A Hierarchy of Modal Event Calculi: Expressiveness and Complexity

Iliano Cervesato[1], Massimo Franceschet[2], Angelo Montanari[2]

[1] Department of Computer Science
Carnegie Mellon University
5000 Forbes Avenue – Pittsburgh, PA 15213-3891
Phone: (412)-268-3069 – Fax: (412)-268-5576
E-mail: *iliano@cs.cmu.edu*

[2] Dipartimento di Matematica e Informatica
Università di Udine
Via delle Scienze, 206 – 33100 Udine, Italy
Phone: +39-432-558477 – Fax: +39-432-558499
E-mail: *francesc@dimi.uniud.it; montana@dimi.uniud.it*

Abstract. We consider a hierarchy of modal event calculi to represent and reason about partially ordered events. These calculi are based on the model of time and change of Kowalski and Sergot's Event Calculus (*EC*): given a set of event occurrences, *EC* allows the derivation of the maximal validity intervals (MVIs) over which properties initiated or terminated by those events hold. The formalisms we analyze extend *EC* with operators from modal logic. They range from the basic Modal Event Calculus (*MEC*), that computes the set of all current MVIs (MVIs computed by *EC*) as well as the sets of MVIs that are true in some/every refinement of the current partial ordering of events ($\Diamond$-/$\Box$-MVIs), to the Generalized Modal Event Calculus (*GMEC*), that extends *MEC* by allowing a free mix of boolean connectives and modal operators. We analyze and compare the expressive power and the complexity of the proposed calculi, focusing on intermediate systems between *MEC* and *GMEC*. We motivate the discussion by using a fault diagnosis problem as a case study.

1 Introduction

The *Event Calculus*, abbreviated *EC* [11], is a simple temporal formalism designed to model situations characterized by a set of *events*, whose occurrences have the effect of starting or terminating the validity of determined *properties*. Given a possibly incomplete description of when these events take place and of the properties they affect, *EC* is able to determine the *maximal validity intervals*, or *MVIs*, over which a property holds uninterruptedly. The algorithm *EC* relies on for the verification or calculation of MVIs is polynomial [6]. It can advantageously be implemented as a logic program. Indeed, the primitive operations of logic programming languages can be exploited to express boolean combinations of MVI computations and limited forms of quantification.

The range of the queries that can be expressed in *EC* is, however, too limited for modeling realistic situations, even when permitting boolean connectives. Expressiveness can be improved either by extending the representation capabilities of *EC* to encompass a wider spectrum of situations (e.g. by permitting precondition-triggered events), or by enriching the query language of the formalism. The first alternative is discussed in [4]; in this paper, we explore extensions to the query language relatively to a specific subclass of *EC* problems.

We limit our investigation to problems consisting of a fixed set of events that are known to have happened, but with incomplete information about the relative order of their occurrences [1, 2, 3, 7, 8, 12]. In these situations, the MVIs derived by *EC* bear little relevance since the acquisition of additional knowledge about the actual event ordering might both dismiss current MVIs and validate new MVIs [3]. It is instead critical to compute precise variability bounds for the MVIs of the (currently underspecified) actual ordering of events. Optimal bounds have been identified in the set of *necessary MVIs*, or *$\Box$-MVIs*, and the set of *possible MVIs*, or *$\Diamond$-MVIs*. They are the subset of the current MVIs that are not invalidated by the acquisition of new ordering information and the set of intervals that are MVIs in at least one completion of the current ordering of events, respectively.

The *Modal Event Calculus*, *MEC* [1], extends *EC* with the possibility of inquiring about $\Box$-MVIs and $\Diamond$-MVIs. The enhanced capabilities of *MEC* do not raise the polynomial complexity of *EC*, but they are still insufficient to model effectively significant situations. This limitation has been overcome with the *Generalized Modal Event Calculus*, *GMEC* [2]. This formalism reduces the computation of $\Box$-MVIs and $\Diamond$-MVIs to the derivation of basic MVIs, mediated by the resolution of the operators $\Box$ and $\Diamond$ from the modal logic *K1.1*, a refinement of *S4* [14]. The query language of *GMEC* permits a free mix of boolean connectives and modal operators, recovering the possibility of expressing a large number of common situations, but at the price of making the evaluation of *GMEC* queries an NP-hard problem (as we will show in Section 5). In this paper, we refine the taxonomy of the modal event calculi by considering two intermediate formalisms between *MEC* and *GMEC*, as shown in Figure 1. The queries of the *Modal Event Calculus with External Connectives*, *ECMEC*, allow combining computations of MVIs, $\Box$-MVIs and $\Diamond$-MVIs by means of boolean connectives. The approach followed in *ICMEC*, the *Modal Event Calculus with Internal Connectives*, is dual: boolean combinations of MVI computations can be prefixed by either $\Box$ or $\Diamond$. Both *ECMEC* and *ICMEC* retain enough of the expressive power of *GMEC* to allow a faithful and usable representation of numerous common situations. However, while the problem of evaluating an *ICMEC* query is still NP-hard, *ECMEC* admits polynomial implementations, making this formalism an ideal candidate for the formalization of a number of applicative domains.

The main contributions of this paper lie in the investigation of intermediate modal event calculi between *MEC* and *GMEC*, and in the individuation of *ECMEC* as an expressive but tractable sublanguage of the latter. The paper provides also a formal analysis of the complexity of various modal event calculi,

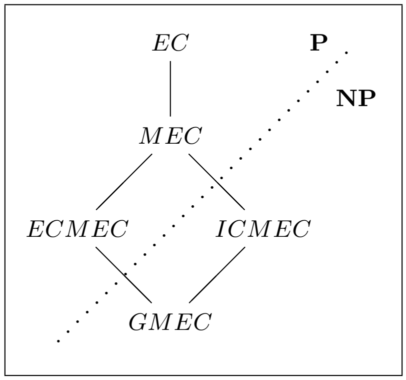

Fig. 1. A Hierarchy of Event Calculi

notably *MEC* and *GMEC*. We invite the interested reader to consult [8] for a more detailed discussion of the topics treated in this paper, and for the proofs of the statements we mention. In order to make the discussion more concrete, we interleave the presentation of our various modal event calculi with the formalization of an applicative example. The paper is organized as follows. We give an informal description of our case study in Section 2. Section 3 defines the modal extensions of *EC* we examine and discusses their main properties. In Section 4, we come back to our case study and provide a formalization. Section 5 gives a complexity analysis for the calculi considered in this paper. Finally, Section 6 summarizes the main contributions of the paper and outlines directions of future work.

2 The Application Domain: an Informal Description

In this section, we introduce a real-world case study taken from the domain of fault diagnosis. In Section 4, we will compare the formalizations obtained by encoding it in *GMEC*, *ICMEC* and *ECMEC*.

We focus our attention on the representation and information processing of fault symptoms that are spread over periods of time and for which current expert system technology is particularly deficient [13]. Consider the following example, which diagnoses a fault in a computerized numerical control center for a production chain.

> *A possible cause for an undefined position of the tool magazine is a faulty limit switch S. This cause can however be ruled out if the status registers R_1, R_2 and R_3 show the following behavior: from a situation in which all three registers contain the value* 0*, they all assume the value* 1 *in successive and disjoint time intervals (first R_1, then R_2, and last R_3), and then return to* 0.

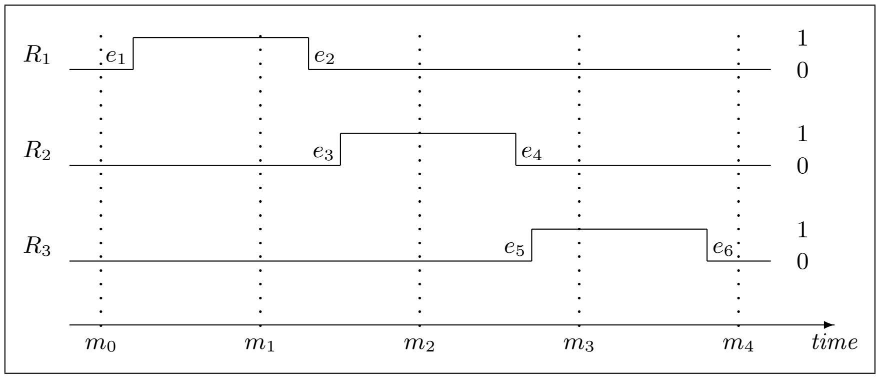

Fig. 2. Expected Register Behavior and Measurements

Figure 2 describes a possible sequence of transitions for R_1, R_2 and R_3, that excludes the eventuality of S being faulty. In order to verify this behavior, the contents of the registers must be monitored over time. Typically, each value (0 or 1) of a register persists for at least t time units. Measurements are made at fixed intervals (sampling periods), asynchronously with the change of value of the status registers. In order to avoid losing register transitions, measurements must be made frequently enough, that is, the sampling period must be less than t. However, it is not possible to avoid in general that transitions of *different* registers take place between two consecutive measurements, making it impossible to recover their relative order.

This situation is depicted, in the case of our example, in Figure 2, where dotted lines indicate measurements. Moreover, we have given names to the individual transitions of state of the different registers. In this specific situation, the values found at measurements m_0 and m_1 allow us to determine that R_1 has been set during this interval (transition e_1). The contents of the registers at measurement m_2 let us infer that R_1 has been reset (transition e_2) and that the value of R_2 has changed to 1 (transition e_3). We know that both e_2 and e_3 have taken place after e_1, but we have no information about the relative order of these transitions. Similarly, m_3 allows us to infer that, successively, R_2 has been reset (e_4) and R_3 has been set (e_5), but cannot decide which among e_4 and e_5 has taken place first. Finally, m_4 acknowledges that R_3 has successively been reset to 0 (e_6). The available information about the ordering of transitions is summarized in Figure 3.

As we will see in Section 4, this example can be easily formalized in *EC*. We will however need the expressive power of a modal version of this formalism to draw conclusions about the possibility that the switch S is faulty.

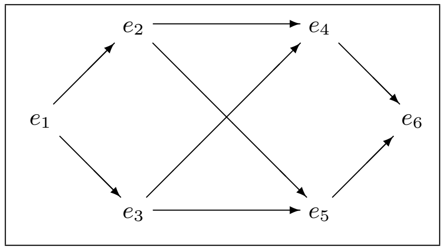

Fig. 3. Ordering of the Events

3 A Hierarchy of Modal Event Calculi

In this section, we recall the syntax and semantics of the modal event calculi *MEC* [1] and *GMEC* [2] and adapt it to define the intermediate calculi *ECMEC* and *ICMEC*. We present some relevant properties of these formalisms. These systems form the linguistic hierarchy shown in Figure 1. Implementations in the language of hereditary Harrop formulas have been given in [8], together with formal proofs of soundness and completeness with respect to the specifications below. Space limitations do not allow us to further discuss this aspect.

The *Event Calculus* (*EC*) [11] and the modal variants we consider aim at modeling situations that consist of a set of events, whose occurrences over time have the effect of initiating or terminating the validity of properties, some of which may be mutually exclusive. We formalize the time-independent aspects of a situation by means of an *EC-structure*, defined as follows.

Definition 1. (EC-structure)

A *structure* for the *Event Calculus* (abbreviated *EC-structure*) is a quintuple $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$ such that:

- $E = \{e_1, \ldots, e_n\}$ and $P = \{p_1, \ldots, p_m\}$ are finite sets of *events* and *properties*, respectively.
- $[\cdot\rangle : P \to \mathbf{2}^E$ and $\langle\cdot] : P \to \mathbf{2}^E$ are respectively the *initiating* and *terminating map* of $\mathcal{H}$. For every property $p \in P$, $[p\rangle$ and $\langle p]$ represent the set of events that initiate and terminate p, respectively.
- $]\cdot,\cdot[\subseteq P \times P$ is an irreflexive and symmetric relation, called the *exclusivity relation*, that models exclusivity among properties. □

Unlike the original presentation of *EC* [11], we focus our attention on situations where the occurrence time of events is unknown. Indeed, we only assume the availability of incomplete information about the relative order in which these events have happened. Therefore, we formalize the time-dependent aspects of an *EC* problem by providing a (strict) *partial order* for the involved event occurrences. We write $W_{\mathcal{H}}$ for the set of all partial orders on the set of events E of an

EC-structure $\mathcal{H}$ and use the letter w to denote individual orderings, or *knowledge states*, in $W_{\mathcal{H}}$ (that is, a knowledge state w is an irreflexive and transitive subset of $E \times E$). Given $w \in W_{\mathcal{H}}$, we will sometimes call a pair of events $(e_1, e_2) \in w$ an *interval*. For reasons of efficiency, implementations generally represent the current situation w as a binary *acyclic* relation o, from which w can be recovered as the transitive closure o^+ of o [2, 6]. In the following, we will often work with *extensions* of an ordering w, defined as any element of $W_{\mathcal{H}}$ that contains w as a subset. We define a *completion* of w as any extension of w that is a total order. We denote with $\mathrm{Ext}_{\mathcal{H}}(w)$ and $\mathrm{Comp}_{\mathcal{H}}(w)$ the set of all extensions and the set of all completions of the ordering w in $W_{\mathcal{H}}$, respectively. We will drop the subscript $_{\mathcal{H}}$ when clear from the context.

Given a structure $\mathcal{H}$ and a knowledge state w, *EC* offers a means to infer the *maximal validity intervals*, or *MVIs*, over which a property p holds uninterruptedly. We represent an MVI for p as $p(e_i, e_t)$, where e_i and e_t are the events that initiate and terminate the interval over which p maximally holds, respectively. Consequently, we adopt as the *query language* of *EC* the set $\mathcal{A}_{\mathcal{H}} = \{p(e_1, e_2) : p \in P \text{ and } e_1, e_2 \in E\}$ of all such property-labeled intervals over $\mathcal{H}$. We interpret the elements of $\mathcal{A}_{\mathcal{H}}$ as propositional letters and the task performed by *EC* reduces to deciding which of these formulas are MVIs and which are not, with respect to the current partial order of events. This is a problem of model checking.

In order for $p(e_1, e_2)$ to be an MVI relatively to the knowledge state w, (e_1, e_2) must be an interval in w, i.e. $(e_1, e_2) \in w$. Moreover, e_1 and e_2 must witness the validity of the property p at the ends of this interval by initiating and terminating p, respectively. These requirements are enforced by conditions (*i*), (*ii*) and (*iii*), respectively, in the definition of valuation given below. The maximality requirement is caught by the meta-predicate $nb(p, e_1, e_2, w)$ in condition (*iv*), which expresses the fact that the validity of an MVI must not be *broken* by any interrupting event. Any event e which is known to have happened between e_1 and e_2 in w and that initiates or terminates a property that is either p itself or a property exclusive with p interrupts the validity of $p(e_1, e_2)$. These observations are formalized as follows.

Definition 2. (EC intended model)

Let $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$ be a EC-structure. The *intended EC-model* of $\mathcal{H}$ is the propositional valuation $\upsilon_{\mathcal{H}} : W_{\mathcal{H}} \to 2^{\mathcal{A}_{\mathcal{H}}}$, where $\upsilon_{\mathcal{H}}$ is defined in such a way that $p(e_1, e_2) \in \upsilon_{\mathcal{H}}(w)$ if and only if

i. $(e_1, e_2) \in w$;
ii. $e_1 \in [p\rangle$;
iii. $e_2 \in \langle p]$;
iv. $nb(p, e_1, e_2, w)$, where $nb(p, e_1, e_2, w)$ iff

$$\neg\exists e \in E.\, (e_1, e) \in w \wedge (e, e_2) \in w \\ \wedge\ \exists q \in P.\, ((e \in [q\rangle \vee e \in \langle q]) \wedge (]p, q[\vee p = q))).$$

□

The set of MVIs of an *EC* problem $(\mathcal{H}, w)$ is not stable with respect to the acquisition of new ordering information. Indeed, as we move to an extension of w, current MVIs might become invalid and new MVIs can emerge [3]. The *Modal Event Calculus*, or *MEC* [1], extends the language of *EC* with the possibility of enquiring about which MVIs will remain valid in every extension of the current knowledge state, and about which intervals might become MVIs in some extension of it. We call intervals of these two types *necessary MVIs* and *possible MVIs*, respectively, using *□-MVIs* and *◇-MVIs* as abbreviations. Formally, the query language $\mathcal{B}_\mathcal{H}$ of *MEC* consists of formulas of the form $p(e_1, e_2)$, $\Box p(e_1, e_2)$ and $\Diamond p(e_1, e_2)$, for every property p and events e_1 and e_2 defined in $\mathcal{H}$. We intend representing in this way the property-labeled interval $p(e_1, e_2)$ as an MVI, a □-MVI and a ◇-MVI, respectively. Clearly, $\mathcal{A}_\mathcal{H} \subseteq \mathcal{B}_\mathcal{H}$.

In order to provide *MEC* with a semantics, we must shift the focus from the current knowledge state w to all knowledge states that are reachable from w, i.e. $\mathrm{Ext}_\mathcal{H}(w)$, and more generally to $W_\mathcal{H}$. Now, by definition, w' is an extension of w if $w \subseteq w'$. Since $\subseteq$ is a reflexive partial order, $(W_\mathcal{H}, \subseteq)$ can be naturally viewed as a finite, reflexive, transitive and antisymmetric modal frame. If we consider this frame together with the straightforward modal extension of the valuation $\upsilon_\mathcal{H}$ to an arbitrary knowledge state, we obtain a modal model for *MEC*.

Definition 3. (MEC intended model)

Let $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$ be an EC-structure. The *MEC-frame* $\mathcal{F}_\mathcal{H}$ of $\mathcal{H}$ is the frame $(W_\mathcal{H}, \subseteq)$. The *intended MEC-model* of $\mathcal{H}$ is the modal model $\mathcal{I}_\mathcal{H} = (W_\mathcal{H}, \subseteq, \upsilon_\mathcal{H})$, where the propositional valuation $\upsilon_\mathcal{H} : W_\mathcal{H} \to 2^{\mathcal{A}_\mathcal{H}}$ is defined as in Definition 2. Given $w \in W_\mathcal{H}$ and $\varphi \in \mathcal{B}_\mathcal{H}$, the truth of φ at w with respect to $\mathcal{I}_\mathcal{H}$, denoted by $\mathcal{I}_\mathcal{H}; w \models \varphi$, is defined as follows:

$\mathcal{I}_\mathcal{H}; w \models p(e_1, e_2)$ iff $p(e_1, e_2) \in \upsilon_\mathcal{H}(w)$ *as from Definition 2*;
$\mathcal{I}_\mathcal{H}; w \models \Box p(e_1, e_2)$ iff $\forall w' \in W_\mathcal{H}$ such that $w \subseteq w'$, $\mathcal{I}_\mathcal{H}; w' \models p(e_1, e_2)$;
$\mathcal{I}_\mathcal{H}; w \models \Diamond p(e_1, e_2)$ iff $\exists w' \in W_\mathcal{H}$ such that $w \subseteq w'$ and $\mathcal{I}_\mathcal{H}; w' \models p(e_1, e_2)$.

A MEC-formula φ is *valid* in $\mathcal{I}_\mathcal{H}$, written $\mathcal{I}_\mathcal{H} \models \varphi$, if $\mathcal{I}_\mathcal{H}; w \models \varphi$ for all $w \in W_\mathcal{H}$. □

We will drop the subscripts $_\mathcal{H}$ whenever this does not lead to ambiguities. Moreover, given a knowledge state w in $W_\mathcal{H}$ and a *MEC*-formula φ over $\mathcal{H}$, we write $w \models \varphi$ for $\mathcal{I}_\mathcal{H}; w \models \varphi$. Similarly, we abbreviate $\mathcal{I}_\mathcal{H} \models \varphi$ as $\models \varphi$.

It is interesting to notice that it would have been equivalent to consider completions instead of extensions in the previous definition. Instead, the two notions are not interchangeable in general in the refinements of *MEC* discussed in the remainder of this section, as shown in [4, 8].

To determine the sets of □- and ◇-MVIs, it is possible to exploit necessary and sufficient *local conditions* over the given partial order, thus avoiding a complete (and expensive) search of all the consistent extensions of the given order. More precisely [2], a property p necessarily holds between two events e_1 and e_2 if and only if the interval (e_1, e_2) belongs to the current order, e_1 initiates p, e_2

terminates p, and no event that either initiates or terminates p (or a property incompatible with p) will ever be consistently located between e_1 and e_2. Similarly, a property p may possibly hold between e_1 and e_2 if and only if the interval (e_1, e_2) is consistent with the current ordering, e_1 initiates p, e_2 terminates p, and there are no already known interrupting events between e_1 and e_2. This is precisely expressed by the following lemma [2].

Lemma 4. (*Local conditions*)

Let $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$ *be a EC-structure. For any pair of events* $e_1, e_2 \in E$*, any property* $p \in P$*, and any* $w \in W_{\mathcal{H}}$*,*

- $\mathcal{I}_{\mathcal{H}}; w \models \Box p(e_1, e_2)$ *if and only if*
  i. $(e_1, e_2) \in w$*;*
  ii. $e_1 \in [p\rangle$*;*
  iii. $e_2 \in \langle p]$*;*
  iv. $nsb(p, e_1, e_2, w)$*, where* $nsb(p, e_1, e_2, w)$ *iff*

$$\neg\exists e \in E.\ (e, e_1) \notin w \quad \wedge \quad e \neq e_1 \quad \wedge \quad (e_2, e) \notin w \quad \wedge \quad e \neq e_2$$
$$\wedge\ \exists q \in P.\, ((e \in [q\rangle\ \vee\ e \in \langle q])\ \wedge\ (]p, q[\ \vee\, p = q))).$$

- $\mathcal{I}_{\mathcal{H}}; w \models \Diamond p(e_1, e_2)$ *if and only if*
  i. $(e_2, e_1) \notin w$*;*
  ii. $e_1 \in [p\rangle$*;*
  iii. $e_2 \in \langle p]$*;*
  iv. $nb(p, e_1, e_2, w)$*.* □

The *Generalized Modal Event Calculus*, *GMEC* [2], broadens the scope of *MEC* by interpreting a necessary MVI $\Box p(e_1, e_2)$ and a possible MVI $\Diamond p(e_1, e_2)$ as the application of the operators $\Box$ and $\Diamond$, respectively, from an appropriate modal logic to the MVI $p(e_1, e_2)$. On the basis of this observation, it extends the language of *MEC* by allowing the combination of property-labeled intervals by means of propositional connectives and modal operators. The query language of *GMEC* is defined as follows.

Definition 5. (GMEC-language)

Let $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$ be an EC-structure. Given the EC-language $\mathcal{A}_{\mathcal{H}}$, the *GMEC-language* of $\mathcal{H}$, denoted $\mathcal{L}_{\mathcal{H}}$, is the modal language with propositional letters in $\mathcal{A}_{\mathcal{H}}$ and logical operators in $\{\neg, \wedge, \vee, \Box, \Diamond\}$. □

Clearly, $\mathcal{B}_{\mathcal{H}} \subseteq \mathcal{L}_{\mathcal{H}}$. Definition 3 can be easily generalized to GMEC as follows.

Definition 6. (GMEC intended model)

Let $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$ be an EC-structure. The *GMEC-frame* $\mathcal{F}_{\mathcal{H}}$ and the *intended GMEC-model* $\mathcal{I}_{\mathcal{H}}$ of $\mathcal{H}$ are defined as in Definition 3. Given $w \in W_{\mathcal{H}}$ and $\varphi \in \mathcal{L}_{\mathcal{H}}$, the truth of φ at w with respect to $\mathcal{I}_{\mathcal{H}}$, denoted by $\mathcal{I}_{\mathcal{H}}; w \models \varphi$, is defined as follows:

$$
\begin{array}{lll}
\mathcal{I}_{\mathcal{H}}; w \models p(e_1, e_2) & \text{iff} & p(e_1, e_2) \in \upsilon_{\mathcal{H}}(w) \;\; \textit{as in Definition 2}; \\
\mathcal{I}_{\mathcal{H}}; w \models \neg\varphi & \text{iff} & \mathcal{I}_{\mathcal{H}}; w \not\models \varphi; \\
\mathcal{I}_{\mathcal{H}}; w \models \varphi_1 \wedge \varphi_2 & \text{iff} & \mathcal{I}_{\mathcal{H}}; w \models \varphi_1 \;\text{ and }\; \mathcal{I}_{\mathcal{H}}; w \models \varphi_2; \\
\mathcal{I}_{\mathcal{H}}; w \models \varphi_1 \vee \varphi_2 & \text{iff} & \mathcal{I}_{\mathcal{H}}; w \models \varphi_1 \;\text{ or }\; \mathcal{I}_{\mathcal{H}}; w \models \varphi_2; \\
\mathcal{I}_{\mathcal{H}}; w \models \Box\varphi & \text{iff} & \forall w' \in W_{\mathcal{H}} \text{ such that } w \subseteq w', \mathcal{I}_{\mathcal{H}}; w' \models \varphi; \\
\mathcal{I}_{\mathcal{H}}; w \models \Diamond\varphi & \text{iff} & \exists w' \in W_{\mathcal{H}} \text{ such that } w \subseteq w' \text{ and } \mathcal{I}_{\mathcal{H}}; w' \models \varphi.
\end{array}
\qquad \Box
$$

The attempt of characterizing *GMEC* within the rich taxonomy of modal logics reveals *Sobocinski logic*, also known as *system K1.1* [14], as its closest relative. Syntactically, this logic extends *S4* for the validity of the formula $\Box(\Box(p \to \Box p) \to p) \to p$, added as a further axiom to the traditional formulation of that system. Semantically, it is characterized by the class of the finite, reflexive, transitive and antisymmetric frames, i.e. by the class of all finite partial orders. The relationship between *GMEC* and *K1.1* is captured by the following theorem, where the validity relation of Sobocinski logic has been indicated as $\models_{K1.1}$.

Theorem 7. (*GMEC and K1.1*)

Each thesis of K1.1 is a valid formula of GMEC, i.e., for each GMEC-formula φ, if $\models_{K1.1} \varphi$, then $\models \varphi$. ■

Since the intended *GMEC*-model $\mathcal{I}_{\mathcal{H}}$ is based on a finite, reflexive, transitive and antisymmetric frame, Theorem 7 immediately follows from the (soundness and) completeness of *K1.1* with respect to the class of all finite partial orders [14]. From the above syntactic characterization of Sobocinski logic, every formula valid in *S4* is valid in *K1.1*. Therefore, Theorem 7 permits lifting to *GMEC* the following well-known equivalences of *S4*.

Corollary 8. (*Some equivalent GMEC-formulas*)

Let φ, φ_1 and φ_2 be GMEC-formulas. Then, for every knowledge state $w \in W$,

- $w \models \Box\neg\varphi$ *iff* $w \models \neg\Diamond\varphi$
- $w \models \Diamond\neg\varphi$ *iff* $w \models \neg\Box\varphi$
- $w \models \Box(\varphi_1 \wedge \varphi_2)$ *iff* $w \models \Box\varphi_1 \wedge \Box\varphi_2$
- $w \models \Diamond(\varphi_1 \vee \varphi_2)$ *iff* $w \models \Diamond\varphi_1 \vee \Diamond\varphi_2$
- $w \models \Box\Box\varphi$ *iff* $w \models \Box\varphi$
- $w \models \Diamond\Diamond\varphi$ *iff* $w \models \Diamond\varphi$
- $w \models \Box\Diamond\Box\Diamond\varphi$ *iff* $w \models \Box\Diamond\varphi$
- $w \models \Diamond\Box\Diamond\Box\varphi$ *iff* $w \models \Diamond\Box\varphi$ ■

Also specific properties of *K1.1* turn out to be useful in order to implement *GMEC*. The following equivalences can be obtained by exploiting the *McKinsey formula*, $\Box\Diamond\varphi \to \Diamond\Box\varphi$, valid in *K1.1* (but not in *S4*).

Corollary 9. (*Further equivalent GMEC-formulas*)

Let φ be a GMEC-formula. Then, for every knowledge state $w \in W$,

- $w \models \Box\Diamond\Box\varphi$ *iff* $w \models \Box\Diamond\varphi$
- $w \models \Diamond\Box\Diamond\varphi$ *iff* $w \models \Diamond\Box\varphi$ ■

An interesting consequence of Corollaries 8 and 9 is that each *GMEC*-formula φ is logically equivalent to a formula of one of the following forms: ψ, $\Box\psi$, $\Diamond\psi$, $\Box\Diamond\psi$, $\Diamond\Box\psi$, where the main operator of ψ is non-modal. In [2], we provided *GMEC* with a sound and complete axiomatization in the language of hereditary Harrop formulas that heavily exploits the above reductions. Unfortunately, there is no way, in general, of reducing formulas of the form $\Box(\varphi_1 \vee \varphi_2)$ and $\Diamond(\varphi_1 \wedge \varphi_2)$ (such a reduction would be quite significant from a computational point of view): we only have that $(\Box\varphi_1 \vee \Box\varphi_2) \rightarrow \Box(\varphi_1 \vee \varphi_2)$ and $\Diamond(\varphi_1 \wedge \varphi_2) \rightarrow (\Diamond\varphi_1 \wedge \Diamond\varphi_2)$. Furthermore, the attempt at overcoming these difficulties by adding to *K.1.1* the axiom $\Box(p \vee q) \rightarrow (\Box p \vee \Box q)$, distributing the $\Box$ operator over $\vee$, or, equivalently, the axiom $(\Diamond p \wedge \Diamond q) \rightarrow \Diamond(p \wedge q)$, has the effect of collapsing *K.1.1* onto the Propositional Calculus as stated by the following (general) theorem [8].

Theorem 10. (*Collapse of Modal Logics onto the Propositional Calculus*)

The addition of the axiom $\Box(p \vee q) \rightarrow (\Box p \vee \Box q)$ to the axiom system of any modal logic over T causes its collapse onto the Propositional Calculus. ■

In the following, we propose two *intermediate* modal event calculi, that lie linguistically between *MEC* and *GMEC*. They are aimed at extending the expressive power of *MEC*, while preserving as much as possible its computational efficiency. Unlike *GMEC*, the proposed calculi only allow a restricted mix of boolean and modal operators. The first calculus, called *ICMEC* (*Modal Event Calculus with Internal Connectives*), can be obtained from *MEC* by replacing atomic formulas (property-labeled intervals) with propositional formulas (boolean combinations of property-labeled intervals), that is, *ICMEC*-formulas are propositional formulas, possibly prefixed by at most one modal operator. The query language of *ICMEC* is defined as follows.

Definition 11. (ICMEC-language)

Let $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$ be an EC-structure. The *ICMEC-language* of $\mathcal{H}$ is the class of formulas $\mathcal{C}_{\mathcal{H}} = \{\varphi, \Diamond\varphi, \Box\varphi : \varphi$ is a boolean combination of formulas over $\mathcal{A}_{\mathcal{H}}\}$. Any element of $\mathcal{C}_{\mathcal{H}}$ is called an ICMEC-formula. □

Clearly, we have $\mathcal{B}_{\mathcal{H}} \subseteq \mathcal{C}_{\mathcal{H}} \subseteq \mathcal{L}_{\mathcal{H}}$. The semantics of *ICMEC* is given as for *GMEC*. The second calculus, called *ECMEC* (*Modal Event Calculus with External Connectives*), extends *MEC* by supporting boolean combinations of *MEC*-formulas. The query language of *ECMEC* is defined as follows.

Definition 12. (ECMEC-language)

Let $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$ be an EC-structure. The *ECMEC-language* of $\mathcal{H}$ is the class of formulas $\mathcal{D}_\mathcal{H} = \{\varphi : \varphi$ is a boolean combination of formulas over $\mathcal{B}_\mathcal{H}\}$. Any element of $\mathcal{D}_\mathcal{H}$ is called an ECMEC-formula. □

Clearly, we have $\mathcal{B}_\mathcal{H} \subseteq \mathcal{D}_\mathcal{H} \subseteq \mathcal{L}_\mathcal{H}$, completing in this way the diagram in Figure 1. Again, the semantics of this modal event calculus is given as in the case of *GMEC*.

We will now consider the expressiveness of the intermediate calculi we just defined by showing how they can be used to encode our case study from Section 2. In Section 5, we will instead analyze the complexity of these calculi.

4 A Formalization of the Application Domain

In this section, we give a formalization of the example presented in Section 2, and use various modal event calculi to draw conclusions about it. The situation depicted in Figure 2 can be represented by the *EC*-structure $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$, whose components are defined as follows:

- $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$;
- $P = \{R_1, R_2, R_3, R_{12}, R_{13}, R_{21}, R_{23}, R_{31}, R_{32}\}$;
- $[R_1\rangle = \{e_1\}$, $[R_2\rangle = \{e_3\}$, $[R_3\rangle = \{e_5\}$, $[R_{12}\rangle = [R_{13}\rangle = \{e_2\}$, $[R_{21}\rangle = [R_{23}\rangle = \{e_4\}$, $[R_{31}\rangle = [R_{32}\rangle = \{e_6\}$;
- $\langle R_1] = \{e_2\}$, $\langle R_2] = \{e_4\}$, $\langle R_3] = \{e_6\}$, $\langle R_{21}] = \langle R_{31}] = \{e_1\}$, $\langle R_{12}] = \langle R_{32}] = \{e_3\}$, $\langle R_{13}] = \langle R_{23}] = \{e_5\}$;
- $]\cdot,\cdot[= \emptyset$.

We have represented transitions as events with the same name, and denoted by R_i the property that register R_i has value 1, for $i = 1, 2, 3$. Furthermore, we have denoted by R_{ij} the fact that the end point of the interval over which R_i assumes value 1 precedes the starting point of the interval over which R_j takes value 1, for $i, j \in \{1, 2, 3\}$ and $i \neq j$. Properties R_{ij} can be exploited to order the time intervals over which the status registers are set to 1.

The partial order of transitions, described in Figure 3, is captured by the following (current) knowledge state:

$$o = \{(e_1, e_2), (e_1, e_3), (e_2, e_4), (e_2, e_5), (e_3, e_4), (e_3, e_5), (e_4, e_6), (e_5, e_6)\}.$$

Consider the formulas

$$\begin{aligned}
\varphi &= R_1(e_1, e_2) \wedge R_{12}(e_2, e_3) \wedge R_2(e_3, e_4) \wedge R_{23}(e_4, e_5) \wedge R_3(e_5, e_6);\\
\psi_1 &= R_1(e_1, e_2) \wedge R_{12}(e_2, e_3) \wedge R_2(e_3, e_4);\\
\psi_2 &= R_2(e_3, e_4) \wedge R_{23}(e_4, e_5) \wedge R_3(e_5, e_6).
\end{aligned}$$

which are in the query language of *ECMEC*, *ICMEC* and *GMEC*. In order to verify that the switch S is not faulty, we must ensure that the registers R_1, R_2 and R_3 display the expected behavior in all refinements of the current knowledge state o. This amounts to proving that the *GMEC*-formula $\Box\varphi$ is true in o. If this is the case, the fault is to be excluded. If we want to determine the existence of at least one extension of o where the registers behave as displayed in Figure 2, we must verify the truth of $\Diamond\varphi$ in o. If this *GMEC*-formula is true, we cannot be sure whether S is faulty or not. Finally, formulas ψ_1 and ψ_2 (which are subformulas of φ), prefixed by the modal operators $\Box$ or $\Diamond$, can be exploited to locally verify the behavior of status registers.

Since we have that $o^+ \models \Diamond\varphi$ and $o^+ \not\models \Box\varphi$, the knowledge contained in o entitles us to assert that the fault is possible but not certain. Moreover, we are not able to localize the fault. In fact, we have that both $o^+ \not\models \Box\psi_1$ (the fault may involve the relative transition of registers R_1 and R_2) and $o^+ \not\models \Box\psi_2$ (the fault may involve the relative transition of R_2 and R_3 as well). Let us denote with o_1 the state of knowledge obtained by adding the pair (e_2, e_3) to o. As in the previous state, we have that $o_1^+ \models \Diamond\varphi$ and $o_1^+ \not\models \Box\varphi$. In this case, however, we are able to localize the possible fault. Since $o_1^+ \models \Box\psi_1$, we can conclude that the fault does not involve the relative transition of registers R_1 and R_2. On the contrary, $o^+ \not\models \Box\psi_2$, and hence the relative transition of registers R_2 and R_3 may be incorrect. Assume now that, unlike the actual situation depicted in Figure 2, we extend o with the pair (e_3, e_2). Let us denote the resulting state with o_2. We have that $o_2^+ \not\models \Diamond\varphi$, that is, the evolution of the values in the registers hints at a fault in switch S. Finally, let us refine o_1 by adding the pair (e_4, e_5), and call o_3 the resulting knowledge state. In this case, we can infer $o_3^+ \models \Box\varphi$, and hence we can conclude that the switch S is certainly not faulty.

The formulas we have used so far belong to the language of both *GMEC* and *ICMEC*. As we will see in Section 5, this is unfortunate since model checking in these languages is intractable. However, the results presented in Section 3 postulate the existence of approximations of these formulas, within the language of *ECMEC*, that have a polynomial validity test. We will use these formulas to analyze the example at hand.

By Corollary 8, $\Box\varphi$ is equiprovable with the *ECMEC*-formula:

$$\varphi' = \Box R_1(e_1, e_2) \wedge \Box R_{12}(e_2, e_3) \wedge \Box R_2(e_3, e_4) \wedge \Box R_{23}(e_2, e_3) \wedge \Box R_3(e_5, e_6).$$

Therefore, we can use *ECMEC* and φ' to establish whether the switch S is fault-free or is possibly defective. For example, since $o_3^+ \models \varphi'$ entails $o_3^+ \models \Box\varphi$, we can exclude the possibility of a misbehavior of S in situation o_3.

The best *ECMEC*-approximation of $\Diamond\varphi$ we can achieve is the formula

$$\varphi'' = \Diamond R_1(e_1, e_2) \wedge \Diamond R_{12}(e_2, e_3) \wedge \Diamond R_2(e_3, e_4) \wedge \Diamond R_{23}(e_2, e_3) \wedge \Diamond R_3(e_5, e_6)$$

which is *not* equivalent to $\Diamond\varphi$. However, we know that for every knowledge state $w \in W_H$, if $w \models \Diamond\varphi$, then $w \models \varphi''$. We can use this fact to draw negative consequences about our example. In particular, we can use φ'' to determine that S is faulty assuming the trace o_2. Indeed, we have that $o_2^+ \not\models \varphi''$, from which it must be the case that $o_2^+ \not\models \Diamond\varphi$. This allows us to conclude that the behavior of S is certainly incorrect.

Finally, in the knowledge state o, both $o^+ \not\models \varphi'$ and $o^+ \models \varphi''$ hold. The former implies $o^+ \not\models \Box\varphi$, and thus a faulty behavior of S cannot be excluded in the current state. Instead, $o^+ \models \varphi''$ neither allows us to conclude that $o^+ \models \Diamond\varphi$ nor that $o^+ \not\models \Diamond\varphi$. In this case, using *ECMEC* we are not able to establish whether the system is certainly faulty or not. The same holds for the knowledge state o_1.

5 Complexity Analysis

This section is dedicated to studying the complexity of the various modal event calculi presented in Section 3. We model our analysis around the satisfiability relation $\models$ given in Definitions 3 and 6, but we also take into account the numerous results that permit improving its computational behavior (in particular, the locality conditions for the computation of $\Box$-MVIs and $\Diamond$-MVIs — Lemma 4). This approach is sensible since these specifications can be directly turned into the clauses of logic programs implementing these calculi [8].

The notion of cost we adopt is as follows: we assume that verifying the truth of the propositions $e \in [p\rangle$, $e \in \langle p]$ and $]p, p'[$ has constant cost $O(1)$, for given event e and properties p and p'. Although possible in principle, it is disadvantageous in practice to implement knowledge states so that the test $(e_1, e_2) \in w$ has constant cost. We instead maintain an acyclic binary relation o on events whose transitive closure o^+ is w (cf. Section 3). Verifying whether $(e_1, e_2) \in w$ holds becomes a reachability problem in o and it can be solved in quadratic time $O(n^2)$ in the number n of events [6].

Given an *EC*-structure $\mathcal{H}$, a knowledge state $w \in W_{\mathcal{H}}$ and a formula φ relatively to any of the modal event calculi presented in Section 3, we want to characterize the complexity of the problem of establishing whether $\mathcal{I}_{\mathcal{H}}; w \models \varphi$ is true (which is an instance of the general problem of model checking). We call the triple $(\mathcal{H}, w, \varphi)$ an *instance* and generally prefix this term with the name of the calculus we are considering. In the following, we will show that, given an instance $(\mathcal{H}, w, \varphi)$, model checking for φ in the intended model $\mathcal{I}_{\mathcal{H}}$ is polynomial in *EC*, *MEC* and *ECMEC*, while it is NP-hard in *ICMEC* and *GMEC*. The reason for such a different computational behavior for the various modal event calculi is that *MEC* and *ECMEC* can exploit local conditions for testing (boolean combinations of) atomic formulas, possibly prefixed by a modal operator, while the latter two cannot avoid of explicitly searching the whole set of extensions of the given partial ordering, whose number is, in general, exponential with respect to the number of events.

Given an *EC*-instance $(\mathcal{H}, w, \varphi)$, the cost of the test $w \models \varphi$ can be derived to be $O(n^3)$ directly from the relevant parts of Definition 2, as proved in [6]. Exploiting the local conditions yields an identical bound in the case of *MEC*:

Theorem 13. (*Complexity of model checking in MEC*)
Given a MEC-instance $(\mathcal{H}, w, \varphi)$, the test $w \models \varphi$ has cost $O(n^3)$. ∎

Unlike *EC* and *MEC*, where the input formula φ is an atomic formula, possibly prefixed by a modal operator in *MEC*, φ can be arbitrarily large in the case of the remaining calculi in the hierarchy in Figure 1. As a consequence, the dimension of the input formula, that does not come into play in the complexity analysis of *EC* and *MEC*, becomes a relevant parameter for the analysis of the cost of the remaining calculi. Thus, their complexity will be measured in terms of both the dimension k of the input formula (the number of occurrences of atomic formulas it includes) and the size n of the input structure (the number of events in E).

An *ECMEC*-formula φ is the boolean combination of a number of *MEC*-formulas. If φ contains k atomic formulas, this number is precisely k. Therefore, by virtue of Definition 6, the test for φ can be reduced to the resolution of k *MEC* problems. Thus, *ECMEC* has polynomial complexity.

Theorem 14. (*Complexity of model checking in ECMEC*)
Given an ECMEC-instance $(\mathcal{H}, w, \varphi)$*, the test* $w \models \varphi$ *has cost* $O(kn^3)$. ■

The placement of the modal operators in *ICMEC* prevents us, in general, from being able to use local conditions in tests. An exhaustive exploration of the extensions of the current knowledge state is unavoidable. This raises the complexity of the problem beyond tractability, as expressed by the following theorem.

Theorem 15. (*Complexity of model checking in ICMEC*)
Given an ICMEC-instance $(\mathcal{H}, w, \varphi)$*, the test* $w \models \varphi$ *is NP-hard.*

Proof.
If φ is a propositional formula, then model checking reduces to verifying whether it evaluates to true with respect to the current state of knowledge. It has a polynomial cost. The remaining cases are $\varphi = \Diamond\psi$ and $\varphi = \Box\psi$.

We first prove that if $\varphi = \Diamond\psi$, then model checking in *ICMEC* is NP-complete. It is easy to see that this problem belongs to NP. Indeed, in order to establish whether $w \models \Diamond\psi$ holds, we non-deterministically generate extensions w' of w, and then test the truth of ψ in w' until an extension where ψ holds is found. There are exponentially many such extensions. Since the formula ψ does not include any modal operator, the test in each extension is polynomial. In order to prove that the considered problem is NP-hard, we define a (polynomial) reduction of 3SAT [9] into *ICMEC*.

Let q be a boolean formula in 3CNF, $p_1, p_2, .., p_n$ be the propositional variables that occur in q, and $q = c_1 \wedge c_2 \wedge \ldots \wedge c_m$, where $c_i = l_{i,1} \vee l_{i,2} \vee l_{i,3}$ and for each i, j either $l_{i,j} = p_k$ or $l_{i,j} = \neg p_k$ for some k.

Let us define an *EC*-structure $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$ such that:

$E = \{e(p_i), e(\neg p_i) : 1 \leq i \leq n\}$;
$P = \{p_i : 1 \leq i \leq n\}$;
$[p_i\rangle : \{e(p_i)\}$ and $\langle p_i] : \{e(\neg p_i)\}$, for $1 \leq i \leq n$;
$]\cdot,\cdot[= \emptyset$.

Moreover, let $w = \emptyset$ and $\psi = c'_1 \wedge c'_2 \wedge \ldots \wedge c'_m$, where $c'_i = l'_{i,1} \vee l'_{i,2} \vee l'_{i,3}$, and for each i, j, if $l_{i,j} = p_k$, then $l'_{i,j} = p_k(e(p_k), e(\neg p_k))$, otherwise ($l_{i,j} = \neg p_k$) $l'_{i,j} = \neg p_k(e(p_k), e(\neg p_k))$ It is not difficult to see that $w \models \Diamond\psi$ if and only if q is satisfiable.

Let us show now that if $\varphi = \Box\psi$, then model checking in *ICMEC* is NP-hard. We prove this result by defining a (polynomial) reduction of the problem of propositional validity into *ICMEC*.

Let q be a boolean formula in 3DNF, $p_1, p_2, .., p_n$ be the propositional variables that occur in q, and $q = d_1 \vee d_2 \vee \ldots \vee d_m$, where $d_i = l_{i,1} \wedge l_{i,2} \wedge l_{i,3}$ and for each i, j, either $l_{i,j} = p_k$ or $l_{i,j} = \neg p_k$. We define the *EC*-structure $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$ as in the previous subcase. Let $w = \emptyset$ and $\psi = d'_1 \vee d'_2 \vee \ldots \vee d'_m$, where $d'_i = l'_{i,1} \wedge l'_{i,2} \wedge l'_{i,3}$, and for each i, j, if $l_{i,j} = p_k$, then $l'_{i,j} = p_k(e(p_k), e(\neg p_k))$, otherwise ($l_{i,j} = \neg p_k$) $l'_{i,j} = \neg p_k(e(p_k), e(\neg p_k))$. It is straightforward to see that $w \models \Box\psi$ if and only if q is valid in propositional logic. ■

Since *ICMEC* is a linguistic fragment of *GMEC*, Theorem 15 allows us to conclude that model checking for *GMEC* is NP-hard too.

Corollary 16. (*Complexity of model checking in GMEC*)
Given a GMEC-instance $(\mathcal{H}, w, \varphi)$*, the test* $w \models \varphi$ *is NP-hard.* ■

We summarize the results obtained in this section in the following table.

Calculus	*EC*	*MEC*	*ECMEC*	*ICMEC*	*GMEC*
Parameters	n events	n events	n events k atomic formulas	n events k atomic formulas	n events k atomic formulas
Model checking	$O(n^3)$	$O(n^3)$	$O(kn^3)$	NP-hard	NP-hard

6 Conclusions and Further Developments

In this paper, we have established a hierarchy of modal event calculi by investigating *ECMEC* and *ICMEC* as intermediate languages between the modal event calculus *MEC* [1] and the generalized modal event calculus *GMEC* [2]. In particular, we showed that *ECMEC* retains enough of the expressive power of *GMEC* while admitting an efficient polynomial implementation in the style of *MEC*. We supported our claims by showing the formalization of an example from the applicative domain of fault diagnosis. Moreover, we gave a rigorous analysis of the complexity of the modal event calculi we considered.

We are developing the proposed framework in several directions. First, the complexity results given in Section 5 can actually be improved. In the proof of Theorem 15, we showed that checking *ICMEC*-formulas of the form $\Diamond\psi$ is NP-complete. Since $\Box = \neg\Diamond\neg$, it easily follows that checking $\Box\psi$ formulas is co-NP(-complete). Thus, the whole problem of testing $w \models \varphi$ (for *ICMEC*) involves either a polynomial check, or an NP-check, or a co-NP check. This means that it can be computed by a Turing machine which can acces an NP-oracle and runs in deterministic polynomial time, and hence the problem is in P^{NP} (since only one call to the oracle is needed, it is actually in $P^{NP[1]}$) [15]. We are currently working at the characterization of the exact complexity of model checking in both *ICMEC* and *GMEC*.

Another issue of interest when working with *EC* and in its modal refinements is the *generation of MVIs*, which can be solved using the same logic programs that implement model checking. In this problem, we replace some, possibly all, events in a formula φ by logical variables and ask which instantiations of these variables make φ true. The problem of MVI generation can still be viewed as a problem of model checking in modal event calculi extended with limited forms of existential quantification. Let *QEC*, *QMEC*, *ECQMEC*, *ICQMEC*, and *GQMEC* respectively be the quantified counterparts of *EC*, *MEC*, *ECMEC*, *ICMEC*, and *GMEC*. It is possible to show that model checking for all quantified modal event calculi essentially lies in the same complexity class of model checking for their unquantified counterparts, except for *ECMEC*, for which the addition of quantification makes the problem NP-hard.

Finally, we are considering the effects of the addition of preconditions to our framework. This step would enlarge the range of applicability of the modal event calculi. However, as proved in [7], an indiscriminated use of preconditions immediately makes the problem NP-hard. Nevertheless, we believe that a formal study of various modal event calculi with preconditions can shed some light on the dynamics of preconditions, and possibly lead to polynomial approximations of the computation of MVIs. Preliminary results in this direction can be found in [4].

Acknowledgments

We would like to thank the reviewers for their useful comments. The first author was partially supported by NFS grant CCR-9303383 and by a scholarship for specialization overseas from the University of Udine. The work of the third author was partially supported by the CNR project *Ambienti e strumenti per la gestione di informazioni temporali.*

References

1. I. Cervesato, L. Chittaro, A. Montanari. "A Modal Calculus of Partially Ordered Events in a Logic Programming Framework", *Proc. of ICLP'95: 12th Int. Conference on Logic Programming*, Kanegawa, Japan, MIT Press, 1995, 299–313.
2. I. Cervesato, A. Montanari. "A General Modal Framework for the Event Calculus and its Skeptical and Credulous Variants (extended and revised version of A. Montanari, L. Chittaro, and I. Cervesato, "A General Modal Framework for the Event Calculus and its Skeptical and Credulous Variants", *Proc. of ECAI'96: 12th European Conference on Artificial Intelligence*, W. Wahlster (ed.), John Wiley & Sons, 1996, 33–37)", *submitted for publication*, July 1996.

3. I. Cervesato, A. Montanari, A. Provetti. “On the Non-Monotonic Behavior of Event Calculus for Deriving Maximal Time Intervals”, *Interval Computations*, 3(2), 1993, 83–119.
4. I. Cervesato, M. Franceschet, A. Montanari. “Modal Event Calculi with Preconditions”, *Proc. of TIME'97: 4th Int. Workshop on Temporal Representation and Reasoning*, L. Khatib, R. Morris (eds.), IEEE Computer Society Press, 1997, 38–45.
5. L. Chittaro, A. Montanari, A. Provetti. “Skeptical and Credulous Event Calculi for Supporting Modal Queries”, *Proc. of ECAI'94: 11th European Conference on Artificial Intelligence*, A. Cohn (ed.), John Wiley & Sons, 1994, 361–365.
6. L. Chittaro, A. Montanari, I. Cervesato. “Speeding up temporal reasoning by exploiting the notion of kernel of an ordering relation”, *Proc. of TIME'95: 2nd Int. Workshop on Temporal Representation and Reasoning*, S. Goodwin, H. Hamilton (eds.), University of Regina, Canada, 1995, 73–80.
7. T. Dean, M. Boddy. “Reasoning about partially ordered events”, *Artificial Intelligence*, 36, 1988, 375–399.
8. M. Franceschet. *Una Gerarchia di Calcoli Modali degli Eventi: Espressività e Complessità* (in Italian), Tesi di Laurea in Scienze dell'Informazione, Università di Udine, Italy, 1996 (to appear as a Research Report in English).
9. M.R. Garey and D.S. Johnson. *Computing and Intractability: A Guide to the Theory of NP-Completeness*, Freeman & Cie, 1979.
10. J. Harland. *On Hereditary Harrop Formulae as a basis for Logic Programming.* PhD thesis, University of Edinburgh, UK, 1991.
11. R. Kowalski, M. Sergot. “A Logic-based Calculus of Events”, *New Generation Computing*, 4, Ohmsha Ltd and Springer-Verlag, 1986, 67–95.
12. D.C. Moffat, G.D. Ritchie. “Modal Queries about Partially-ordered Plans”, *Journal of Expt. Theor. Artificial Intelligence*, 2, 1990, 341–368.
13. K. Nökel. *Temporarily Distributed Symptoms in Technical Diagnosis*, Springer-Verlag, 1991.
14. K. Segerberg. *An Essay in Classical Modal Logic.* Uppsala Filosofiska Studier, 1971.
15. L. Stockmeyer. “Classifying the computational complexity of problems”, *Journal of Symbolic Logic*, 52(1), 1987, 1–43.

This article was processed using the LaTeX macro package with LLNCS style