A General Modal Framework for the Event Calculus and its Skeptical and Credulous Variants

Iliano Cervesato[1] and **Angelo Montanari**[2] and **Luca Chittaro**[3]

Abstract. We propose a general and uniform modal framework for the Event Calculus (EC) and its skeptical and credolous variants. The resulting temporal formalism, called the Generalized Modal Event Calculus (GMEC), extends considerably the expressive power of EC when information about the ordering of events is incomplete. It provides means of inquiring about the evolution of the maximal validity intervals of properties relatively to all possible refinements of the ordering data by allowing free mixing of propositional connectives and modal operators. We first give a semantic definition of GMEC; then, we propose a declarative encoding of GMEC in the language of hereditary Harrop formulas and state the soundness and completeness of the resulting logic program.

1 INTRODUCTION

This paper proposes a general and uniform modal framework for Kowalski and Sergot's Event Calculus (EC) [9] and its skeptical (SKEC) and credulous (CREC) variants [2, 4, 5]. In general, given a set of event occurrences, EC allows one to derive maximal validity intervals (MVIs hereinafter) over which properties initiated or terminated by those events hold. Most approaches based on EC assume the occurrence time of each event to be known; here, we explore the case of partially ordered events devoid of an explicit occurrence time.

The problem of computing which facts must be or may possibly be true over certain time intervals in presence of partially ordered events has been already addressed in the literature (e.g. [2, 4, 5, 6, 7, 11]) and case studies in the domains of diagnosis and planning have been analyzed in [5] and [11], respectively. In particular, Dean and Boddy [6] showed that this computation is intractable in the general case and propose polynomial approximations that compute either a subset of necessary facts or a superset of possible ones. In [4] and [5], we propose two variants of EC, called Skeptical EC (SKEC) and Credulous EC (CREC), which respectively compute (a subset of) the necessarily true MVIs and (a superset of) the possibly true MVIs. Consider the following example [1]. Take two properties p and q which are respectively initiated by events e_1 and e_2, provided that precondition r holds. Moreover, suppose that both events terminate r. Consider a scenario where (i) both e_1 and e_2 occurred, but their ordering is unknown, and (ii) an event e_0, occurred before e_1 and e_2, initiated r. A precise temporal reasoner should conclude that either p or q must hold. On the contrary, CREC concludes that *both* p *and* q may hold, while SKEC concludes that *neither* p *nor* q necessary hold. However, it is possible to identify suitable hypotheses under which SKEC and CREC precisely compute the sets of necessarily true and possibly true MVIs, respectively. This is the case, for instance, when boolean connectives and preconditions are not used. In [2], we defined a uniform modal interpretation for EC, SKEC and CREC, called Modal Event Calculus (MEC). For the sake of simplicity, we restricted ourselves to the case of events devoid of preconditions. MEC encompasses both atomic formulas (MVIs computed by EC) and simply modalized atomic formulas, i.e. atomic formulas prefixed by only one modality (MVIs computed by SKEC and CREC). It is provided with a sound and complete axiomatic formulation in a logic programming framework.

In this paper, a different approach is taken (a complete account of the work is given in [3]). We initially give a *semantic* formulation of EC and extend it to a modal interpretation by taking into account all possible evolutions of the ordering data. Unlike MEC, the resulting formalism, called the Generalized Modal Event Calculus (GMEC), allows free mixing of propositional connectives and modal operators. Such a capability is essential to deal with real-world applications, as pointed out in [5]. Then, we provide GMEC with a sound and complete axiomatization in the language of hereditary Harrop formulas.

The paper is organized as follows. Section 2 first recalls some basic definitions about orderings and tailors them to the needs of the subsequent discussion; then, it formally defines GMEC and presents its fundamental properties. Section 3 summarizes the definition and operational semantics of hereditary Harrop formulas and uses this language to give a sound and complete encoding of GMEC. The conclusions provide an assessment of the work done and discuss future developments.

2 THE GENERALIZED MODAL EC

In this section, we formally define the Generalized Modal Event Calculus (GMEC). We consider the case in which the set of event occurrences has been fixed once and for all and the input process consists in the addition of information about the relative ordering of event pairs. Furthermore, we assume that events do not happen simultaneously and that the ordering information is always consistent. We call *knowledge state* a partial specification of the ordering.

The section is organized as follows. We first recall some useful notions about ordering relations. Then, we provide EC with a semantic interpretation that validates, in the current knowledge state, precisely the MVIs computed by EC. By considering all possible knowledge states with the associated reachability relation, this model is naturally lifted to a modal interpretation. The corresponding extension of EC with propositional connectives and modalities substantially augments the expressive power of EC.

[1] Department of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue – Pittsburgh, PA 15213-3891. Email: iliano@cs.cmu.edu
[2] Dipartimento di Matematica e Informatica, Università di Udine, Via delle Scienze, 206 – 33100 Udine, Italy. Email: montana@dimi.uniud.it.
[3] Dipartimento di Matematica e Informatica, Università di Udine, Via delle Scienze, 206 – 33100 Udine, Italy. Email: chittaro@dimi.uniud.it.

ECAI 96. *12th European Conference on Artificial Intelligence*
Edited by W. Wahlster
Published in 1996 by John Wiley & Sons, Ltd.

2.1 On ordering relations

In the following, we will rely upon different notions of ordering and ordered set. The ordering information, as usually *represented* in EC, constitutes a quasi-order, i.e. an ordering relation missing some transitive links; however, this information is *used* in EC as a strict order. Moreover, the structure representing the possible evolutions of the ordering data constitutes a non-strict order.

Definition 1 (*Quasi-orders, strict orders, non-strict orders*)
Let E be a set and R a binary relation on E. R is called a quasi-order *if it is acyclic; a* strict order *if it is irreflexive, asymmetric and transitive; a* non-strict order *if it is reflexive, antisymmetric and transitive. The structure (E, R) is respectively called a* quasi-ordered set, *a* strictly ordered set *and a* non-strictly ordered set. □

We denote the sets of all quasi-orders and of all strict orders on E as O_E and W_E, respectively. It is easy to show that, for any set E, $W_E \subseteq O_E$ (actually, $W_E \subset O_E$ if E has at least three elements). We will use the letters o and w possibly subscripted to denote quasi-orders and strict orders, respectively. We indicate the transitive closure of a relation R as R^+. Clearly, if (E, R) is a quasi-ordered set, then (E, R^+) is a strictly ordered set. Two quasi-orders $o_1, o_2 \in O_E$ are *equally informative* if $o_1^+ = o_2^+$. This induces an equivalence relation $\sim$ on O_E. It is easy to prove that, for any set E, $O_{E/\sim}$ and W_E are isomorphic. In the following, we will often identify a quasi-order o with the corresponding element o^+ of W_E.

The set $\mathbf{2}^{E\times E}$ of all binary relations on E naturally becomes a non-strictly ordered set when considered together with the usual subset relation $\subseteq$. Moreover, $(\mathbf{2}^{E\times E}, \cup, \cap, ^{-}, E\times E, \emptyset)$ is a boolean lattice. Being W_E a subset of $\mathbf{2}^{E\times E}$, the restriction of $\subseteq$ to this set still forms a non-strict order. Indeed, we have that, for any set E, $(W_E, \subseteq)$ is a non-strictly ordered set. It can be easily proved that $(W_E, \cap, \emptyset)$ forms a lower semi-lattice. Moreover, for any $w_1, w_2 \in W_E$, the relation $w_1 \uparrow w_2 = (w_1 \cup w_2)^+$ is the least upper bound (*lub*) of w_1 and w_2 whenever this element belongs to W_E. Note that $w_1 \uparrow w_2 \notin W_E$ if w_1 and w_2 contain symmetric pairs. Given w in W_E, any $w' \in W_E$ such that $w \subseteq w'$ is called an *extension* of w. We denote the set of all extensions of w as $Ext(w)$. We have that for any $w \in W_E$, if $(e_1, e_2) \in w$, then $\forall w' \in Ext(w)$, $(e_1, e_2) \in w'$. For any $w \in W_E$, $Ext(w)$ enjoys the same properties as W_E. More precisely, $(Ext(w), \subseteq)$ is a non-strictly ordered set, $(Ext(w), \cap, w)$ is a lower semi-lattice, and $\uparrow$ characterizes the partial operation of *lub* over this semi-lattice. Notice in particular that $Ext(\emptyset) = W_E$.

We conclude the treatment of orderings by giving some definitions related to the notion of interval. Let E be a set and $w \in W_E$. A pair $(e_1, e_2) \in w$ is called an *interval* of w. Given two distinct intervals (e_1, e_2) and (e_1', e_2') over w, we say that (e_1, e_2) is a *subinterval* of (e_1', e_2') (or (e_1', e_2') is a *superinterval* of (e_1, e_2)) with respect to w if either $e_1 = e_1'$ or $(e_1', e_1) \in w$ and dually $e_2 = e_2'$ or $(e_2, e_2') \in w$. We write in this case $(e_1, e_2) \sqsubset_w (e_1', e_2')$. We have that, for any ordering $w \in W_E$, $(w, \sqsubset_w)$ is a strictly ordered set.

2.2 Formalization of GMEC

EC proposes a general approach to representing and reasoning about events and their effects in a logic programming framework. It defines a model of change in which *events* happen at time-points and initiate and/or terminate MVIs over which some *property* holds. EC also embodies a notion of default persistence according to which properties are assumed to persist until an event that interrupts them occurs. Its basic constituents can be formally defined as follows.

Definition 2 (*GMEC-structure*)
A structure *for the* Generalized Modal Event Calculus *(hereinafter* GMEC-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.* □

Since we consider situations where events are ordered relatively to one another, we will represent an MVI for a property p as a pair of events (e_i, e_t) that initiate and terminate p, respectively. MVIs are thus intervals labeled by properties. Let us adopt the set of all property-labeled intervals as the language of EC. The task performed by EC reduces to deciding which formulas are MVIs and which are not. GMEC extends this language by allowing combinations of property-labeled intervals by means of propositional connectives and modal operators.

Definition 3 (*GMEC-language*)
Let $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$ be a GMEC-structure. The base language *of $\mathcal{H}$ is the set of propositional letters $\mathcal{A}_\mathcal{H} = \{p(e_1, e_2) : p \in P$ and $e_1, e_2 \in E\}$. 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 $\{\neg, \wedge, \vee, \Box, \Diamond\}$. We refer to the elements of $\mathcal{A}_\mathcal{H}$ and $\mathcal{L}_\mathcal{H}$ as* atomic formulas *and* GMEC-formulas, *respectively.* □

Notice that, beyond the structured notation we use for atomic formulas, $\mathcal{L}_\mathcal{H}$ is a propositional modal language.

Standard implementations of EC represent knowledge states as quasi-orders, and take their transitive closure in order to make inferences concerning MVIs. Therefore, given a GMEC-structure $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$, we need to interpret atomic formulas relatively to the elements of W_E (denoted $W_\mathcal{H}$ in this context) associated with the current state of knowledge w. The semantics of EC is defined by the (propositional) valuation $v_\mathcal{H}^w$, which discriminates MVIs from other intervals in w.

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 (iii), (i) and (ii), 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 $p(e_1, e_2)$.

GMEC expands the scope of EC by shifting the focus from the current knowledge state, w, to all knowledge states that are reachable from w. By definition, w' is an extension of w if $w \subseteq w'$. Since $\subseteq$ is a non-strict order, $(W_\mathcal{H}, \subseteq)$ can be naturally viewed as a finite, reflexive and transitive modal frame. If we consider this frame together with the straightforward modal extension of the valuation $v_\mathcal{H}^w$ to an arbitrary knowledge state, we obtain a modal model for GMEC.

Definition 4 (*GMEC-model*)
Let $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$ be a GMEC-structure. We denote as $O_\mathcal{H}$ and $W_\mathcal{H}$ the set O_E of quasi-orders and the set W_E of

strict orders over E, respectively. We call the elements of $O_{\mathcal{H}}$ (and consequently of $W_{\mathcal{H}}$) knowledge states. *The* GMEC-frame $\mathcal{F}_{\mathcal{H}}$ *of* $\mathcal{H}$ *is the frame* $(W_{\mathcal{H}}, \subseteq)$. *The* intended GMEC-model *of* $\mathcal{H}$ *is the modal model* $\mathcal{I}_{\mathcal{H}} = (W_{\mathcal{H}}, \subseteq, \upsilon_{\mathcal{H}})$, *where the valuation* $\upsilon_{\mathcal{H}} \subseteq W_{\mathcal{H}} \times \mathcal{A}_{\mathcal{H}}$ *is defined in such a way that* $(w, p(e_1, e_2)) \in \upsilon_{\mathcal{H}}$ *if and only if*

(i) $e_1 \in [p\rangle$; *(ii)* $e_2 \in \langle p]$; *(iii)* $(e_1, e_2) \in w$;
(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 satisfiability relation is defined as follows:

$\mathcal{I}_{\mathcal{H}}; w \models p(e_1, e_2)$ *iff* $(w, p(e_1, e_2)) \in \upsilon_{\mathcal{H}}$;
$\mathcal{I}_{\mathcal{H}}; w \models \neg\varphi$ *iff* $\mathcal{I}_{\mathcal{H}}; w \not\models \varphi$;
$\mathcal{I}_{\mathcal{H}}; w \models \varphi_1 \wedge \varphi_2$ *iff* $\mathcal{I}_{\mathcal{H}}; w \models \varphi_1$ *and* $\mathcal{I}_{\mathcal{H}}; w \models \varphi_2$;
$\mathcal{I}_{\mathcal{H}}; w \models \varphi_1 \vee \varphi_2$ *iff* $\mathcal{I}_{\mathcal{H}}; w \models \varphi_1$ *or* $\mathcal{I}_{\mathcal{H}}; w \models \varphi_2$;
$\mathcal{I}_{\mathcal{H}}; w \models \Box\varphi$ *iff* $\forall w' \in W_{\mathcal{H}}\ w \subseteq w' \Rightarrow \mathcal{I}_{\mathcal{H}}; w' \models \varphi$;
$\mathcal{I}_{\mathcal{H}}; w \models \Diamond\varphi$ *iff* $\exists w' \in W_{\mathcal{H}}\ w \subseteq w' \wedge \mathcal{I}_{\mathcal{H}}; w' \models \varphi$. □

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 GMEC-formula φ over $\mathcal{H}$, we write $w \models \varphi$ for $\mathcal{I}_{\mathcal{H}}; w \models \varphi$.

Let us now state a number of results concerning the adequacy of the definition of GMEC-structure with respect to the informal concept of MVI introduced in [9], and the modal extensions defined in [2, 4, 5]. We have already shown that a satisfiable atomic formula $p(e_1, e_2)$ identifies an interval during which the property p holds. These intervals are maximal and uninterrupted, i.e. p does not hold on any superinterval or subinterval of (e_1, e_2):

Property 5 (*Satisfiable atomic formulas are MVIs*)
Let $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$ *be a GMEC-structure and* $w \in W$ *such that* $w \models p(e_1, e_2)$. *Then* $\forall e'_1, e'_2 \in E$,

a. if $(e'_1, e'_2) \sqsubset_w (e_1, e_2)$, *then* $w \not\models p(e'_1, e'_2)$;
b. if $(e_1, e_2) \sqsubset_w (e'_1, e'_2)$, *then* $w \not\models p(e'_1, e'_2)$. ■

The sets of MVIs that are necessarily and possibly valid in the current state of knowledge w correspond respectively to the $\Box$- and $\Diamond$-moded atomic formulas which are valid in w. We define the sets $MVI(w)$, $\Box MVI(w)$ and $\Diamond MVI(w)$ of MVIs, necessary MVIs and possible MVIs with respect to w, respectively, as follows:

$$MVI(w) = \{p(e_1, e_2) : w \models p(e_1, e_2)\};$$
$$\Box MVI(w) = \{p(e_1, e_2) : w \models \Box p(e_1, e_2)\};$$
$$\Diamond MVI(w) = \{p(e_1, e_2) : w \models \Diamond p(e_1, e_2)\}.$$

In the following, it will be useful to view these sets as functions $MVI(\cdot)$, $\Box MVI(\cdot)$ and $\Diamond MVI(\cdot)$ of the knowledge state w.

It is possible to prove that: (i) $\Box MVI(w)$ will persist whatever the evolution of the ordering information will be; (ii) each element in $\Diamond MVI(w)$ is valid in at least one extension of w; (iii) $\Box MVI(w)$, $MVI(w)$, and $\Diamond MVI(w)$, where w is the current state of knowledge, form an inclusion chain; (iv) $\Box MVI(\cdot)$ monotonically grows as the current ordering information is completed; (v) $\Diamond MVI(\cdot)$ shrinks monotonically as we acquire more ordering information and a smaller number of future states is viable. These properties can be formalized as follows:

Property 6 *Let* $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$ *be a GMEC-structure and* $w \in W$, *then*

(i) if $p(e_1, e_2) \in \Box MVI(w)$, *then* $\forall w' \in Ext(w)$, $p(e_1, e_2) \in MVI(w')$;
(ii) if $p(e_1, e_2) \in \Diamond MVI(w)$, *then* $\exists w' \in Ext(w)$, $p(e_1, e_2) \in MVI(w')$;
(iii) $\Box MVI(w) \subseteq MVI(w) \subseteq \Diamond MVI(W)$;
(iv) if $w \subseteq w'$ *then* $\Box MVI(w) \subseteq \Box MVI(w')$;
(v) if $w \subseteq w'$ *then* $\Diamond MVI(w') \subseteq \Diamond MVI(w)$. ■

The following lemma shows how the satisfiability test for an arbitrary GMEC-formula having a modality as its main connective can be reduced to first testing the satisfiability of its immediate subformula in the current world and then checking the satisfiability of the original formula in the 'one-step' extensions of the current knowledge state. This result stands as the basis of the treatment of the modal operators in section 3.

Lemma 7 (*Unfolding modalities*)
Let $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$ *be a GMEC-structure,* $\varphi \in \mathcal{L}_{\mathcal{H}}$ *a GMEC-formula over* $\mathcal{H}$, *and* $w \in W$. *Then*

a. $w \models \Box\varphi$ *iff* $w \models \varphi$ *and* $\forall(e_1, e_2)$ *such that* $(e_1, e_2), (e_2, e_1) \notin w$, $w \uparrow \{(e_1, e_2)\} \models \Box\varphi$;
b. $w \models \Diamond\varphi$ *iff* $w \models \varphi$ *or* $\exists(e_1, e_2)$ *such that* $(e_1, e_2), (e_2, e_1) \notin w$, $w \uparrow \{(e_1, e_2)\} \models \Diamond\varphi$. ■

Lemma 7 allows one to prove interesting properties of GMEC-models. As an example, it is possible to show that GMEC-models validate the so-called McKinsey formula $\Box\Diamond\phi \rightarrow \Diamond\Box\phi$. Consider a GMEC-model $\mathcal{I}_{\mathcal{H}}$ and a world $w \in W_{\mathcal{H}}$ such that $w \models \Box\Diamond\phi$. By Lemma 7, we have that $w \models \Diamond\phi$ and for every (e_1, e_2) such that $(e_1, e_2), (e_2, e_1) \notin w$, $w \uparrow \{(e_1, e_2)\} \models \Box\Diamond\phi$. By recursively applying such an argument, we have that for all w' such that $w \subseteq w'$, $w' \models \Diamond\phi$. Since, by Definition 2, the set of events is finite, at last we arrive at a world w_f in which for every pair of events (e_1, e_2), it is either $(e_1, e_2) \in w_f$ or $(e_2, e_1) \in w_f$ and $w_f \models \Diamond\phi$. Here we can apply again Lemma 7 ($\Diamond$ part) to conclude that $w_f \models \phi$ or there exists (e_1, e_2) such that $(e_1, e_2), (e_2, e_1) \notin w$ and $w_f \uparrow \{(e_1, e_2)\} \models \Diamond\phi$. However, since w_f is final, we have that $w_f \models \phi$. Another application of Lemma 7 ($\Box$ part) yields $w_f \models \Box\phi$. Then, another application of it ($\Diamond$ part) leads to $w_f \models \Diamond\Box\phi$. We can then go back to w by using Lemma 7 ($\Diamond$ part), e.g. if $w_f = w^* \uparrow \{(e_1, e_2)\}$ for some (e_1, e_2) and $w_f \models \Diamond\Box\phi$, then we have that $w^* \models \Diamond\Box\phi$, and so on.

3 IMPLEMENTATION OF GMEC

In this section, we present a logic programming implementation of GMEC in the language of hereditary Harrop formulas and claim its soundness and completeness with respect to the GMEC semantics presented in Section 2. In Section 3.1, we first briefly recall the definition of hereditary Harrop formulas (HH-formulas for short) and their operational semantics as a logic programming language. In Section 3.2, we define an encoding of GMEC-structures, orderings and GMEC-formulas as HH-formulas. We also describe a program modeling the validity relation over GMEC. Section 3.3 shows the soundness and completeness of this program wth respect to the notion of GMEC-model.

3.1 Hereditary Harrop formulas

So far, the implementation language for EC has always been the language of Horn clauses augmented with negation-as-failure, which constitutes the core of the logic programming language Prolog. This traditional Prolog implementation can be easily extended to cover the propositional connectives. In [2], we showed how a restriction of the purely modal extension of EC can be encoded in Prolog. However, when arbitrarily mixing propositional connectives and modalities, as in GMEC, a direct encoding in Prolog becomes unsatisfactory: the resulting program is either highly non-declarative (for the necessary presence of a large number of `assert` and `retract` statements), or extremely inelegant (as we experienced in [5]).

In order to implement GMEC, we chose the language of First-Order hereditary Harrop formulas [10] augmented with negation-as-failure. Hereditary Harrop formulas extend Horn clauses by allowing the presence of implication and universal quantification in goal formulas. The former feature will give us a declarative means of temporarily augmenting the program with new facts and perform in this manner a form of hypothetical reasoning. It will play an essential role in the implementation of GMEC, because it allows one to emulate the required behavior of `assert` and `retract` in a purely declarative fashion. A complete treatment of the operational semantics of the language of HH-formulas is given in [10].

Given a HH-program $\mathcal{P}$ and a HH-goal G, we express the fact that G is derivable from $\mathcal{P}$ as $\mathcal{P} \vdash_{HH} G$. An implication goal D `=>` G is provable in $\mathcal{P}$ if G is provable in $\mathcal{P}$ augmented with the clause(s) D. In symbols, $\mathcal{P} \vdash_{HH} D$ `=>` G iff $\mathcal{P}, D \vdash_{HH} G$. For convenience, we use a curried notation for terms and atoms, writing for example `(f a (h b))` for the Prolog term `f(a,h(b))`.

3.2 Encoding of GMEC as hereditary Harrop formulas

We define a family of representation functions $\ulcorner\cdot\urcorner^{\cdot}$ that relate the mathematical entities we have been using in Section 2 to the terms of the logic programming language we have chosen for the implementation. Specifically, we will need to encode GMEC-structures, the associated orderings, and the GMEC-language. In the remainder of this section, we will refer to the GMEC-structure $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$.

In order to represent $\mathcal{H}$, we need to give an encoding of the entities that constitute it. To this aim, we first specify the functions $\ulcorner\cdot\urcorner^E$ and $\ulcorner\cdot\urcorner^P$ that give the concrete syntax of individual events and properties, respectively. We explicitly assume that these functions are injective, i.e. that every event e in E (property p in P) has a representation that is different from that of all other events (resp. properties). Moreover, we want $\ulcorner\cdot\urcorner^E$ and $\ulcorner\cdot\urcorner^P$ to give distinct representations to events and properties.

The next step consists in defining the translation maps for $[\cdot\rangle$, $\langle\cdot]$ and $]\cdot,\cdot[$. We represent these relations by means of the binary predicates `initiates`, `terminates` and `exclusive`, respectively. The traditional formulations of EC give an explicit representation to the occurrences of events. We utilize the unary predicate `happens` for this purpose. The corresponding representation functions are defined as follows:

$$\begin{aligned}
\ulcorner[\cdot\rangle\urcorner^I &= \{\texttt{initiates}\ \ulcorner e\urcorner^E\ \ulcorner p\urcorner^P : e \in E, p \in P, e \in [p\rangle\};\\
\ulcorner\langle\cdot]\urcorner^T &= \{\texttt{terminates}\ \ulcorner e\urcorner^E\ \ulcorner p\urcorner^P : e \in E, p \in P, e \in \langle p]\};\\
\ulcorner]\cdot,\cdot[\urcorner^X &= \{\texttt{exclusive}\ \ulcorner p\urcorner^P\ \ulcorner q\urcorner^P : p, q \in P,]p,q[\};\\
\ulcorner E\urcorner^H &= \{\texttt{happens}\ \ulcorner e\urcorner^E : e \in E\}.
\end{aligned}$$

At this point, we define the representation of the GMEC-structure $\mathcal{H}$ by taking the union of the representations of its constituent entities:

$$\ulcorner\mathcal{H}\urcorner^S = \ulcorner E\urcorner^H \cup \ulcorner[\cdot\rangle\urcorner^I \cup \ulcorner\langle\cdot]\urcorner^T \cup \ulcorner]\cdot,\cdot[\urcorner^X.$$

In Section 2, we assumed that the ordering information of a GMEC problem was specified by means of strict orders in W. When integrating GMEC into practical applications, e.g. [5], this assumption turns out to be inadequate since, in general, the host system will simply pass the raw ordering data over to the GMEC module as they are recorded. Therefore, we choose to represent this kind of information as our knowledge states and to reconstruct the corresponding strict ordering as needed. We assume the information source to be reliable, therefore, the raw ordering information constitutes a quasi-order in O. We use the binary predicate `beforeFact` to represent the atomic ordered pairs contained in a quasi-order$o \in O$. The function $\ulcorner\cdot\urcorner^O$ relates a knowledge state to its concrete syntax. It is defined as follows:

$$\ulcorner o\urcorner^O = \{\texttt{beforeFact}\ \ulcorner e_1\urcorner^E\ \ulcorner e_2\urcorner^E : (e_1, e_2) \in o\}.$$

The last entity we need to represent is the GMEC-language of $\mathcal{H}$. We encode the formulas in $\mathcal{L}_{\mathcal{H}}$ as terms in the language of hereditary Harrop formulas. Specifically, we use the ternary function symbol `period` to represent atomic formulas and the constants `not`, `and`, `or`, `must` and `may` with the obvious arities as the concrete syntax of the logical symbols of GMEC $\neg$, $\wedge$, $\vee$, $\Box$ and $\Diamond$ respectively. The representation function $\ulcorner\cdot\urcorner^L$ for GMEC-formulas is specified by the following recursive definition, based on the structure of the formula in $\mathcal{L}_{\mathcal{H}}$ being represented:

$$\begin{aligned}
\ulcorner p(e_1, e_2)\urcorner^L &= \texttt{period}\ \ulcorner e_1\urcorner^E\ \ulcorner p\urcorner^P\ \ulcorner e_2\urcorner^E\\
\ulcorner \neg\varphi\urcorner^L &= \texttt{not}\ \ulcorner\varphi\urcorner^L\\
\ulcorner \varphi_1 \wedge \varphi_2\urcorner^L &= \texttt{and}\ \ulcorner\varphi_1\urcorner^L\ \ulcorner\varphi_2\urcorner^L\\
\ulcorner \varphi_1 \vee \varphi_2\urcorner^L &= \texttt{or}\ \ulcorner\varphi_1\urcorner^L\ \ulcorner\varphi_2\urcorner^L\\
\ulcorner \Box\varphi\urcorner^L &= \texttt{must}\ \ulcorner\varphi\urcorner^L\\
\ulcorner \Diamond\varphi\urcorner^L &= \texttt{may}\ \ulcorner\varphi\urcorner^L
\end{aligned}$$

Notice that we have overloaded the symbol `not`. However, its position dictates its use: within a term, it represents the negation of $\mathcal{L}_{\mathcal{H}}$, and at the predicate level it stands as the negation-as-failure operator. In order to simplify the notation, we will write the previously defined translation maps as $\ulcorner\cdot\urcorner$ whenever the omitted subscript will be easily deducible from the context.

An implementation of GMEC in the language of HH formulas, called `GMEC`, is given below:

```
% - Transitive closure of knowledge states

before E1 E2 :- beforeFact E1 E2.

before E1 E2 :- beforeFact E1 E, before E E2.

% - Propositional formulas

holds (period Ei P Et) :-
 happens Ei, initiates Ei P, happens Et,
 terminates Et P, before Ei Et, not (broken Ei P Et).

broken Ei P Et :-
 happens E, before Ei E, before E Et,
 (initiates E Q ; terminates E Q),
 (exclusive P Q; P = Q).

holds (not X) :- not (holds X).

holds (and X Y) :- holds X, holds Y.

holds (or X Y) :- holds X; holds Y.

% - Modal formulas

holds (must X) :- not (fails_must X).

fails_must X :- not (holds X).

fails_must X :-
 happens E1, happens E2,
 not (before E1 E2), not (before E2 E1),
 beforeFact E1 E2 => not (holds (must X)).

holds (may X) :- holds X.

holds (may X) :-
 happens E1, happens E2,
 not (before E1 E2), not (before E2 E1),
 beforeFact E1 E2 => holds (may X).
```

3.3 Soundness and completeness results

In this section, we show that `GMEC` is a faithful implementation of the semantics given in Section 2.2 for GMEC. This statement is formalized in the soundness and completeness theorem (Theorem 11) that concludes the section. This result is accomplished in a number of steps: first we will need to prove that `before` is a sound and complete implementation of the transitive closure over knowledge states, then we will show that the implementation of atomic formulas and simply modalized atomic formulas is sound and complete, and finally we will be able to freely mix boolean connectives and modal operators. Due to the length of the proofs and the many details involved, this section will only sketch the soundness and completeness results for GMEC. The interested reader is referred to [3] for full detail.

We begin with a lemma about the properties of `before`. When only ordering information is concerned, we do not need to refer to the representation of the underlying GMEC-structure, but only implicitly to the representation of events. First, we show that the HH-formula `before` $\ulcorner e_1 \urcorner$ $\ulcorner e_2 \urcorner$ is provable precisely when (e_1, e_2) is in the transitive closure of the current knowledge state. Moreover, the goal `before` $\ulcorner e_1 \urcorner$ $\ulcorner e_2 \urcorner$ finitely fails exactly when (e_1, e_2) is not in the transitive closure of the current knowledge state.

Lemma 8 (*Soundness and completeness of* `before` *w.r.t. transitive closure*)

Let $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$ *be a GMEC-structure and* o *a state of knowledge, then for any* $e_1, e_1 \in E$

a. `GMEC`,$\ulcorner o \urcorner \vdash_{HH}$ `before` $\ulcorner e_1 \urcorner$ $\ulcorner e_2 \urcorner$ *iff* $(e_1, e_2) \in o^+$*;*
b. `GMEC`,$\ulcorner o \urcorner \vdash_{HH}$ `not (before` $\ulcorner e_1 \urcorner$ $\ulcorner e_2 \urcorner$`)` *iff* $(e_1, e_2) \notin o^+$. ■

On the basis of this result, we address the problem of proving that the clauses for atomic GMEC-formulas, possibly preceded by one occurrence of the connectives $\Box$ or $\Diamond$ implement the semantics of $MVI(\cdot), \Box MVI(\cdot)$, and $\Diamond MVI(\cdot)$. We start by proving a lemma that states that the predicate `broken` behaves like the negation of the meta-predicates nb.

Lemma 9 (*Correspondence between* `broken` *and* nb)

Let $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$ *be a GMEC-structure and* o *a state of knowledge, then*

a. `GMEC`,$\ulcorner \mathcal{H} \urcorner$,$\ulcorner o \urcorner \vdash_{HH}$ `broken` $\ulcorner e_1 \urcorner$ $\ulcorner p \urcorner$ $\ulcorner e_2 \urcorner$ *iff* $\neg nb(p, e_1, e_2, o^+)$ *holds in* $\mathcal{H}$*;*
b. `GMEC`,$\ulcorner \mathcal{H} \urcorner$,$\ulcorner o \urcorner \vdash_{HH}$ `not (broken` $\ulcorner e_1 \urcorner$ $\ulcorner p \urcorner$ $\ulcorner e_2 \urcorner$`)` *iff* $nb(p, e_1, e_2, o^+)$ *holds in* $\mathcal{H}$. ■

At this point, we have all the tools we need to prove that the implementation of `holds` on bare atomic formulas and atomic formulas subject to the application of a single occurrence of a modal operator behaves isomorphically to the satisfiability relation on these formulas. Therefore, `GMEC` effectively calculates the sets $MVI(w)$, $\Box MVI(w)$, and $\Diamond MVI(w)$, for any w.

Theorem 10 (`GMEC` *computes* $MVI(\cdot), \Box MVI(\cdot)$*, and* $\Diamond MVI(\cdot)$)

Let $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$ *be a GMEC-structure and* o *a state of knowledge, then*

a. `GMEC`,$\ulcorner \mathcal{H} \urcorner$,$\ulcorner o \urcorner \vdash_{HH}$ `holds(period` $\ulcorner e_1 \urcorner$ $\ulcorner p \urcorner$ $\ulcorner e_2 \urcorner$`)` *iff* $p(e_1, e_2) \in MVI(o^+)$*;*
b. `GMEC`,$\ulcorner \mathcal{H} \urcorner$,$\ulcorner o \urcorner \vdash_{HH}$ `must (period` $\ulcorner e_1 \urcorner$ $\ulcorner p \urcorner$ $\ulcorner e_2 \urcorner$`)` *iff* $p(e_1, e_2) \in \Box MVI(o^+)$*;*
c. `GMEC`,$\ulcorner \mathcal{H} \urcorner$,$\ulcorner o \urcorner \vdash_{HH}$ `may (period` $\ulcorner e_1 \urcorner$ $\ulcorner p \urcorner$ $\ulcorner e_2 \urcorner$`)` *iff* $p(e_1, e_2) \in \Diamond MVI(o^+)$. ■

We conclude this section by stating the soundness and completeness of `GMEC` with respect to the GMEC semantics.

Theorem 11 (*Soundness and completeness of* `GMEC` *w.r.t. the GMEC semantics*)

Let $\mathcal{H} = (E,\ P,\ [\cdot\rangle,\ \langle\cdot],\ ]\cdot,\cdot[)$ *be a GMEC-structure,* o *a state of knowledge and* φ *and GMEC-formula, then*

`GMEC`,$\ulcorner \mathcal{H} \urcorner$,$\ulcorner o \urcorner \vdash_{HH}$ `holds` $\ulcorner \varphi \urcorner$ *iff* $o^+ \models \varphi$. ■

4 CONCLUSIONS

This paper proposed and formally analyzed GMEC, a modal extension of EC to compute current, necessary and possible MVIs in a context where the ordering of events is relative, partial and incremental. Moreover, it presented a sound and complete implementation of GMEC as a logic program in the language of hereditary Harrop formulas. We are developing this work in several directions. On the one hand, we are considering the possibility of dealing with more complex specifications of the ordering information such as non-committed data (e.g. disjunctive orderings) and possibly inconsistent orderings. On the other hand, we are comparing `GMEC` with classical modal logics such as Sobocinski's K1.1 [8], which is characterized by the class of all finite partial orderings, i.e. by the class of finite frames whose accessibility relation is reflexive, transitive and antisymmetric.

REFERENCES

[1] K. van Belleghem: personal communication, 1995.
[2] I. Cervesato, L. Chittaro, A. Montanari: "A Modal Calculus of Partially Ordered Events in a Logic Programming Framework", *Proc. of ICLP'95: 12th International Conference on Logic Programming*, Tokyo, Japan, MIT Press, 1995, 299–313.
[3] I. Cervesato, L. Chittaro, A. Montanari: "*A General Modal Framework for the Event Calculus and its Skeptical and Credolous Variants*", Research Report, Dipartimento di Matematica e Informatica, Università di Udine, Italy, 1996 (forthcoming).
[4] I. Cervesato, A. Montanari, A. Provetti: "On the Non-Monotonic Behavior of Event Calculus for Deriving Maximal Time Intervals", *Interval Computations*, 2, 1993, 83–119.
[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] T. Dean, M. Boddy: "Reasoning about partially ordered events", *Artificial Intelligence*, 36, 1988, 375–399.
[7] M. Denecker, L. Missiaen, M. Bruynooghe: "Temporal reasoning with abductive Event Calculus", *Proc. of ECAI'92: 10th European Conference on Artificial Intelligence*, B. Neumann (ed.), John Wiley & Sons, 1992, 384–388.
[8] G. Hughes, M. Cresswell: "*An Introduction to Modal Logic*", Methuen, London, 1968.
[9] R. Kowalski, M. Sergot: "A Logic-based Calculus of Events", *New Generation Computing*, 4, 1986, 67–95.
[10] D. Miller, G. Nadathur, F. Pfenning, A. Scedrov: "Uniform proofs as a foundation for logic programming", *Annals of Pure and Applied Logic*, 51, 1991, 125–157.
[11] D.C. Moffat, G.D. Ritchie: "Modal Queries about Partially-ordered Plans", *Journal of Expt. Theor. Artificial Intelligence*, 2, 1990, 341–368.