and (2) *[CN] InflE causal_relation E {0}*. This way, the interaction involves relations of type (2) with precise (null) delays. The final effects are produced by the starts and ends of influences, following the relative position of the opposite influences.

Rule 3: strong negative causation - causal link $impose$; negative effect.

$$\frac{\mathbf{C}\ impose\ not(E)\ \{D\}}{\begin{aligned}&\forall T_C \exists ! T_E\ happens(end(E), T_E) \leftarrow \\ &\qquad T_E = T_C + D \wedge initiated(\mathbf{C}, T_C)\end{aligned}}$$

Rule 3 is used to translate the causal rule 1.b (fig. 4) into the axiom 2.b (fig. 5).

Rule 4: weak negative causation - causal link $cause$; negative effect.

$$\frac{\mathbf{C}\ cause\ not(E)\ \{D\}\ \{R'_i | \underline{\mathbf{C}}'_i\ impose\ E\ \{D'\}\}}{\begin{aligned}&\forall T_C \exists ! T_E\ happens(end(E), T_E) \leftarrow \\ &\qquad T_E = T_C + D \wedge initiated(\mathbf{C}, T_C) \wedge \neg Prev\end{aligned}}$$

where $Prev$ is the same as in Rule 2.

Rule 5: influence of causation on end instants.

$$\frac{\begin{aligned}&[CN]\underline{\mathbf{C}}\ cause/impose\ E\ \{D\} \\ &\qquad \{R'_i | \underline{\mathbf{C}}'_i\ cause\ E\ \{D'_i\}, \underline{\mathbf{C}}'_i \neq \underline{\mathbf{C}}\} \\ &\qquad \{R'_j | \underline{\mathbf{C}}'_j\ impose\ E\ \{D'_j\}, \underline{\mathbf{C}}'_j \neq \underline{\mathbf{C}}\}\end{aligned}}{\begin{aligned}&\forall T_C \exists ! T_E\ happens(end(E), T_E) \leftarrow \\ &\qquad T_E = T_C + D \wedge terminated(\underline{\mathbf{C}}, T_C) \wedge \neg Prev\end{aligned}}$$

$\neg Prev$ is used to verify that no other cause produces the effect E. $Prev$ is the same as in rule 2 and 4.

$$\begin{array}{l}
\forall T_C \exists! T_E \; happens(start(C_{\wedge(')}), T_E) \leftarrow \\
\quad T_E = T_C \wedge initiated(sparks, T_C) \wedge (\exists T_1, T_2 \; holdsOnMax(gasoline\_loss, T_1, T_2) \wedge T_1 \leq T_C \wedge T_C \leq T_2 \\
\forall T_C \exists! T_E \; happens(start(C_{\wedge(')}), T_E) \leftarrow \\
\quad T_E = T_C \wedge initiated(gasoline\_loss, T_C) \wedge (\exists T_1, T_2 \; holdsOnMax(sparks, T_1, T_2) \wedge T_1 \leq T_C \wedge T_C \leq T_2 \\
\forall T \; happens(end(C_{\wedge}), T) \leftarrow terminated(sparks/gasoline\_loss, T) \\
\forall T \; happens(end(C_{\wedge'}), T) \leftarrow terminated(gasoline\_loss, T)
\end{array}$$

Figure 6: Event calculus domain axioms related to example 4 (fig. 4)

Rule 6: end of a cause masking an effect E.

$$\frac{\begin{array}{l} [CN]\underline{\mathbf{C}} \; impose \; not(E) \; \{D\} \\ \qquad\qquad \{R'_i | [C/CN]\mathbf{C}'_i \; cause \; E \; \{D'_i\}\} \\ \qquad\qquad \{R'_j | \underline{\mathbf{C}}'_j \; impose \; not(E) \; \{D'_j\}, \underline{\mathbf{C}}'_j \neq \underline{\mathbf{C}}\}\} \end{array}}{\begin{array}{l} \forall T_C \exists! T_E \; happens(start(E), T_E) \leftarrow \\ \quad T_E = T_C + D \wedge terminated(\underline{\mathbf{C}}, T_C) \wedge \neg Prev \wedge Nec \end{array}}$$

If the delays are precise: $D = [d, d]$ and $D' = [d', d']$, then Nec indicates that a cause sustaining E must be present:

$$\bigvee_i \exists T_{i1}, T_{i2} \; holdsOnMax(\mathbf{C}_i, T_{i1}, T_{i2}) \wedge TC(T_{i1}, T_{i2}, T_C)$$

The constraints TC are:

$$T_{i1} + d' \leq T_C + d \wedge T_C + d \leq T_{i2} + d'$$

$Prev$ is the same as in rule 2, 4 and 5.

Rule 6 is used to translate the causal rule 1.b (fig. 4) into the axiom 2.c (fig. 5).

If the delays are imprecise, we use the same transformation of causal relations as in rule 2, which adds intermediate influences.

Due to lack of space, we do not give the precise translation rules used when a minimal duration *MD* of the cause is necessary for the effect to occur. This constraint is expressed using the predicate $HoldsSince(P, T_1, T_2)$:

$$holdsSince(P, T_1, T_2) \leftarrow initiated(P, T_1) \wedge \neg \; clipped(T_1, P, T_2)$$

The precondition $holdsSince(C, T_1, T_2) \wedge T_2 - T_1 \geq MD$ is used to constrain the duration of the cause C.

Conjunctive causes.

We have presented translation rules operating on causal relations with single antecedent. To deal with causal relations having multiple antecedents, the fluent $C_{\wedge}$ corresponding to the conjunction of the causes (true when all the causes are true) is introduced. If the effect of the causal relation is not persistent, a second intermediate fluent $C_{\wedge'}$ is added, representing the part of the conjunction necessary for the persistence of the effect (the "sustaining" part). The start of $C_{\wedge'}$ corresponds to the start of $C_{\wedge}$, but $C_{\wedge'}$ ends only when one cause of type *CN* ends. The axioms corresponding to example 4 (fig. 4) are represented on fig. 6. During the translation process $C_{\wedge}$ or $C_{\wedge'}$ is considered as the unique cause of the causal rule. The translation rules covering the multiple causes are those given before, where **C** is replaced by $C_{\wedge}$ and $\underline{\mathbf{C}}$ by $C_{\wedge'}$. If all the causes are of type C, the unique cause is of type C. In the remaining cases, if one of the cause is *OS*, the unique cause is *OS*, else the unique cause is *CN*.

5 Application to diagnosis

Our work originates from an application in diagnosis requiring to deal with interacting faults. The problem is to explain a set of imprecisely dated observations by a set of possibly interacting faults. The observations are expressed by statements $holdsOnMax(Obs, T_1, T_2) \wedge CT(T_1, T_2)$, where T_1 and T_2 define the temporal interval on which the observation Obs has been observed and CT specifies the constraints on this possibly imprecise interval. A candidate diagnosis is expressed by using predefined "abducible" predicates: $happens(start(IC_i), T_i)$, $happens(end(IC_j), T_j)$ associated with temporal constraints on the instants T_i and T_j and where the IC are initial faults. We are then faced to a temporal abductive task as usual in a diagnostic context.

The domain knowledge is described by a set of causal relations using the formalism defined in 3 and automatically translated into the event calculus formalism as explained in 4. A first solution would have been to make use of existing abductive tools. For instance, the ACLP framework [Kakas *et al.*, 1999] integrates abductive and constraint logic programming. As such, it is well adapted for encoding variants of event calculus implementing abductive planning or scheduling tasks. But, for efficiency reasons - ACLP suffers from a lack of efficiency mainly due to its standard resolution strategy causing many backtracks - we decided to implement our own algorithm which takes advantage of domain-dependent heuristics to cope with the inherent complexity of the abductive task. This algorithm is described in [Grosclaude and Quiniou, 2000].

The method has been implemented in Java and experimented on a real causal graph provided by EDF (Électricité de France) which contains about 500 nodes and 1000 arcs. Following the approach in [Brusoni *et al.*, 1997], the consistency of temporal constraints is efficiently checked by testing the consistency of a temporal constraints network every time constraints are added, in order to reject an hypothesis as soon as possible. The tests confirm that the approach is feasible despite its inherent computational complexity. The system has been able to detect potential interaction cases. The graphical interface displays the original causal graph as well as a graphical representation of the axioms in the event calculus to the user, so as to let him notice the consequences of the interaction and correct his model if he wants to. We are currently investigating additional heuristics in order to reduce the computation time.

6 Discussion

Deduction on a causal graph including negative effects have been studied by [Gazza and Torasso, 1995]. Causal relations are categorized according to the relative position of the occurrences of the cause and the effect. The considered properties associated to causal relations, as one-shot or continuous are closed to ours. Nevertheless, since the property is associated globally to the causal relation, it is not clear how conjunctive causes of different kinds can be dealt with. With regard to reasoning with opposite effects, a priority is associated to each relation. For every pair of relations, a specific interaction scheme is determined. The approach is used in order to maintain temporal databases and makes the assumption that all the temporal constraints are precise. As far as we can see, our method has two advantages: first, patterns of interaction are clearly related to particular features of causal relations; second, interaction is represented explicitly in the domain axioms of our event based language. We propose a graphical representation of the domain axioms that can be displayed to the expert, who can detect more easily erroneous modeling leading to bad prediction of interaction.

Beyond the diagnosis area, some recent works on the deduction of indirect effects of actions have introduced causality. Few of them are explicitly dealing with temporal aspects. [Karlsson *et al.*, 1998] have proposed a logical formalism to deal with delayed and concurrent effects of actions with explicit time. The causal theory is described by a narrative in the TAL-C language, allowing to express features such as the persistence of effects and imprecise delays. In order to avoid inconsistency, the notion of influence is introduced: actions have no direct effects but influences, and rules for combining influences are given. The narrative is translated into a first-order language in order to test the consistency of a scenario containing actions and observations. As far as we can see, time is managed by logical inference and the treatment of imprecision should not be as efficient as ours which is based on temporal constraint propagation. With respect to this work, our main contribution is the high-level language which allows us to express the causal relations at an ontological level instead of at an operational one. It would be interesting to test how ECG could be used as a high-level language for TAL-C.

7 Conclusion

We have presented a formalism dedicated to reasoning on interacting causes. It takes advantage, on the one hand, of the knowledge acquisition benefits offered by causal languages developed in the abductive diagnosis context and, on the other hand, of the well-defined formalisms developed to reason about change. The idea is to express the knowledge about potentially interacting causes into a high-level language, the extended causal graph language, and to provide an automatic translation into an event calculus based formalism. The method have been experimented in an abductive diagnosis context and the complexity of the domain axioms show the relevance of their automatic computation from a high-level causal language. Moreover, updating the causal knowledge is made easier and can be performed without bothering of the whole set of relations.

In the domain of diagnostic, future work concerns the application of the method for maintenance purposes. The ability to deal with contradictory knowledge makes it possible to represent the effects of repairs in the causal graph. It is thus interesting, especially when the system is evolving slowly and when maintenance is expensive, to rely on the causal graph in order to detect the dates at which some repair is mandatory. Another perspective is related to the interest of the ECG language and its event-calculus translation for reasoning tasks involving actions and changes as in planning for instance.

References

[Boutilier and Brafman, 1997] C. Boutilier and R.I. Brafman. Planning with concurrent interacting actions. *Proc. of the 14th National Conference on Artificial Intelligence (AAAI-97)*, 1997.

[Brusoni *et al.*, 1995] V. Brusoni, L. Console, P. Terenziani, and D. Theseider-Dupré. Characterizing temporal abductive diagnosis. *Proc. of 6th Int. Workshop on Principles of Diagnosis (DX-95)*, p. 34–40, 1995.

[Brusoni *et al.*, 1997] V. Brusoni, L. Console, L. Terenziani, and D. Theseider-Dupré. An efficient algorithm for temporal abduction. *Lecture Notes in Artificial Intelligence 1321*, p. 195–206, 1997.

[Console *et al.*, 1989] L. Console, D. Theseider-Dupré, and P. Torasso. A theory of diagnosis for incomplete causal models. *Proc. of the Int. Joint Conference on Artificial Intelligence (IJCAI-89)*, p. 1311–1317, 1989.

[Denecker *et al.*, 1998] M. Denecker, D. Theseider Dupré, and K. Van Belleghem. An inductive definition approach to ramifications. *Linköping Electronic Articles in Computer and Information Science, http://www.ep.liu.se/ea/cis/1998/007/, Vol. 3(1998):7*, 1998.

[Gamper and Nejdl, 1997] J. Gamper and W. Nejdl. Abstract temporal diagnosis in medical domains. *Artificial Intelligence in Medicine 10*, p. 209–234, 1997.

[Gazza and Torasso, 1995] M. Gazza and P. Torasso. Temporal prediction: dealing with change and interaction within a causal framework. *Topics in Artificial Intelligence*, p. 79–90, 1995.

[Giunchiglia and Lifschitz, 1998] E. Giunchiglia and V. Lifschitz. An action language based on causal explanation: preliminary report. *Proc. of the 15th National Conference on Artificial Intelligence (AAAI-98)*, p. 623–630, 1998.

[Grosclaude and Quiniou, 2000] I. Grosclaude and R. Quiniou. Dealing with interacting faults in temporal abductive diagnosis. *Proc. of the 8th Int. Workshop on Principles of Diagnosis (DX-00)*, 2000.

[Kakas *et al.*, 1999] A.C. Kakas, A. Michael, and C. Mourlas. Aclp: Abductive constraint logic programming. *Journal of Logic Programming*, 44(1-3):129–177, 1999.

[Karlsson *et al.*, 1998] L. Karlsson, J. Gustafsson, and P. Doherty. Delayed effects of actions. *Proc. of the 13th European Conference on Artificial Intelligence ECAI-98*, p. 542–546, 1998.

[Kowalski and Sergot, 1986] R. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing, Vol.4*, p. 67–95, 1986.

[Long, 1996] W. Long. Temporal reasoning for diagnosis in a causal probabilistic knowledge base. *Artificial Intelligence in Medicine,8*, p. 193–215, 1996.

[Miller and Shanahan, 1999] R. Miller and M. Shanahan. The event calculus in classical logic - alternative axiomatisations. *Linköping Electronic Articles in Computer and Information Science*, 4(16), 1999.

[Sandewall, 1994] E. Sandewall. *Features and Fluents, A Systematic Approach to the Representation of Knowledge about Dynamical Systems.* Oxford University Press, 1994.