Noname manuscript No.
(will be inserted by the editor)

Revising Event Calculus Theories to Recover from Unexpected Observations

Nikoleta Tsampanaki · Theodore Patkos · Giorgos Flouris · Dimitris Plexousakis

Received: date / Accepted: date

Abstract Recent extensions of the Event Calculus resulted in powerful formalisms, able to reason about a multitude of commonsense phenomena in causal domains, involving epistemic notions, functional fluents and probabilistic aspects, among others. Less attention has been paid to the problem of automatically revising (correcting) a Knowledge Base when an observation contradicts inferences made regarding the world state. Despite mature work on the related belief revision field, adapting such results for the case of action theories is non-trivial. This paper describes how to address this problem for deterministic, yet partially observable, domains, by proposing a generic framework in the context of the Event Calculus, along with ASP encodings of the revision algorithm and a web-based tester of the formalism implementation.

Keywords Event Calculus · Belief Revision · Action Theories · Answer Set Programming

1 Introduction

Action languages are well-established logical theories for reasoning about the dynamics of changing worlds, aiming at "*formally characterizing the relationship between the knowledge, the perception and the action of autonomous agents*" [1]. One of the most prominent action languages is the Event Calculus [2,3], which incorporates certain useful features for representing causal and narrative information that differentiate it from other similar formalisms. The Event Calculus explicitly represents temporal knowledge, enabling reasoning

The authors wish to thank the three anonymous reviewers for their insightful and very detailed comments, who helped improve significantly the content of the paper.

Institute of Computer Science, FORTH, Greece
E-mail: {tsaban, patkos, fgeo, dp}@ics.forth.gr

about the effects of a narrative of events along a time line. It also relies on a non-monotonic treatment of events, in the sense that by default there are no unexpected effects or event occurrences.

Powerful extensions of the main formalism have been developed to accommodate, for instance, epistemic extensions [4–6], probabilistic uncertainty [7,8] or knowledge derivations with non-binary-valued fluents [4]. Moreover, progress in generalizing the stable model semantics used in Answer Set Programming (ASP) has opened the way for the reformulation of Event Calculus axiomatizations into logic programs that can be executed with efficient ASP solvers [9]. This allowed for exploiting state-of-the-art tools that outperform previous SAT- or logic programming-based solvers in almost all classes of problems related to practical applications [10].

However, to the best of our knowledge, little work has been done on supporting *belief change* in the Event Calculus, in cases when the new information contradicts the already inferred conclusions. Specifically, the existing non-epistemic extensions accommodate *belief update*, which concerns beliefs that change as the result of the realization that the world has changed through some action. The epistemic extensions further focus on modeling the notions of knowledge and belief, supporting *belief expansion*, where newly acquired information can enrich the belief state of agents about aspects that were previously considered unknown.[1] Yet, the ability to accommodate, through proper *revisions*, sensed information that contradicts existing beliefs is not supported. This problem is more general than belief expansion or even than diagnosis, as it not only intends to identify the reasons that explain the contradictions, but also to suggest proper modifications of the belief state of the agent under certain, possibly domain-dependent, criteria, as placed in [13].

In this paper, we present a formal method for accommodating belief revision on top of Event Calculus axiomatizations for partially observable, deterministic domains, relying on the possible-worlds representation to give formal semantics to an agent's belief state. We formalize notions of commonsense revisions that take into consideration different belief states, such as factual (or observed), default, inferred and unknown beliefs. We further present a methodology and an ASP encoding that implements the formalism, along with optimizations to improve its performance and a web-based UI for accessing its features. This work can form the substrate for further extensions concerning a richer set of commonsense features, such as non-determinism, state constraints, introspective belief changes, non-binary aspects etc. We use the following example to foreground some of the key aspects of our problem statement.

[1] Epistemic logic is concerned with reasoning about knowledge, i.e., about facts that must be true, while doxastic logic is concerned with reasoning about the beliefs of agents. In relevant literature, it is very common to talk about epistemic notions, even when referring to reasoning about beliefs, as in [11,12], since both modal logics are derived by assigning different properties to the same accessibility relation. In the current study, we concentrate on an agent's belief state but, occasionally, we may refer to epistemic notions without necessarily referring to knowledge exclusively.

Example 1: Consider the classical Yale Shooting scenario, where a gun, that is believed to be loaded, is fired against a walking turkey. An observer may believe that, after the shot, the turkey is dead. If future observations contradict her beliefs, e.g., by noticing that the turkey is still walking, the observer will need to assess different potential revisions of her belief state, some being more intuitive than others: can it be that she was so mistaken and the shooter did not fire the gun in the first place? Or is it just that the initial, potentially default belief about the gun being loaded was not accurate? Moreover, how would the related inferences and revisions be affected if the initial state of the gun is unknown? □

Although simplistic and well-studied in relevant literature, this setting of the Yale Shooting scenario can still highlight a combination of aspects related to belief revision that can be communicated in a comprehensive manner. For example, it can be generalized to account for different levels of commonsense inferences, some of which may be domain-independent, e.g., by giving preference to revising aspects that were initially unknown rather than aspects that have already been observed, while others may be domain-dependent, e.g., considering certain observations as being less reliable than others. For such types of domains, we present in the sequel a formal methodology for revising the belief state of an agent, taking into consideration commonsense and epistemic notions. We use the Yale Shooting scenario as a running example, but also present a more generic use case in Section 6.

It should be noted that, even though the belief change literature has been used as a source of inspiration for addressing the problem, the related technical results cannot be directly applied, as they are based on assumptions that are not relevant for our setting (e.g., monotonicity of the underlying representation language). Thus, our approach leverages only the related methodologies, establishing connections among our ideas and the corresponding ideas from belief change.

A preliminary version of this work was published in [14]. Compared to that work, this paper presents progress in the following aspects. First, we extend the framework with supporting default initial knowledge and formalize the treatment of this new feature in Section 3. Second, we present a more sophisticated composite preference relation by introducing sub-preference relations and a weight function that degrades the weight over time in Section 4. Additionally, we improve our initial implementation with an optimized mechanism based on irrelevant/relevant fluents and events in Section 5. Finally, we offer a web-based tester (UI) of our formalism and implementation.

The rest of the paper is structured as follows: in Section 2,we remind the reader of the basics about the Event Calculus that will be needed in the next sections. Sections 3 and 4 give the theoretical underpinnings of our methodology describing the problem (and the proposed solution) in formal terms; in particular, Section 3 describes the Event Calculus axiomatization used for formalizing dynamic and partially observable domains, whereas Section 4 describes the belief revision formalism that was applied for revising such domains. In Section 5 we describe the implementation of the methodology in

ASP along with the web-based user interface and the related optimizations, while in Section 6 we show how our methodology is applied in a use case scenario, underlining its features and functionality. Finally, in Section 7 we discuss related work, and conclude in Section 8 with remarks about next steps.

2 Preliminaries

The Event Calculus is a narrative-based many-sorted first-order language for reasoning about action and change [2,15]. Our account of change, causality and belief is influenced and inspired by two dialects, the non-epistemic discrete time Event Calculus ($\mathcal{DEC}$) axiomatized in [16], and the epistemic extension of the Functional Event Calculus ($\mathcal{EFEC}$) proposed in [4]. The two dialects differ significantly in the signature of their modeling language, but provide well-defined foundations to model a simple doxastic action theory. From the former, we borrow the domain-independent axiomatization of causal domains restricted to deterministic settings, while from the latter we keep the modeling of possible worlds for representing epistemic notions.

As our goal is not to suggest an elaborate action theory for reasoning about beliefs in a wide range of causal domains, but rather to focus on studying a novel revision process, we make certain assumptions in the representation capacity of our underlying doxastic action language. In particular, apart from modeling deterministic domains, we also do not support the modeling of beliefs about fluent formulas or nested beliefs, and as a result beliefs about the beliefs of other agents or belief introspection, we do not model state constraints, potentially triggered events, and also we do not consider fluents released from the commonsense law of inertia. These assumptions help us build a concise set of clear and well-defined revision principles for a restricted class of causal domains, and can be lifted in future extensions.

Formally, $\mathcal{DEC}$ considers a sort $\mathcal{E}$ for events (variables $e, e', e_1, ...$), a sort $\mathcal{F}$ for fluents ($f, f', f_1, ...$) and a sort $\mathcal{T}$ for timepoints ($t, t', t_1, ...$), restricted to the integers.[2] The key predicates are $HoldsAt() \subseteq \mathcal{F} \times \mathcal{T}$ denoting the truth value of a fluent at a particular timepoint, $Happens() \subseteq \mathcal{E} \times \mathcal{T}$ capturing the occurrence of events or actions, $Initiates() \subseteq \mathcal{E} \times \mathcal{F} \times \mathcal{T}$ and $Terminates() \subseteq \mathcal{E} \times \mathcal{F} \times \mathcal{T}$, denoting that an event E causes fluent F to become true or false respectively in the next timepoint. The notions of cause, effect and inertia are captured in the *domain independent* set of axioms defined next.[3]

Definition 1 The axiomatization of the commonsense notions of persistence and causality in the deterministic version of the Discrete Event Calculus, denoted as $d\mathcal{DEC}$ in the sequel, is captured by the following set of axioms:

[2] In the sequel, variables start with a small letter, and constants with a capital letter. Wherever not explicitly stated, variables are assumed to be universally quantified. Timepoints in $\mathcal{DEC}$ are restricted to the integers, but this is going to be generalized in this study.

[3] As we are not handling fluents that are released from the law of inertia, in our subset of $\mathcal{DEC}$ we omit the *ReleasedAt* and *Releases* predicates introduced in [16].

$HoldsAt(f,t) \wedge \neg\exists e(Happens(e,t) \wedge Terminates(e,f,t)) \rightarrow$
$HoldsAt(f,t+1)$ (dDEC1)
$\neg HoldsAt(f,t) \wedge \neg\exists e(Happens(e,t) \wedge Initiates(e,f,t)) \rightarrow$
$\neg HoldsAt(f,t+1)$ (dDEC2)
$Happens(e,t) \wedge Initiates(e,f,t) \rightarrow HoldsAt(f,t+1)$ (dDEC3)
$Happens(e,t) \wedge Terminates(e,f,t) \rightarrow \neg HoldsAt(f,t+1)$ (dDEC4)

Axiom (d$\mathcal{DEC}$1) says that if a fluent is true at timepoint T and the fluent is not terminated by any event that occurs at T, then the fluent is true at $T+1$. Axiom (d$\mathcal{DEC}$2) defines a similar behavior for fluents that are false at T. Axiom (d$\mathcal{DEC}$3) says that if a fluent is initiated by some event that occurs at timepoint T, then the fluent will be true at $T+1$. Likewise, axiom (d$\mathcal{DEC}$4) says that if a fluent is terminated by some event that occurs at timepoint T, then the fluent will be false at $T+1$.

Example 2: The Yale shooting example described in Section 1 can be formalized with the following set of axioms that illustrate how the Discrete Event Calculus can be used to reach conclusions in a dynamic domain. The next three axioms constitute the domain axiomatization:
$Initiates(Load, Loaded, t)$ (2.1)
$HoldsAt(Loaded,t) \rightarrow Terminates(Shoot, Loaded, t)$ (2.2)
$HoldsAt(Loaded,t) \rightarrow Terminates(Shoot, Alive, t)$ (2.3)
(2.1) expresses that the *Load* action causes the *Loaded* fluent to become true, while (2.2) and (2.3) express that the *Shoot* action causes the fluents *Loaded* and *Alive* respectively to become false, given that *Loaded* holds true. We also need to specify the initial conditions and narrative of actions. Let us assume that the gun is loaded initially and that it is fired at $T=0$:
$HoldsAt(Loaded, 0)$ (2.4)
$Happens(Shoot, 0)$ (2.5)
In order to reach commonsense conclusions given the $d\mathcal{DEC}$ axiomatization, we need to minimize the extension of the $Happens$, $Initiates$ and $Terminates$ predicates, i.e., to assume that no unexpected events or unexpected effects occur. Traditionally, this is accomplished in the Event Calculus by means of the circumscription technique, but other techniques can be used as well. These lead to closure axioms, such as the following:
$f = Loaded \wedge (e = Shoot \vee e = Load)$ (2.6)
$Initiates(e,f,t) \equiv e = Load \wedge f = Loaded$ (2.7)
$Terminates(e,f,t) \equiv$
$\quad e = Shoot \wedge (f = Loaded \vee f = Alive) \wedge HoldsAt(Loaded, 0)$ (2.8)
$Happens(e,t) \equiv e = Shoot \wedge t = 0$ (2.9)
Now, we can reach conclusions about the state of fluents at various timepoints:
(2.6-9)$\wedge d\mathcal{DEC} \models$
$\quad HoldsAt(Alive, 0) \wedge \neg HoldsAt(Alive, 1) \wedge \neg HoldsAt(Alive, 2)$ □

The expressiveness of non-epistemic dialects, such as $\mathcal{DEC}$, is restricted to causal domains, and cannot represent the mental state of an agent having incomplete world beliefs. For example, we cannot reason about what will happen if initially the state of the *Loaded* fluent is unknown to the agent. In order

to support epistemic reasoning, we consider the dialect of $\mathcal{EFEC}$, which is inspired by the possible-worlds approach that necessitates certain ontological extensions to be defined on top of non-epistemic dialects. This is mainly due to the fact that the latter are based on a linear (not branching) time structure.

$\mathcal{EFEC}$ adds two new sorts, a sort $\mathcal{W}$ for representing possible worlds (variables $w, w', w_1, ...$) and a sort $\mathcal{I}$ for instants (variables $i, i', i_1, ...$). $\mathcal{EFEC}$ also defines a partial (possibly total) ordering $\prec \subseteq \mathcal{I} \times \mathcal{I}$ over $\mathcal{I}$. The idea is to represent time as a system of parallel lines, where each world W is understood as an identifier for a possible time line. The constant W_a of sort $\mathcal{W}$ signifies the actual world.

The function $\langle . \rangle : \mathcal{W} \times \mathcal{I} \rightarrow \mathcal{T}$ is introduced to map world/instant pairs to instances of the timepoints sort, which in $\mathcal{EFEC}$ and in the rest of this paper is no longer considered to be integers: timepoint $\langle W, I \rangle$ represents instant I in possible world W. The time structure is formalized in $\mathcal{EFEC}$ as:

$\forall t \exists w, i . t = \langle w, i \rangle$ (EFEC1)

$\langle w, i \rangle = \langle w', i' \rangle \equiv (w = w' \wedge i = i')$ (EFEC2)

$\langle w, i \rangle < \langle w', i' \rangle \equiv (w = w' \wedge i \prec i')$ (EFEC3)

$\leq, >, \geq$ have their usual meanings in terms of $=$ and $<$. In the rest of this paper, we assume that elements of the sort $\mathcal{I}$ take values in $\mathbb{N}$. Also, for simplicity, we write $\langle I \rangle$ as a shorthand for $\langle W_a, I \rangle$.

The time lines believed to be accessible at any given moment are captured by the relation $K \subseteq \mathcal{W} \times \mathcal{W}$. The epistemic fluent $K(W, W')$ represents the accessibility relation between possible worlds, as in epistemic modal logics. As usual, we formally define belief of some fluent F at some timepoint as the fact that this fluent has the same truth value in all worlds that are accessible from the world specified by the given timepoint at the given instant :

$Bel(f, \langle w', i \rangle) \equiv$

$\forall w HoldsAt(K(w, w'), \langle w', i \rangle) \rightarrow HoldsAt(f, \langle w, i \rangle)$ (DOX1)

$BelNot(f, \langle w', i \rangle) \equiv$

$\forall w HoldsAt(K(w, w'), \langle w', i \rangle) \rightarrow \neg HoldsAt(f, \langle w, i \rangle)$ (DOX2)

$BelWh(f, \langle w', i \rangle) \equiv$

$Bel(f, \langle w', i \rangle) \vee BelNot(f, \langle w', i \rangle)$ (DOX3)

We formally define $BelWh()$ as the belief that some fluent F at some timepoint has the same truth value in all worlds that are accessible from the world specified by the given timepoint, but we are not certain whether the truth value is true or false; thus, $\neg BelWh()$ represents the case that some fluent F at some timepoint is unknown, being true in at least one world and false in at least one other, all of which are accessible by the world specified at the given timepoint.

As we are interested in modeling belief rather than knowledge, we cannot assume the accessibility relation K to be an equivalence relation, as defined in $\mathcal{EFEC}$. Instead, we only assume that K is serial, which is equivalent to stating that the agent cannot believe contradictions (see [17]):

$Bel(f, \langle w, i \rangle) \rightarrow \neg BelNot(f, \langle w, i \rangle)$ (DOX4)

This is also known as the Consistency Axiom, capturing that property that there is always one possible world accessibly related to any other world. Notice

that the above axiom does not assume that the actual world is accessible too (K is not reflexive). As a result, erroneous beliefs can still be inferred, requiring a revision mechanism whenever observations (that always reflect W_a) do not comply with the agent's beliefs.

Finally, in order to apply the $d\mathcal{DEC}$ treatment of time with the $\mathcal{EFEC}$ definition of timepoints, we need to also define the addition operation:

$\langle w, i\rangle + n \equiv \langle w, i+n\rangle$, where $n \in \mathbb{N}$ (DOX5)

3 Formalization of Dynamic, Partially Observable Domains

3.1 The Event Calculus Setting

The previous subsection defined a framework for representing the belief state of an agent in a dynamic setting. Yet, the axiomatization still does not guarantee the existence of a sufficiently large number of accessible possible worlds at the initial instant to reflect an agent's initial belief state. We need a domain-independent axiom to ensure that, in the initial state, all possible worlds that are consistent with the agent's beliefs exist.

In particular, and similar in style with $\mathcal{EFEC}$, we wish to ensure that for each fluent whose truth value is unknown, any proper combination of $HoldsAt()/\neg HoldsAt()$ statements will be represented in some possible worlds. To achieve this, we first axiomatize the notion that two worlds differ at instant 0 by at most one fluent, by means of the predicate $InitiallyDifferAtMostBy \subseteq \mathcal{W} \times \mathcal{W} \times \mathcal{F}$:

$InitiallyDifferAtMostBy(w_1, w_2, f) \equiv$ (DOX6)

$\quad \forall f'.f' \neq f \rightarrow (HoldsAt(f', \langle w_1, 0\rangle) \leftrightarrow HoldsAt(f', \langle w_2, 0\rangle))$

As domain descriptions will typically include a partial specification of initial beliefs at instant 0, axiom (DOX7) below makes sure that the number of possible worlds that are accessible by the actual world justify all the initial beliefs and only these beliefs:

$\forall f.\neg BelWh(f, \langle W_a, 0\rangle) \rightarrow$ (DOX7)

$\quad \exists w_1, w_2.K(w_1, W_a) \wedge K(w_2, W_a) \wedge$

$\quad HoldsAt(f, \langle w_1, 0\rangle) \wedge \neg HoldsAt(f, \langle w_2, 0\rangle) \wedge$

$\quad InitiallyDifferAtMostBy(w_1, w_2, f)$

Given our assumption of deterministic domains, it is sufficient to preserve the number of possible worlds generated at the initial timepoint while reasoning, since there is no means of generating more worlds in future timepoints. We do not eliminate worlds either, so that we can support reasoning about the past; we can still change the truth value of the accessibility relation, though.

Based on the aforementioned formalization of belief, the knowledge base (KB) of an agent accommodating partial world descriptions at the initial and at future timepoints is described formally as follows:

Definition 2 We define an epistemic KB as $e\Phi = d\mathcal{DEC} \wedge \mathcal{EFEC} \wedge \mathcal{DOX} \wedge \Sigma \wedge e\Gamma^0 \wedge \Delta \wedge \Omega$, where

- $d\mathcal{DEC}$ is the conjunction of the deterministic Discrete Event Calculus domain-independent axioms,
- $\mathcal{EFEC}$ and $\mathcal{DOX}$ are the conjunction of the epistemic axioms that model beliefs in a setting supporting parallel time lines,
- Σ is the conjunction of the domain-dependent axioms,
- eΓ^0 is the initial beliefs, i.e., a conjunction of ground $Bel(F_i, \langle W_a, 0\rangle)$, $BelNot(F_j, \langle W_a, 0\rangle)$ axioms at instant 0,
- Δ is the narrative of actions, i.e., a conjunction of $Happens(E, \langle w, I\rangle)$,
- Ω is a conjunction of unique name axioms.

Domain axioms in Δ can be partially defined and then minimized to address the Frame Problem and related issues, as shown in Example 2.[4] eΓ^0 axioms can be partially defined as well; fluents that are explicitly stated as believed to hold or not at time instant 0 generate the set of possible worlds according to axioms (DOX5,6). We denote by $e\Phi \models \phi$ the fact that $e\Phi$ implies ϕ, where ϕ is any valid Event Calculus formula containing epistemic and/or non-epistemic predicates. We assume that from time to time we observe some part of the world, i.e., we obtain the truth value of certain fluents. Our current assumption is that observations can only contain a conjunction of $(\neg)HoldsAt()$ statements, i.e., we cannot ask for the truth value of belief predicates, as this would lead to introspection and nested beliefs. We denote by Γ^I an observation obtained at instant I.

It is interesting to note that our theory models the epistemic (belief) case, while the simpler non-epistemic case occurs as a special instance, when the agent has initially complete (and consistent) world knowledge. This results in the generation of a single possible world, which contains only $Bel()$ and $BelNot()$, but no $\neg BelWh()$ statements. Consequently, our work can be applied in both settings, without alterations.

Example 3: Consider again Example 1, but let us assume that the observer does not have information whether the gun is loaded initially. The observer's KB $e\Phi_{Yale}$ can now be expressed as follows:

$Initiates(Load, Loaded, t)$ (3.1.1)
$HoldsAt(Loaded, t) \rightarrow Terminates(Shoot, Loaded, t)$ (3.1.2)
$HoldsAt(Loaded, t) \rightarrow Terminates(Shoot, Alive, t)$ (3.1.3)
$Bel(Alive, \langle 0\rangle)$ (3.1.4)
$Happens(Shoot, \langle w, 1\rangle)$ (3.1.5)

That is, Σ = (3.1.1) $\wedge$ (3.1.2) $\wedge$ (3.1.3), $e\Gamma^0$ = (3.1.4) and Δ = (3.1.5), whereas the remaining components of $e\Phi_{Yale}$ follow from Definition 2. It can be shown that $e\Phi_{Yale} \models \neg BelWh(Loaded, \langle 0\rangle) \wedge \neg BelWh(Alive, \langle 2\rangle)$. That is, the observer cannot be certain that the turkey is alive after the occurrence of the action, due to the fact that the state of the gun is unknown before the action. Assume now that the observer receives information that is not inferred

[4] Minimization ensures that the information given is all there is and no unexpected events occur while reasoning. Although circumscription was the de facto choice for the Event Calculus in the past, our implementation relies on the use of default negation supported by the Answer Set Programming solver, as explained in Section 5.

by her current beliefs, e.g., $\Gamma^2 = HoldsAt(Alive, \langle 2 \rangle)$. Our contribution in this paper is a method for finding possible revisions that avoid the creation of inconsistency, while at the same time give preference to revisions that, considering the dynamics of the domain, seem more intuitive than others. □

3.2 Default Fluents

A vital reasoning ability of human commonsense is to make assumptions about the type and essence of ordinary situations encountered. Default reasoning, i.e., making assumptions about certain world aspects until more information is gained, can be supported by our theory by extending the axiomatization with a set of axioms defining the behavior of fluents with default values. The general idea is that an agent can initially assume that some fluents are believed to hold or not to hold by default, and to maintain this default belief for as long as no event causes a change to them. Future non-deterministic extensions of the theory may also specify a behavior, where fluents under conditions may become default again as time progresses, but for the purposes of this study it suffices to focus only on initial default beliefs. The overall objective is to enable a qualitative distinction of default from inferred beliefs at the agents' mental state, which can become a key aspect to consider when comparing alternative revisions, as explained later on.

Towards this end, we first introduce the special fluent $Def()$ to capture the default truth value of certain domain fluents. Specifically, $HoldsAt(Def(F, v), T)$ expresses that fluent F will obtain the truth value v at timepoint T when considered a default fluent, where $v \in \mathcal{V} = \{True, False\}$. We further extend the conjunction of initial agent beliefs $\mathrm{e}\Gamma^0$ to also account for default fluents. Specifically, we let $\mathrm{ed}\Gamma^0$ denote the conjunction of ground belief formulas, where for each domain fluent F, $\mathrm{ed}\Gamma^0$ may contain at most one of $Bel(F, \langle W_a, 0 \rangle)$, $BelNot(F, \langle W_a, 0 \rangle)$, $Bel(Def(F, True), \langle W_a, 0 \rangle)$, or $Bel(Def(F, False), \langle W_a, 0 \rangle)$. That is, an agent initially can believe that some domain fluents are true, some are false and some are believed true or false by default. The rest will be unknown.

Next, we axiomatize the behavior of default beliefs. For start, we assume that initially an agent can only believe that a fluent is by default true, by default false or that it is not default. In other words, we wish the agent's beliefs about the scope of default fluents to be complete at instant 0:

$\neg(Bel(Def(f, True), \langle w, 0 \rangle) \vee Bel(Def(f, False), \langle w, 0 \rangle)) \rightarrow$ (DEF1.1)
$BelNot(Def(f, True), \langle w, 0 \rangle)$

$\neg(Bel(Def(f, True), \langle w, 0 \rangle) \vee Bel(Def(f, False), \langle w, 0 \rangle)) \rightarrow$ (DEF1.2)
$BelNot(Def(f, False), \langle w, 0 \rangle)$

$\forall v, v'.Bel(Def(f, v), \langle w, i \rangle) \wedge v' \neq v \rightarrow BelNot(Def(f, v'), \langle w, i \rangle)$ (DEF1.3)

The particular truth value that default fluents are believed to obtain is captured by the following domain-independent axioms:

$Bel(Def(f, True), \langle w, 0 \rangle) \rightarrow Bel(f, \langle w, 0 \rangle)$ (DEF2.1)

$Bel(Def(f, False), \langle w, 0 \rangle) \rightarrow BelNot(f, \langle w, 0 \rangle)$ (DEF2.2)

Specifically, the combination of axioms (DEF1.1-3) and (DEF2.1-2) with the definition of belief from the previous sections specifies that at instant 0 a fluent can be unknown only if it is not believed explicitly and also it is not believed to be a default fluent. We denote the conjunction of the aforementioned (DEF1,2) axioms for default fluents as $\mathcal{DEF}$.

As it has become clear, so far, we are only interested in the initial instant. The key point with default fluents, as far as belief revision is concerned, is to enable the agent to be aware of when the truth value is due to some inference and when it is based on an initial default assumption. To accommodate this behavior, the domain dependent part of the axiomatization needs to be extended, too. Specifically, for each of the domain-dependent formulas in Σ that represent the effects of actions on the domain fluents, namely,
$\bigwedge_i (\neg)\ HoldsAt(F_i, t) \rightarrow Terminates(E, F, t)$ or
$\bigwedge_i (\neg)\ HoldsAt(F_i, t) \rightarrow Initiates(E, F, t)$
Σ needs to also contain an additional logical formula, ensuring that if the truth value of a default fluent changes, then it ceases to be considered as default:
$\bigwedge_i (\neg)\ HoldsAt(F_i, t) \wedge HoldsAt(Def(F, v), t) \rightarrow$
$Terminates(E, Def(F, v), t)$ (DEF3)
(DEF3) is not taking into consideration the beliefs of the agent, rather it is triggered at each possible world separately, potentially leading the agent to lose information whether a fluent has remained a default fluent or not, as time progresses. This is reasonable to expect, since a default fluent may be affected by some action with unknown preconditions; in that case, a causal dependency is created between this fluent and the preconditions, in the sense that if the preconditions turn out to hold the effect should hold as well, otherwise the fluent should remain unaffected. It would be counterintuitive to maintain a default truth for fluents in such cases, therefore the given axiomatization causes default fluent to become unknown. Since (DEF3) is a kind of meta-axiom that is formulated according to the structure of other domain axioms, we consider it as a *directive* for generating domain axioms when using default fluents.

Example 4: Extending the previous example, assume now that the observer considers the loaded gun as a predefined default fluent, which should be assumed to hold if there is no evidence to the contrary:
$Bel(Def(Loaded, True), \langle 0 \rangle)$ (3.2.1)
Given the effect axiom (3.1.2) defined in the domain axiomatization of $e\Phi_{Yale}$ and considering the (DEF3) directive, the axiomatization should also include:
$HoldsAt(Loaded, t) \wedge HoldsAt(Def(Loaded, v), t) \rightarrow$
$Terminates(Shoot, Def(Loaded, v), t)$ (3.2.2)
That is, $\Sigma^* = \Sigma \cup$ (3.2.2), $ed\Gamma^{*0} = e\Gamma^0 \cup$ (3.2.1) and $\Delta =$ (3.1.5), whereas the remaining components of $e\Phi^*_{Yale}$ follow from Definition 2, including also $\mathcal{DEF}$. It can be shown that $e\Phi^*_{Yale} \models Bel(Loaded, \langle 0 \rangle) \wedge BelNot(Alive, \langle 2 \rangle)$. Thus, based on the assumption that the gun is loaded by default, the observer will now infer that the turkey is not alive at instant 2 after the shooting at instant 1, a conclusion reached based on default beliefs.

We still have not resolved the contradiction, though: if the new information is $\Gamma^2 = HoldsAt(Alive, \langle 2 \rangle)$, inconsistency will emerge and the observer can be uncertain on how to revise the KB to recover from inconsistency: should she question her initial default beliefs or is it more intuitive to consider that the gun did not fire? Both choices lead to a consistent new KB. In the following, we present a methodology to consistently revise the observer's KB while accepting the new piece of information. □

4 Revision Methodology

4.1 The Revision Operator

Now, let us turn our attention to the problem of revising a KB $e\Phi$ with an observation Γ^I. Our approach will be based on well-established ideas from the belief revision literature, appropriately adapted for the problem at hand. In particular, we follow the *Principle of Consistency Maintenance* [18], which requires that the result of revising $e\Phi$ with Γ^I should be consistent. In addition, we make the standard assumption that is expressed by the *Principle of Primacy of New Information* [18] (and formalized by the *postulate of success* in the AGM postulates [13]), which states that the new observation should always be entailed after the revision.

The special characteristics of the Event Calculus force us to introduce two new principles. The first is the *Principle of Persistence of Background Knowledge*, which states that the revision process will only affect the initial beliefs ($ed\Gamma^0$) and/or the narrative (Δ). Thus, the domain-independent ($\mathrm{d}\mathcal{DEC}$), the doxastic ($\mathcal{DOX}$), the default axioms ($\mathcal{DEF}$) and the domain-dependent (Σ) axioms, as well as the unique name axioms (Ω), should not be affected. This avoids issues associated to the problem of learning the domain from observations, which is not in the scope of this paper.

The second new requirement is the *Principle of Disallowing Proactive Change*, which, informally, states that we cannot use an observation referring to a time instant I in order to add events in the narrative beyond that time instant (i.e., for $I' > I$). Essentially, this limits the direct effects of an observation (and the corresponding revision) to past timepoints, even though such effects may also have indirect ramifications related to the truth value of fluents in the future.

Finally, we adopt the *Principle of Minimal Change* [19] (also known as the *Principle of Persistence of Prior Knowledge* [18]), which states that the new KB should be as "close" as possible to the original KB; in other words, from all the possible change results (revision candidates) that satisfy the other principles, we should choose the one that retains "the most information". Note that, unlike the other principles which disallow results, the Principle of Minimal Change essentially enforces a preference among candidate results.

Following the above principles, we can formally define the set of *revision candidates* as follows:

Definition 3 Given a KB $e\Phi = d\mathcal{DEC} \wedge \mathcal{DOX} \wedge \mathcal{DEF} \wedge \Sigma \wedge e\Gamma^0 \wedge \Delta \wedge \Omega$ and an observation Γ^I, a KB $e\Phi'$ is a *revision candidate* of $e\Phi$ with Γ^i iff:

- $e\Phi'$ is of the form $e\Phi' = d\mathcal{DEC} \wedge \mathcal{DOX} \wedge \mathcal{DEF} \wedge \Sigma \wedge e\Gamma'^0 \wedge \Delta' \wedge \Omega$ (Principle of Persistence of Background Knowledge).
- No formula in $(\Delta \setminus \Delta') \cup (\Delta' \setminus \Delta)$ refers to instants $i \geq I$ (Principle of Disallowing Proactive Change).
- $e\Phi'$ is a consistent KB (Principle of Consistency).
- $e\Phi' \models \Gamma^I$ (Principle of Primacy of New Information).

The set of all revision candidates of $e\Phi$ with Γ^I will be denoted by $RC(e\Phi, \Gamma^I)$.

It is interesting to note that $RC(e\Phi, \Gamma^I)$ is always finite in finite domains, thanks to the restriction associated with the Principle of Disallowing Proactive Change. Also, Definition 3 imposes that the part $d\mathcal{DEC} \wedge \mathcal{DOX} \wedge \mathcal{DEF} \wedge \Sigma \wedge \Omega$ of all revision candidates is identical to the corresponding part of the original KB (following the Principle of Persistence of Background Knowledge), and also formalizes all other principles (except from the Principle of Minimal Change). The latter is not considered because $RC(e\Phi, \Gamma^I)$ is meant to represent all the *conceivably possible* revision results, not the optimal ones. The notion of minimal change is often subjective, context- and/or domain-dependent, so we chose to include it as a separately configurable component of our framework.

To formalize the Principle of Minimal Change, we will introduce a *preference relation* $\prec^{e\Phi,I}$. The idea is that if $e\Phi_1 \prec^{e\Phi,I} e\Phi_2$, $e\Phi_1$ is strictly more preferred than $e\Phi_2$ for the result of the revision of $e\Phi$ with an observation at time instant I. Note that this is different from the relations among interpretations [19] and formulas [20] that have been used elsewhere for the same purpose. Establishing the connection among our preference relation and these works is left for future work. It suffices to assume that $\prec^{e\Phi,I}$ is well founded (so that we can always find a minimal element in a non-empty set).

We are now ready to define the revision operator. Intuitively, the idea is that we select those elements of $RC(e\Phi, \Gamma^I)$ that are minimal with respect to $\prec^{e\Phi,I}$. In case multiple minimal elements exist, their disjunction is taken. It is also interesting to note that, in the special case when $RC(e\Phi, \Gamma^I) = \emptyset$, we do not revise the KB; this can happen, e.g., when the observation itself is inconsistent or when there is no way to satisfy the observation without changing background knowledge, such as the domain axioms. Formally:

Definition 4 The *revision operator* $\bullet$ is a binary operator, defined as follows:

- $e\Phi \bullet \Gamma^I = e\Phi$ if $RC(e\Phi, \Gamma^I) = \emptyset$.
- $e\Phi \bullet \Gamma^I = \bigvee\{e\Phi' \mid e\Phi' \in RC(e\Phi, \Gamma^I)$ and there is no $e\Phi'' \in RC(e\Phi, \Gamma^I)$ such that $e\Phi'' \prec^{e\Phi,I} e\Phi'\}$ otherwise.

4.2 Preference Relation

The above definitions (in particular Definition 4) do not specify the preference relation to be used. As already mentioned, this is a customisable component of

our framework that depends on the setting, the domain, and the wishes of the modeller for the application at hand. Nevertheless, in this subsection, we will define, for illustration purposes, a reasonable preference relation that would make sense in many domains. Although some of the choices associated with this relation may seem arbitrary, it is important to remember that it is just provided as an example. Also, our approach for defining this relation can be seen as a "template" for defining alternative relations, although, in principle, one could define $\prec^{e\Phi,I}$ in many different ways; as mentioned before, the only restriction imposed is for $\prec^{e\Phi,I}$ to be well-founded.

To define the proposed preference relation, we will leverage a parametrized *cost-based model* which assesses minimality based on the amount of information lost, modified, or gained from the original KB in order to accommodate the observation. In particular, considering two KBs $e\Phi, e\Phi'$, the *cost* to modify $e\Phi$ to result to $e\Phi'$ will be defined on the basis of the formulas that can be inferred by one of these KBs but not the other. Also, we will consider that different types of changes should weigh differently (e.g., it is more plausible that the agent erred on the truth value of a fluent, compared to not realising that an event has happened), and that this weight is also affected by the instant when the change occurs (e.g., changes in beliefs earlier in the timeline, that have stood the test of time, should be considered less prone to change, compared to beliefs later in the timeline). In the following, we formalise the above intuitions.

Types of changes. We start by defining the different types of changes that can happen. As mentioned above, different changes have different importance, so we define one set for each different type of change:

Modified non-Default Beliefs $MB^I(e\Phi, e\Phi') =$
$\{BelWh(f, \langle W_a, I\rangle) \mid e\Phi \models \neg\exists v, Bel(Def(f, v), \langle W_a, I\rangle)$, and either $e\Phi \models Bel(f, \langle W_a, I\rangle)$ and $e\Phi' \models BelNot(f, \langle W_a, I\rangle)$, or $e\Phi \models BelNot(f, \langle W_a, I\rangle)$ and $e\Phi' \models Bel(f, \langle W_a, I\rangle)\}$.
This represents all $Bel()$ or $BelNot()$ statements about non-default fluents whose truth value changed in the evolution from $e\Phi$ to $e\Phi'$ at instant I.

Modified Default Beliefs $MDB^I(e\Phi, e\Phi') =$
$\{BelWh(f, \langle W_a, I\rangle) \mid e\Phi \models Bel(Def(f, v), \langle W_a, I\rangle)$, and either $e\Phi \models Bel(f, \langle W_a, I\rangle)$ and $e\Phi' \models BelNot(f, \langle W_a, I\rangle)$, or $e\Phi \models BelNot(f, \langle W_a, I\rangle)$ and $e\Phi' \models Bel(f, \langle W_a, I\rangle)\}$.
This represents all $Bel()$ or $BelNot()$ statements about default fluents whose truth value changed in the evolution from $e\Phi$ to $e\Phi'$ at instant I.

New Beliefs $NB^I(e\Phi, e\Phi') =$
$\{BelWh(f, \langle W_a, I\rangle) \mid e\Phi \models \neg BelWh(f, \langle W_a, I\rangle)$ and $e\Phi' \models BelWh(f, \langle W_a, I\rangle)\}$.
This represents all $\neg BelWh()$ statements whose unknown value changed to known in the evolution from $e\Phi$ to $e\Phi'$ at instant I.

Lost Beliefs $LB^I(e\Phi, e\Phi') =$
$\{\neg BelWh(f, \langle W_a, I\rangle) \mid e\Phi \models BelWh(f, \langle W_a, I\rangle)$ and

$e\Phi' \models \neg BelWh(f, \langle W_a, I\rangle)$.
This represents all $Bel()$ or $BelNot()$ statements whose truth value changed to unknown in the evolution from $e\Phi$ to $e\Phi'$ at instant I.

New Events $NE^I(e\Phi, e\Phi') = \{Happens(e, \langle W_a, I\rangle) \mid$
$e\Phi \models \neg Happens(e, \langle W_a, I\rangle)$ and $e\Phi' \models Happens(e, \langle W_a, I\rangle)\}$.
This represents all events that were added in the narrative of $e\Phi$ at instant I.

Lost Events $LE^I(e\Phi, e\Phi') = \{Happens(e, \langle W_a, I\rangle) \mid$
$e\Phi \models Happens(e, \langle W_a, I\rangle)$ and $e\Phi' \models \neg Happens(e, \langle W_a, I\rangle)\}$.
This represents all events that were retracted from the narrative of $e\Phi$ at instant I.

Note that the above definitions only consider the changes at a certain instant I. This way, we can discriminate not only the different types of changes, but also the instant in which they occurred. Further, note that we do not only consider the formulas that are explicitly in $e\Phi$, but also all formulas that are implied by $e\Phi$.

The above change types are stratified in 5 levels of decreasing importance:

1. At the first level, we consider lost or new events (represented by LE, NE respectively), which are considered the most important type of change. This expresses the intuition that the agent is generally confident that the events occurring will be generally observed correctly, with no false positives or false negatives.
2. At the second level, we consider lost beliefs (represented by LB), expressing the intuition that agents are generally reluctant of "losing" beliefs and creating uncertainty.
3. At the third level, we consider modified beliefs about non-default fluents (MB), i.e., cases where the agent erred on the truth value of a non-default fluent. This case is more plausible than losing beliefs, because there is no increase in uncertainty.
4. At the fourth level, we consider modified beliefs about default fluents (MDB), i.e., cases where the agent erred on the truth value of a default fluent. This is more plausible than modifying non-default fluents, due to the intuition that the agent is less confident that her (somewhat arbitrary) default assumptions are correct, compared to non-default beliefs being correct.
5. Finally, at the fifth level, we consider the least important type of change, namely gaining new beliefs (NB); although this is generally a positive thing (compared to the other types of change), it is still worse than no change at all.

The time degradation function. As mentioned, it makes sense to assume that the truth value of fluents and events earlier in time are more "established", and thus harder to change, than later ones. To support this, we will use a function that degrades over time, expressing the intuition that it is more expensive to change beliefs about past instants than beliefs about more recent ones.

Towards this we define the time degradation function $s^I : \mathcal{I} \mapsto \mathbb{Z}$, as:
$s^I(I') = I - I' + 1$
This function increases linearly as we move towards the beginning of time (by decreasing I'), and decreases as we move towards the instant of the observation (I). For example, if an observation takes place at timepoint $\langle 3 \rangle$ and a revision candidate suggests some change in the value of a fluent or in the occurrence of an event at timepoint $\langle 1 \rangle$, then the weight of this change (expressed by $s^3(1)$) would be 3.

Cost functions. Now, we are ready to define the *cost functions* that will be used to determine the *cost* of the modifications that $e\Phi$ underwent to result to $e\Phi'$. Our cost function will be based on the sets defined above; intuitively, the larger these sets are, the more "distant" the revision candidate ($e\Phi'$) from the original KB ($e\Phi$). To support the aforementioned stratification of changes, we will in fact define five different cost functions, one for each stratum. Formally:

Definition 5 For every time instant I, we define five cost functions $cost_j^I$, $j = 1, \ldots, 5$, whose input is a pair of KBs and output an integer, such that:

1. $cost_1^I(e\Phi, e\Phi') = \sum_{I'=0}^{I} s^I(I') \cdot (|NE^{I'}(e\Phi, e\Phi')| + |LE^{I'}(e\Phi, e\Phi')|)$
2. $cost_2^I(e\Phi, e\Phi') = \sum_{I'=0}^{I} s^I(I') \cdot |LB^{I'}(e\Phi, e\Phi')|$
3. $cost_3^I(e\Phi, e\Phi') = \sum_{I'=0}^{I} s^I(I') \cdot |MB^{I'}(e\Phi, e\Phi')|$
4. $cost_4^I(e\Phi, e\Phi') = \sum_{I'=0}^{I} s^I(I') \cdot |MDB^{I'}(e\Phi, e\Phi')|$
5. $cost_5^I(e\Phi, e\Phi') = \sum_{I'=0}^{I} s^I(I') \cdot |NB^{I'}(e\Phi, e\Phi')|$

Intuitively, each cost function $cost_j^I$ considers only the changes belonging to the stratum j (as defined previously), and essentially counts them up to the time instant of the observation (I), while taking into account the instant when each change occurred by appropriately weighting it through the time degradation function s^I. This represents the cost of modifying $e\Phi$ to result to $e\Phi'$ up to time instant I, for each stratum.

The preference relation $\prec^{e\Phi,I}$. Given a cost function $cost_j^I$, it is trivial to construct a relation $\preceq_j^{e\Phi,I}$, as follows:
$e\Phi' \preceq_j^{e\Phi,I} e\Phi''$ iff $cost_j^T(e\Phi, e\Phi') \leq cost_j^T(e\Phi, e\Phi'')$.

However, this leads to 5 different preference relations, whereas Definition 4 requires a single one. To get there, we will define a *composite preference*, defined through a stratification of multiple sub-preferences in prioritized levels [21,22]. In a composite preference, we first consider the most important sub-preference; in case of a tie, we check the second most important sub-preference, and so on. Formally:

Definition 6 Given relations $\leq_1, \ldots, \leq_n$, a *prioritized composition* of $\leq_1, \ldots, \leq_n$ is a relation $\leq$, denoted by $\leq = \leq_1 \,\&\, \ldots \,\&\, \leq_n$ such that $a \leq b$ if and only if $a \leq_j b$ and $b \leq_j a$ for all j, or there is some j such that $a \leq_j b$, $b \not\leq_j a$ and $a \leq_k b$, $b \leq_k a$ for all $k < j$.

Now the preference relation $\prec^{e\Phi,I}$ is defined as the prioritised composition of $\preceq_j^{e\Phi,I}$, as follows:

Definition 7 The preference relation $\preceq^{e\Phi,I}$ is defined as the prioritised composition of $\preceq_j^{e\Phi,I}$, $j = 1, \ldots, 5$, i.e.:
$\preceq^{e\Phi,I} = \preceq_1^{e\Phi,I} \,\&\, \preceq_2^{e\Phi,I} \,\&\, \preceq_3^{e\Phi,I} \,\&\, \preceq_4^{e\Phi,I} \,\&\, \preceq_5^{e\Phi,I}$.
As usual, we write $e\Phi' \prec^{e\Phi,I} e\Phi''$ if and only if $e\Phi' \preceq^{e\Phi,I} e\Phi''$ and $e\Phi'' \not\preceq^{e\Phi,I} e\Phi'$.

It is easy to see that each $\preceq_j^{e\Phi,I}$ is a total preorder, and this is true regardless of the choice of the cost function. Thus, $\preceq^{e\Phi,I}$ is also a total preorder, and so $\prec^{e\Phi,I}$ is well-founded in finite domains (recall that $RC(e\Phi, \Gamma^I)$ is always finite in finite domains).

Moreover, the proposed $\prec^{e\Phi,I}$ relation has certain intuitively desirable formal properties. In particular, $\prec^{e\Phi,I}$ implies that "less" changes (with respect to the standard set-theoretic subset relation) are better than "more":

Proposition 1 *Consider three KBs $e\Phi, e\Phi_1, e\Phi_2$ and some time instant I. For $j = 1, 2$, set:*
$CH^I(e\Phi_j) = MK^I(e\Phi, e\Phi_j) \cup MDK^I(e\Phi, e\Phi_j) \cup NK^I(e\Phi, e\Phi_j) \cup LK^I(e\Phi, e\Phi_j) \cup NE^I(e\Phi, e\Phi_j) \cup LE(e\Phi, e\Phi_j)$.
Then, $CH^I(e\Phi_1) \subset CH^I(e\Phi_2)$, implies that $e\Phi_1 \prec^{e\Phi,I} e\Phi_2$.

Proof Consider some element $c \in MK^I(e\Phi, e\Phi_1)$. Then $c \in CH^I(e\Phi_1) \subset CH^I(e\Phi_2)$, so there is some set $(MK^I(e\Phi, e\Phi_2), MDK^I(e\Phi, e\Phi_2), \ldots)$ that contains c. However, by the definitions of those sets, the only possible case is that $c \in MK^I(e\Phi, e\Phi_2)$, so $MK^I(e\Phi, e\Phi_1) \subseteq MK^I(e\Phi, e\Phi_2)$. The same inclusion can be shown for the other sets.
Based on these results, it is easy to see that, for all $j = 1, \ldots, 5$, $cost_j^I(e\Phi, e\Phi_1) \leq cost_j^I(e\Phi, e\Phi_2)$, i.e., $e\Phi_1 \preceq_j^{e\Phi,I} e\Phi_2$, so $e\Phi_1 \preceq^{e\Phi,I} e\Phi_2$. Also, given that $CH^I(e\Phi_1) \subset CH^I(e\Phi_2)$, there will be at least one set for which the strict inclusion holds, so it cannot be the case that $e\Phi_2 \preceq^{e\Phi,I} e\Phi_1$, which concludes the proof.

As a corollary, we get that not changing a KB is always cheaper than changing it, and this will happen whenever the observation does not contradict our expectations; this is intuitively expected, because we do not want to support unsolicited changes. This also implies that $\prec^{e\Phi,I}$ is a faithful ranking, in the terminology of [19]. Formally:

Corollary 1 *$e\Phi \prec^{e\Phi,I} e\Phi'$ for all $e\Phi, e\Phi', I$.*

Corollary 2 *If $e\Phi \in RC(e\Phi, \Gamma^I)$, then $e\Phi \bullet \Gamma^I = e\Phi$.*

Corollary 3 *If $e\Phi \models \Gamma^I$, then $e\Phi \bullet \Gamma^I = e\Phi$.*

Alternatives for the cost function. It is interesting to note that there are many different preference relations that can be defined using the same ideas, and be fine-tuned for different applications. Some examples follow:

- One could consider different stratifications of the change types.
- One could consider certain fluents and/or events being more/less important than others (and having changes related to these fluents/events cost more/less). To support this, the changes corresponding to these fluents/events should be contained in different sets, and the stratifications adapted accordingly.
- One could define the time degradation function differently, or omit it altogether.
- One could adopt a less strict differentiation among the various change types, by including them all in the same stratum (or in a smaller number of strata), while assigning different numerical weights to each change type to express the difference in their importance.

Of course, even more alternatives could arise if one foregoes the cost-based definition of the preference relation altogether, and considers a different way to compute it, e.g., based on non-numerical considerations. However, we argue that the cost-based definition is quite versatile to support different needs, while at the same time being more natural and easy to understand than other alternatives, and thus more suitable for most applications. Moreover, the cost-based definition is simple to compute and well within the expressive power of most formalisms (including ASP, that is being used in this paper – see Section 5).

Example 5: Consider Example 3 again, without default knowledge, where the turkey is believed to be alive at timepoint 0. Let the shot occur at instant 1 and the observer does not know whether the gun is loaded at instant 0. Thus, the observer does not know whether the turkey is alive at instant 2. Then, a new observation is received that the turkey is alive at instant 2.

A possible reaction to the observation would be that the observer was mistaken and the gun was not fired at timepoint 1. That is, $\Delta' = \emptyset$ and $e\Gamma'^0 = e\Gamma^0$. So, a revision candidate of $e\Phi_{Yale}$ would be $e\Phi'_{Yale} = \mathcal{DEC} \wedge \mathcal{DOX} \wedge \Sigma \wedge e\Gamma'^0 \wedge \Delta' \wedge \Omega$. Another possible revision would be that the gun was not loaded in the first place. That is, $\Delta'' = \Delta$ and $e\Gamma''^0 = (3.1.4) \wedge BelNot(Loaded, \langle 0 \rangle)$. The revision candidate of $e\Phi_{Yale}$ is now $e\Phi''_{Yale} = \mathcal{DEC} \wedge \mathcal{DOX} \wedge \Sigma \wedge e\Gamma''^0 \wedge \Delta'' \wedge \Omega$.

Consequently, $e\Phi'_{Yale}, e\Phi''_{Yale} \in RC(e\Phi_{Yale}, \Gamma^2)$. Note that many other KBs are included in $RC(e\Phi_{Yale}, \Gamma^2)$, but all of them would introduce more changes (with regards to the subset relation) than these two (see also Proposition 1), so they are not considered for the sake of simplicity.

To find the $\prec^{e\Phi,I}$-minimal element(s) of $RC(e\Phi_{Yale}, \Gamma^2)$, we only need to compare $e\Phi'$ with $e\Phi''$. The weight associated with the sub-preference cost functions is set to $s(I) = I - I' + 1$, so the corresponding costs are: $cost_1^{\langle 2 \rangle}(e\Phi_{Yale}, e\Phi'_{Yale}) = 2$, $cost_1^{\langle 2 \rangle}(e\Phi_{Yale}, e\Phi''_{Yale}) = 0$, so $e\Phi''_{Yale} \prec_2^{e\Phi,I} e\Phi'_{Yale}$ and $e\Phi''_{Yale}$ is the revision result, as it keeps the narrative of events intact. No-

tice that we do not need to check for the second sub-preference relation as there is inequality from the first level of preferences.

Intuitively, our algorithm (and preference) implies that the optimal revision is to assume that the gun was not loaded in the first place, replacing the lack of information about this fluent. This gives the least possible cost, as we gain information on the state of the turkey at instant 2, as well as on the state of the gun at instants 0 and 1, while retaining the event narrative intact. Had we accepted the revision that the shooter did not fire the gun, we would lose inferred beliefs and event occurrences at various instants, and as a result, the cost would be greater at the first level of sub-preference relations. The initial beliefs and the belief revision results are also presented in Figure 4 below, where a discussion about how such examples can be implemented by logic-based reasoners is also given. □

5 Implementation

This section discusses how the proposed framework has been encoded as an ASP program and how domain axiomatizations can be executed through a web-based user interface running a state-of-the-art ASP reasoner at the back-end. We also discuss computational complexity aspects and suggest directions for improvements. The system and all encodings are available online[5].

5.1 System Architecture and Encodings

The system architecture and information flow implementing the proposed framework is shown in Figure 1. The figure shows the two main reasoning steps performed whenever a new observation arrives. The first step is concerned with checking whether the new piece of information is consistent with the agent's beliefs and, if not, the second step generates appropriate revisions. A Java parser is responsible for automating the whole process, feeding the appropriate axiomatizations to the reasoner at each step and transferring the result to the UI. Reasoning is performed by implementing the Event Calculus axiomatizations as ASP rules and by utilizing the Clingo reasoner [23].

The first reasoning step consists of the Discrete Event Calculus epistemic and non-epistemic axioms (i.e., $\mathcal{DOX}$, $d\mathcal{DEC}$), the domain-dependent axioms, the initial beliefs, the main program, and the $\mathcal{DEF}$ axioms, all implemented in ASP. This reasoning step is responsible for progressing the world step, given a narrative of actions. At any point during this narrative, reasoning can be interrupted by a new observation.

The output of this step is a number of answer sets, denoting the set of worlds that the agent believes possible. In other words, our implementation relies on the *cautious* inference of ASP, in the sense referred to as the *ASP-world* methodology in [24]: each answer set specifies the state of fluents in

[5] `https://demos.isl.ics.forth.gr/BRinEC/`

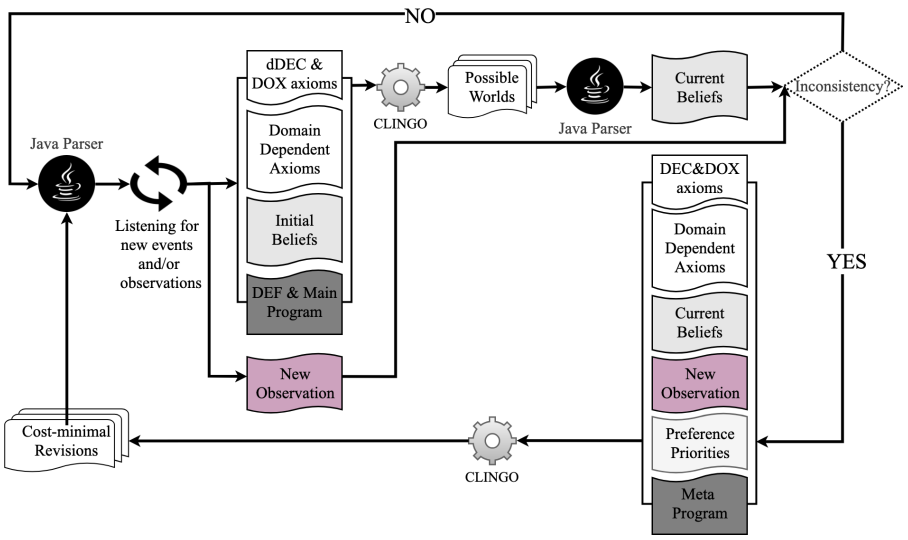

Fig. 1: The reasoning loop for revising the KB of an agent.

Table 1: Encoding of the doxastic axiomatization.

$\%\mathcal{DOX}$ axiomatization
% DOX1.1
$holdsAt(F, I)$:- $bel(F, (I))$.
% DOX2.1
-$holdsAt(F, I)$:- $belNot(F, (I))$.
% DOX3
$belWh(F, (I))$:- $bel(F, (I))$.
$belWh(F, (I))$:- $belNot(F, (I))$.
% DOX4
:- $bel(F, (I)), belNot(F, (I))$.
% DOX6,7
$\{holdsAt(F, 0)\}$:- $domFluent(F)$, not $belWh(F, (0))$.

each possible world and, for a fluent to be believed true (or false), it must hold (resp., not hold) in all generated answer sets.

Table 1 presents the $\mathcal{DOX}$ encoding that implements the aforementioned representation of possible worlds as answer sets. The implementation of (DOX6,7) axioms shown is in the form of a choice rule that generates for each unknown fluent at time instant 0 one answer set where it holds and another where it does not. In other words, it creates all combinations of the truth values of unknown fluents. Similarly, for fluents that are believed to be true (or false) axioms (DOX1.1) and (DOX 1.2) make sure that they hold (resp. do not hold) in *all* generated answer sets. These axioms are triggered based on the infor-

mation stored in the initial beliefs module, which contains the world state at instant 0, the narrative of actions, as well as the fluents that are considered to obtain a default value. This module encodes for example axioms, such as $bel(alive, (0))$., referring to the actual world only. Any fluent not mentioned in this module will consequently be regarded as unknown.

The implementation of the Event Calculus axiomatization is based on a discrete treatment to time following previous similar implementations (see for instance the F2LP encoding[6]). Finally, the main program module contains a rule that restricts the domain to fluents not released from the commonsense law of inertia, as well as, two more rules that encode the axiomatization of the default fluents behavior, denoted as $\mathcal{DEF}$.

A Java parser after this reasoning step transforms the information contained in the generated answer sets, i.e., the possible worlds, into the agent's belief predicates. Four special ASP predicates, in particular $belOriginal(F, I)$, $belNotOriginal(F, I)$, $unknownOriginal(F, I)$ and $happensOriginal(F, I)$ are used to represent the belief of the agent before any revision takes place. These are stored in the module named "current beliefs". This module and the new observations module, which is implemented as an ASP constraint, are used to examine whether an inconsistency arises.

In case of no inconsistency, the program goes back to the initial state of the loop, asserting the new information, progressing inference according to the remaining actions in the narrative, and listening to new events or observations. In case of inconsistency, the second reasoning step is performed, aiming to generate cost-optimal revisions, taking as input the new observations module, the current beliefs module and the set of worlds that were produced in the first reasoning step. This step also considers the preference priorities and the meta program, which encode in ASP the axiomatization of the belief revision process.

The most important module is the revision meta-program, which implements the revision algorithm. It computes the revision using the cost-optimal revisions generator, via a logic program. Roughly, this program generates combinations of fluents in the initial state, as well as combinations of event occurrences at each instant, for every possible world, in order to find models that lead to a consistent KB and comply with the new observation (revision candidates).

The cost associated with each revision candidate is calculated, based on the cost function described in Section 4.2 and the associated preference relations. The encoding of this behavior in ASP is shown in Table 2. The cost predicate calculates the penalty generated for each truth value or event that is different from the output of the current beliefs module. Then, optimization statements filter out all non-optimal revision candidates (answer sets), according to the predefined preference priorities. At the end, the program returns a disjunction of the optimal candidates.

[6] `http://reasoning.eas.asu.edu/f2lp/index.html`

Table 2: Encoding of the prioritized composition of the preference relation.

% Time degradation function
$point(I, X)$:- $time(I), X = maxstep - I + 1$.

% Lost Beliefs Change
$penaltyLB(X, F, I)$:-
not $belWh(F, (I)), belOriginal(F, (I)), time(I), fluent(F), point(I, X)$.
$penaltyLB(X, F, I)$:-
not $belWh(F, (I)), belNotOriginal(F, (I)), time(I), fluent(F), point(I, X)$.
% (similarly for the other types of change)

% Calculation of the overall cost for each type of change
$costE(N)$:- $N = \#sum\{X, F, T : penaltyE(X, F, I)\}$.
$costLB(N)$:- $N = \#sum\{X, F, I : penaltyLB(X, F, I)\}$.
$costMB(N)$:- $N = \#sum\{X, F, T : penaltyMB(X, F, I)\}$.
$costMDB(N)$:- $N = \#sum\{X, F, T : penaltyMDB(X, F, I)\}$.
$costNB(N)$:- $N = \#sum\{X, F, T : penaltyNB(X, F, I)\}$.

% Optimization of costs. The higher the number is next to @, the higher its priority is
$\#minimize\{S@5 : costE(S)\}$.
$\#minimize\{S@4 : costLB(S)\}$.
$\#minimize\{S@3 : costMB(S)\}$.
$\#minimize\{S@2 : costNB(S)\}$.
$\#minimize\{S@1 : costMDB(S)\}$.

5.2 Analysis of Computational Complexity and Performance Improvements

The performance of our implementation is affected in part by the computational complexity of reasoning with the underlying ASP encodings. In principle, the complexity of an answer set program stems from the number of terms that the reasoner has to ground for the computation of answer sets, which is the most time-intensive task. During this process, a problem expressed as an ASP logic program, is cast as a variable-free program by a grounder, which is then fed into an ASP solver. The solver takes the propositional program and computes its answer sets (or aggregations of them).

Different classes of ASP programs belong to different complexity classes, as analysed in [25], depending for instance on whether disjunction is used in the head of rules, depending on the type of decision problem etc. In our case, the complexity class that our system belongs to is mainly affected by the use of optimization statement and the fact that we need to generate all answer sets. As such, deciding whether a fluent holds in an optimal stable model of our encoding is in Δ_2^P. If optimization is not a key prerequisite, the same reasoning task would be in NP. These results apply to solving the grounded program. For the first-order case, the ground instantiation of a program is exponential in the size of it.

Apart from the ASP encodings, the modeling of the reasoning problem itself can be characterized as a computationally intensive one. The second reasoning step of the workflow in particular, constitutes an important parameter affecting the overall performance, which is dominated by the number of

possible elements of $RC(e\Phi, \Gamma^I)$. Specifically, the meta-program generates all possible combinations of fluents in the initial state for every possible world, as well as all combinations of event occurrences at each later instant. According to the possible-worlds specifications, the number of possible worlds depends on the number of unknown fluents, i.e., in a domain of F fluents, U of which are unknown, we have 2^U possible worlds. In the worst case scenario, where all F fluents are unknown, the number of possible initializations of all fluents at instant 0 for every possible world is $O(2^F * 2^F) = O(2^{2F})$. Additionally, the number of all possible event occurrences combinations at each instant, supposing that we have E events and I instants, is $O(2^E * (I-1))$ (we make no use of event occurrences at instant I as their effects occur in the next instant that we do not consider). Thus, when computing $RC(e\Phi, \Gamma^I)$, fluent initializations and event narratives are combined, leading to a size of $O(2^{2F+E} * (I-1))$. Note that this is only the worst-case scenario, as it is unusual to have all fluents unknown in the domain.

To downscale the number of fluents and events combination that are generated in our belief revision algorithm, we use the notion of *relevance*. The idea stems from the realization that certain fluents and events may obtain a truth value that cannot cause changes to the truth value of the observed fluents. As a result, these "irrelevant" fluents/events need not be considered during computation. This reduces the number of fluents/events considered (F, E in the above formula), thereby reducing significantly the average complexity of the algorithm.

To formally define this notion, we first need the following concept:

Definition 8 Consider a KB $e\Phi$, a fluent F, an event E and an instant I. We define the following:

- $e\Phi^{+F} = e\Phi \setminus \{HoldsAt(F, \langle w, 0\rangle), \neg HoldsAt(F, \langle w, 0\rangle)\} \cup \{HoldsAt(F, \langle w, 0\rangle)\}$
- $e\Phi^{-F} = e\Phi \setminus \{HoldsAt(F, \langle w, 0\rangle), \neg HoldsAt(F, \langle w, 0\rangle)\} \cup \{\neg HoldsAt(F, \langle w, 0\rangle)\}$
- $e\Phi^{+E,I} = e\Phi \setminus \{Happens(E, \langle w, I\rangle), \neg Happens(E, \langle w, I\rangle)\} \cup \{Happens(E, \langle w, I\rangle)\}$
- $e\Phi^{-E,I} = e\Phi \setminus \{Happens(E, \langle w, I\rangle), \neg Happens(E, \langle w, I\rangle)\} \cup \{\neg Happens(E, \langle w, I\rangle)\}$

Informally, $e\Phi^{+F}$ is the KB resulting by replacing whatever initial beliefs an agent has about F with a positive initial belief about F i.e., $Bel(F, \langle W, I\rangle)$; similarly for the other cases, but note that, for events, the instant is also relevant.

Then, we define when two KBs have an equivalent axiomatization:

Definition 9 Given a KB $e\Phi = d\mathcal{DEC} \wedge \mathcal{DOX} \wedge \mathcal{DEF} \wedge \Sigma \wedge ed\Gamma^0 \wedge \Delta \wedge \Omega$, we denote by $e\bar{\Phi} = \mathcal{DEC} \wedge \mathcal{DOX} \wedge \mathcal{DEF} \wedge \Sigma \wedge \Omega$ the part of the KB consisting of the Discrete Event Calculus domain-independent axioms, the epistemic axioms, the default axioms, the domain-dependent axioms and the unique name axioms, respectively. Two KBs $e\Phi_1, e\Phi_2$ will be called *axiomatization-equivalent*,

denoted by $e\Phi_1 \equiv_a e\Phi_2$, if and only if $e\bar{\Phi}_1 \equiv e\bar{\Phi}_2$, where $\equiv$ is the standard logical equivalence.

Now, we have the tools necessary to define irrelevant fluents and events:

Definition 10 Consider a KB $e\Phi$ and an observation Γ^I.

- We say that a fluent F is *irrelevant* to the KB $e\Phi$ and the observation Γ^I, if and only if, for all $e\Phi' \equiv_a e\Phi$, $e\Phi'^{+F} \vDash \Gamma^I$ if and only if $e\Phi'^{-F} \vDash \Gamma^I$.
- We say that an event E is *irrelevant* to the KB $e\Phi$ and the observation Γ^I, if and only if, for all $e\Phi' \equiv_a e\Phi$, and for all $i \leq I$, $e\Phi'^{+E,i} \vDash \Gamma^I$ if and only if $e\Phi'^{-E,i} \vDash \Gamma^I$.

Intuitively, a fluent is irrelevant if and only if a change in its initial truth value does not affect the truth value of the observation. Note that the check is made against all KBs with the same domain axiomatization; this is necessary, because changes in other fluents/events (as part of the revision process) may render relevant a seemingly irrelevant fluent. This can happen if there is no causal connection (through domain axioms) between the irrelevant fluent and the observed fluent. For events, a similar definition is used, except that the check is made against all timepoints.

Since irrelevant fluents and events do not affect the truth value of the observed fluent, the revision operator should not change their truth value; in other words, revision candidates that change the truth values of irrelevant fluents/events should not be optimal ones. Given this, the algorithm can safely ignore them, and consider a much smaller space while searching for the optimal revision candidate. The following proposition proves the above claims in more formal terms:

Proposition 2 *Consider a KB $e\Phi$ and an observation Γ^I. Suppose a revision algorithm based on the preference relation $\prec^{e\Phi,I}$, as defined in Section 4.2, and set $e\Phi' = e\Phi \bullet \Gamma^I$. Consider also a fluent F that is irrelevant to $e\Phi$ and Γ^I, and an event E that is irrelevant to $e\Phi$ and Γ^I. Then:*

- *$e\Phi \vDash Bel(F, \langle W_a, 0\rangle)$ if and only if $e\Phi' \vDash Bel(F, \langle W_a, 0\rangle)$*
- *$e\Phi \vDash BelNot(F, \langle W_a, 0\rangle)$ if and only if $e\Phi' \vDash BelNot(F, \langle W_a, 0\rangle)$*
- *For all timepoints T, $e\Phi \vDash Happens(E, T)$ if and only if $e\Phi' \vDash Happens(E, T)$*
- *For all timepoints T, $e\Phi \vDash \neg Happens(E, T)$ if and only if $e\Phi' \vDash \neg Happens(E, T)$*

Proof . Suppose initially that $e\Phi \vDash Bel(F, \langle W_a, 0\rangle)$. Suppose also, for the sake of contradiction, that $e\Phi' \nvDash Bel(F, \langle W_a, 0\rangle)$. Then, $HoldsAt(F, \langle w, 0\rangle) \in e\Phi$ and $HoldsAt(F, \langle w, 0\rangle) \notin e\Phi'$ for all w that are accessible by W_a. Now, set $e\Phi'' = e\Phi' \setminus \{\neg HoldsAt(F, \langle w, 0\rangle), HoldsAt(F, \langle w, 0\rangle)\} \cup \{HoldsAt(F, \langle w, 0\rangle)\}$. Using Definitions 8, 10, the definition of $RC(e\Phi, \Gamma^T)$ (Definition 3), and the fact that $e\Phi' \in RC(e\Phi, \Gamma^I)$, it is easy to see that $e\Phi'' \in RC(e\Phi, \Gamma^T)$. Also, by construction, $e\Phi'' \prec^{e\Phi,I} e\Phi'$, because it contains at least one less change. This

contradicts the construction of $e\Phi'$.
Using similar arguments, we can prove that, if $e\Phi \models BelNot(F, \langle W_a, 0\rangle)$ then $e\Phi' \models BelNot(F, \langle W_a, 0\rangle)$ and that, if $e\Phi \models \neg BelWh(F, \langle W_a, 0\rangle)$ then $e\Phi' \models \neg BelWh(F, \langle W_a, 0\rangle)$. This essentially proves the first two bullets.
For the third and fourth bullets, we can similarly prove, for any given timepoint T, that $e\Phi \models Happens(E, T)$ implies that $e\Phi' \models Happens(E, T)$, and that $e\Phi \models \neg Happens(E, T)$ implies that $e\Phi' \models \neg Happens(E, T)$. This, combined with the fact that there is no uncertainty with regards to the occurrence or not of an event, proves the third and fourth points.

Now, we are ready to describe the optimization algorithm. The first Java program parses the new observation module/file to define which fluent is mentioned in it. Then, it makes use of the domain-dependent axioms and calculates a relation graph among the domain fluents and events and finds which of them are related to this specific fluent and which are not, as described above. Then, it generates two new files of domain-dependent axioms and initial beliefs, containing only the relevant rules, choice rules, constraints, initial information at timepoint 0, etc. Those modules/files are used in place of the originals whenever necessary. Thus, the proposed revision program deals with relevant fluents only.

5.3 System Overview

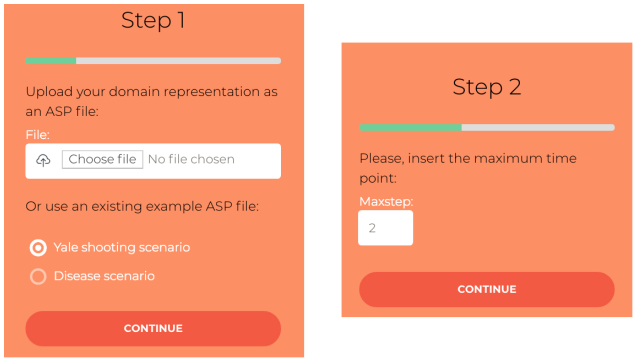

Fig. 2: Steps 1 & 2.

Our system is a prototype designed to show case our work, providing users the ability to test the framework in five simple steps. First, the users can either upload their own ASP axiomatizations or select any of the existing examples (Fig. 2, left). Second, the maximum instant up to which the cost of changes will be considered needs to be provided; this coincides with the time when the new observation is received (Fig. 2, right). In the third step, the users insert

the initial beliefs at instant 0 about the domain fluents and about the event narrative (Fig. 3, left). They can additionally define which of the fluents are to be treated as default and specify their corresponding default truth value. In the next step (Fig. 3, middle), the users can insert a new observation, which will trigger the revision. Last, the priorities among the sub-preference relations can be specified, as introduced in Section 4.2. There is also the option to use the predefined priorities that were proposed in the same section (Fig. 3, right).

Once the users complete these steps, the system visualizes the result of the belief revision process as produced by the system (Figure 4); note that the original beliefs (before the revision) are also presented for comparison, as well as the differences between the two KBs.

Fig. 3: Steps 3 & 4 & 5.

6 Use Case Scenario: Medical Assistant Agent

The following scenario is a modification and simplification of real use cases met by a human medical assistant treating patients with diabetes, whose responses regarding the way they follow their treatment sometimes do not comply with what they actually do. Consider the case of a medical assistant agent who is treating a patient. Based on the symptoms, the agent believes that the patient is having disease 1 and maybe disease 2 too, for each of which a specific cure exists. The agent also believes that the patient is normally reliable and follows

Fig. 4: Yale Shooting Scenario(Example 3): Initial Beliefs & Belief Revision Results.

the prescribed instructions. To begin with, the agent's KB $e\Phi_{disease}$ can be described by the following axiomatization:

$$HoldsAt(ReliablePatient, t) \wedge \neg HoldsAt(Disease2, t) \wedge HoldsAt(Disease1, t) \rightarrow Initiates(ProposeCure1, CuredPatient, t) \quad (6.1)$$

$$HoldsAt(ReliablePatient, t) \wedge \neg HoldsAt(CuredPatient, t) \wedge HoldsAt(Disease1, t) \rightarrow Terminates(ProposeCure1, Disease1, t) \quad (6.2)$$

$$HoldsAt(ReliablePatient, t) \wedge \neg HoldsAt(Disease1, t) \wedge HoldsAt(Disease2, t) \rightarrow Initiates(ProposeCure2, CuredPatient, t) \quad (6.3)$$

$$HoldsAt(ReliablePatient, t) \wedge \neg HoldsAt(CuredPatient, t) \wedge HoldsAt(Disease2, t) \rightarrow Terminates(ProposeCure2, Disease2, t) \quad (6.4)$$

$$\neg HoldsAt(ReliablePatient, t) \rightarrow Initiates(ChangeAttitude, ReliablePatient, t) \quad (6.5)$$

$$HoldsAt(ReliablePatient, t) \rightarrow Terminates(ChangeAttitude, ReliablePatient, t) \quad (6.6)$$

$$HoldsAt(Def(ReliablePatient, v), t) \rightarrow Terminates(ChangeAttitude, Def(ReliablePatient, v), t) \quad (6.7)$$

The domain axiomatization above expresses how each cure affects the corresponding disease and how the state of the patient is affected, accordingly. It also specifies that the assumption of the patient being a reliable one will cease to hold if the agent changes her attitude.

Initially, the agent believes that the agent is having disease 1, but is unsure about disease 2. It also believes that the patient is sick and will follow the prescriptions correctly:

$$Bel(Def(ReliablePatient, True), \langle 0 \rangle) \quad (6.8)$$

$$BelNot(CuredPatient, \langle 0 \rangle) \quad (6.9)$$

$$Bel(Disease1, \langle 0 \rangle) \quad (6.10)$$

The agent suggests to the patient to first take cure 1 at instant 0 and then cure 2 at instant 1:

$$Happens(ProposeCure1, \langle w, 0 \rangle) \quad (6.11)$$

$$Happens(ProposeCure2, \langle w, 1 \rangle) \quad (6.12)$$

So far, we have $\Sigma = (6.1) \wedge (6.2) \wedge (6.3) \wedge (6.4) \wedge (6.5) \wedge (6.6) \wedge (6.7)$, $ed\Gamma^0 = (6.8) \wedge (6.9) \wedge (6.10)$ and $\Delta = (6.11) \wedge (6.12)$, whereas the remaining components of $e\Phi_{disease}$ follow from Definition 2, along with the $\mathcal{DEF}$ set of axioms. It can be shown that $e\Phi_{disease} \models Bel(CuredPatient, \langle 2 \rangle)$.

Assume now that the agent receives information that contradicts its current inferences, e.g., $\Gamma^2 = \neg HoldsAt(CuredPatient, \langle 2 \rangle)$ (note that $e\Phi_{disease} \not\models BelNot(CuredPatient, \langle 2 \rangle)$), meaning that the patient is still sick at instant 2. The agent needs to assess different potential revisions of its belief state. Considering our prioritization scheme described in the previous section, the only optimal and commonsense explanation would be that the patient is probably not reliable and does not take the proposed medication, as neither of the cures worked.

That is, $e\Gamma'^0 = BelNot(ReliablePatient, \langle 0 \rangle) \wedge BelNot(CuredPatient, \langle 0 \rangle) \wedge Bel(Disease1, \langle 0 \rangle)$. So, a revision candidate of $e\Phi_{disease}$, would be $e\Phi'_{disease} = \mathcal{DEC} \wedge \mathcal{DOX} \wedge \mathcal{DEF} \wedge \Sigma \wedge e\Gamma'^0 \wedge \Delta \wedge \Omega$. Consequently, $e\Phi'_{disease} \in RC(e\Phi_{disease}, \Gamma^2)$. Note that many other KBs are included in $RC(e\Phi_{disease}, \Gamma^2)$, but other revision candidates would propose retracting actions from the narrative, which are more expensive alternatives according to our chosen preference relation. This represents the agent's certainty that it actually does propose cures to the patient, except for extreme cases of failure.

Fig. 5 shows the optimal revision that accommodates the new observation into our knowledge base. The commonsense notion behind this revision is that the patient must be unreliable, contrary to what the agent initially and by default believed. Thus, it is less "expensive" to revise a default truth value than to retract event occurrences from the KB narrative. Notice that the left part of Figure 5 presents the initial state of the agent's KB, while the right part presents the revised KBs, marking with red the beliefs that are different between the two. In our case, there is one optimal revision, but it could have been more equally optimal.

7 Related Work

The problem of identifying how to adapt an agent's Knowledge Base to accommodate a new observation has been addressed in various ways and in many different contexts. For instance, related works include work on abduction, (Assumption-based) Truth Maintenance Systems or view updating, all of which propose ways to identify which of the existing knowledge to adapt or remove in order to accommodate the new information consistently.

However, the most relevant related field is the field of *belief change* (also known as *belief revision*), which is a mature field of study dealing with the adaptation of a KB in the face of new information [13]. Traditionally, two types of change have been considered: *revision* and *update* [19]. Revision deals with cases where the new information is some observation or refinement of our knowledge about the world, whereas update deals with cases where the adaptation is dictated by some action or event that changed the world itself.

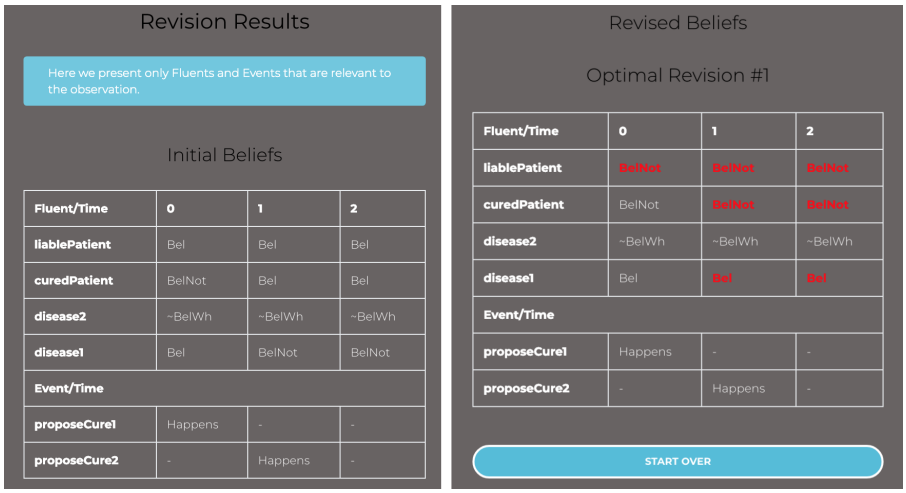

Fig. 5: Medical Assistant Agent: Initial Beliefs & Belief Revision Results.

The case of belief update is inherently captured by the semantics and reasoning of Event Calculus, where one can explicitly declare events, as well as the effects and preconditions of such events. However, studies regarding the revision of action theories when our observations of the actual world are inconsistent with the theory's predictions regarding the world's state are limited.

Most works in the classical belief change literature are dealing with the so-called *classical logics* [26], which have certain nice properties, both in terms of semantics (monotonicity, compactness, inclusion of the classical tautological implication, etc) and in terms of syntax (closed with respect to the usual operators $\wedge, \vee, \neg$, etc), Extensions of these theories to apply for ontological languages [26,27], or compact and monotonic logics in general [28] have been considered as well. However, to the best of our knowledge, no such study exists for non-monotonic formalisms, partly because many non-monotonic formalisms (most notably, defeasible logic, default logic and paraconsistent logics) have inherent ways to reason under inconsistency without trivializing inference. Thus, technical results from the related literature are not directly applicable in our setting.

Studies that account for epistemic considerations of the Event Calculus are more closely related to ours. More specifically, the $\mathcal{EFEC}$ variant introduced in [4,5] is the first to rely on the possible worlds semantics to reason about knowledge. $\mathcal{EFEC}$ supports a multitude of features, such as reasoning about the future and past, or dealing with non-determinism and concurrency. Our work utilizes the same underlying structures to formalize the treatment of epistemic notions, extending them with the ability to revise contradicting knowledge, although it is currently significantly limited in the set of supporting features. In [6], a different epistemic extension of discrete-time Event Calculus theories is presented, using a deduction-oriented model of knowledge.

Beyond the Event Calculus, epistemic extensions based on possible worlds for reasoning about actions and knowledge have been developed in the context of other calculi. The first approach that inspired this direction of research is owed to [29], who presented a Kripke-like formulation of epistemic notions of modal logic in action languages by reifying possible worlds as situations. [30] adapted this framework in the Situation Calculus, using possible situations to specify how the mental state of an agent should change with ordinary and sense actions, providing also a solution to the frame problem for knowledge. Other studies introduced further features: [31] adapted the model in the context of the Fluent Calculus, [32] covered concurrent actions, while [33] introduced epistemic modalities for groups of agents. Non-possible-worlds based epistemic action frameworks include [34–39]. In all these frameworks, knowledge is assumed to be always correct and observations that contradict inferred knowledge will lead to inconsistency.

The ability to deal with belief changes has lately started to gain interest within other action languages, as in [11] and [12] in the Situation Calculus, but without taking time into account. [40] developed a new action formalism for revision of temporal belief bases; even though related to our work, in [40] the problem of revising theories in Event Calculus is not directly addressed, but instead a new logic of action is defined that is closer to propositional logic, thereby allowing technical results from the belief change literature to be directly applicable in that framework.

It is also worth mentioning that the same goal of revising a knowledge base to accommodate observations not initially inferred is served by another field in Knowledge Representation, namely Inductive Logic Programming (ILP). Belief revision and ILP approach the problem from different perspectives. For instance, ILP is primarily interested in learning new domain rules, laying a lot of emphasis on the aspect of generalization in learning. In approaches such as ours, the beliefs about the domain axiomatization are not questioned. Also, the notion of minimality is different; in ILP minimality is seen as preference towards hypotheses with fewer literals. Yet, as some of the technical differences can, under conditions, be overcome, studies, such as [41,42] can be considered closely related to ours. The theory in [41,42] uses a different Event Calculus dialect to learn causal rules from observations through an ILP treatment of learning. Although the formalism cannot explicitly accommodate lack of knowledge or the assertion of event occurrences as possible revisions, one can see similarities in the treatment of revision of logical formulae.

8 Conclusion

This paper reported on work related to a formal framework for changing Event Calculus theories in the face of new (and potentially unexpected) observations. Our framework is necessary in the cases where an intelligent agent observes, or otherwise becomes aware of, information that contradicts what was expected (predicted) by the underlying theory. Even though the rich technical results

from the belief change literature are not generally applicable to our setting, we leveraged on some key ideas and adapted them for our purposes. Our approach was based on a set of principles and a preference relation that models the well-known Principle of Minimal Change. We used the standard approach of introducing a *preference relation* $\prec^{e\Phi,I}$ among different potential revision results; the idea is that if $e\Phi_1 \prec^{e\Phi,I} e\Phi_2$, then $e\Phi_1$ is strictly more preferred than $e\Phi_2$ for the result of the revision of $e\Phi$ with an observation.

In the future, we are planning on establishing stronger connections with existing results from belief change (e.g., satisfaction of certain postulates, or connections between our preference relation and various selection functions or orderings that have been used in other contexts), thereby more thoroughly understanding the properties of the proposed framework. Further, even though our theoretical framework is generic enough to support more complex flavours of action theories and $\mathcal{DEC}$, our implementation will need to be significantly extended to support different Event Calculus dialects and a richer set of commonsense features such as non-determinism, state constraints, and introspective belief changes. Last but not least, in order to support a complete software agent for on-line reasoning that responses to a sequence of observations, we will also need to extend our framework with iterated belief revision formalisms, as discussed in [43].

References

1. F. Van Harmelen, V. Lifschitz, and B. Porter. *Handbook of Knowledge Representation.* Elsevier Science, San Diego, USA, 2007.
2. R. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4(1):67–95, 1986.
3. R. Miller and M. Shanahan. Some alternative formulations of the event calculus. *Computational Logic: Logic Programming and Beyond, Essays in Honour of R. Kowalski Part 2*, 2408(1):452–490, 2002.
4. R. Miller, L. Morgenstern, and T. Patkos. Reasoning about knowledge and action in an epistemic event calculus. In *Commonsense-13*, 2013.
5. J. Ma, R. Miller, L. Morgenstern, and T. Patkos. An Epistemic Event Calculus for ASP-based reasoning about knowledge of the past, present and future. In *LPAR-13*, pages 75–87, 2013.
6. T. Patkos and D. Plexousakis. Reasoning with knowledge, action and time in dynamic and uncertain domains. In *IJCAI-09*, 2009.
7. A. Skarlatidis, A. Artikis, J. Filippou, and G. Paliouras. A probabilistic logic programming event calculus. *TPLP*, 15:213–245, 2015.
8. F. A. D'Asaro, A. Bikakis, L. Dickens, and R. Miller. Foundations for a probabilistic event calculus. In *LPNMR-17*, pages 57–63, 2017.
9. P. Ferraris, J. Lee, and V. Lifschitz. Stable models and circumscription. *Artificial Intelligence*, 175(1):236–263, 2011.
10. J. Lee and R. Palla. Reformulating the situation calculus and the event calculus in the general theory of stable models and in answer set programming. *JAIR*, 43(1):571–620, 2012.
11. S. Shapiro, M. Pagnucco, Y. Lespérance, and H.J. Levesque. Iterated belief change in the situation calculus. *Artificial Intelligence*, 175(1):165–192, 2011.
12. C. Schwering, G. Lakemeyer, and M. Pagnucco. Belief revision and progression of knowledge bases in the epistemic situation calculus. In *IJCAI-15*, 2015.
13. C. Alchourron, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50:510–530, 1985.

14. N. Tsampanaki, G. Flouris, and T. Patkos. Steps towards commonsense-driven belief revision in the event calculus. In *Proceedings of the Thirteenth International Symposium on Commonsense Reasoning, COMMONSENSE*, 2017.
15. Rob Miller and Murray Shanahan. Some alternative formulations of the event calculus. In *Computational logic: logic programming and beyond*, pages 452–490. Springer, 2002.
16. E.T. Mueller. *Commonsense Reasoning: An Event Calculus Based Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2^{nd} edition, 2015.
17. Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning About Knowledge*. MIT Press, Cambridge, MA, USA, 2003.
18. M. Dalal. Investigations into a theory of knowledge base revision: Preliminary report. In *AAAI-88*, pages 475–479, 1988.
19. H. Katsuno and A.O. Mendelzon. On the difference between updating a knowledge base and revising it. In *KR-91*, 1991.
20. P. Gardenfors and D. Makinson. Revisions of knowledge systems using epistemic entrenchment. In *TARK-88*, pages 83–95, 1988.
21. Periklis Georgiadis, Ioannis Kapantaidakis, Vassilis Christophides, Elhadji Mamadou Nguer, and Nicolas Spyratos. Efficient rewriting algorithms for preference queries. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 1101–1110. IEEE, 2008.
22. Werner Kießling. Foundations of preferences in database systems. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 311–322. VLDB Endowment, 2002.
23. Martin Gebser, Benjamin Kaufmann, Roland Kaminski, Max Ostrowski, Torsten Schaub, and Marius Schneider. Potassco: The Potsdam answer set solving collection. *AI Communications*, 24(2):107–124, 2011.
24. Marc Denecker, Joost Vennekens, Hanne Vlaeminck, Johan Wittocx, and Maurice Bruynooghe. *Answer Set Programming's Contributions to Classical Logic*, pages 12–32. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
25. Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. *Answer Set Solving in Practice*. Morgan & Claypool Publishers, 2012.
26. G. Flouris, D. Plexousakis, and G. Antoniou. On generalizing the AGM postulates. In *STAIRS-06*, pages 132–143, 2006.
27. G. Qi and J. Du. Model-based revision operators for terminologies in Description Logics. In *IJCAI-09*, pages 891–897, 2009.
28. M.M. Ribeiro, R. Wassermann, G. Flouris, and G. Antoniou. Minimal change: Relevance and recovery revisited. 201:59–80, 2013.
29. R. C. Moore. A formal theory of knowledge and action. In *Formal Theories of the Commonsense World*, pages 319–358. J. Hobbs, R. Moore (Eds.), 1985.
30. R. Scherl and H. Levesque. Knowledge, action, and the frame problem. *Artificial Intelligence*, 144(1-2):1–39, 2003.
31. M. Thielscher. Representing the knowledge of a robot. In *KR-00*, pages 109–120, 2000.
32. R.B. Scherl. Reasoning about the interaction of knowlege, time and concurrent actions in the situation calculus. In *IJCAI-03*, pages 1091–1096, 2003.
33. R.F. Kelly and A.R. Pearce. Complex epistemic modalities in the situation calculus. In *KR-08*, pages 611–620, 2008.
34. L. Morgenstern. Knowledge preconditions for actions and plans. In *IJCAI-87*, 1987.
35. R. Demolombe and M. P. Pozos-Parra. A simple and tractable extension of situation calculus to epistemic logic. In *ISMIS-00*, pages 515–524, 2000.
36. T. C. Son and C. Baral. Formalizing sensing actions – a transition function based approach. *Artificial Intelligence*, 125(1-2):19–91, 2001.
37. R. Petrick and H. Levesque. Knowledge equivalence in combined action theories. In *KR-02*, pages 303–314, 2002.
38. S. Vassos and H. Levesque. Progression of situation calculus action theories with incomplete information. In *IJCAI-07*, 2007.
39. Y. Liu and G. Lakemeyer. On first-order definability and computability of progression for local-effect actions and beyond. In *IJCAI-09*, 2009.
40. M. Van Zee, D. Doder, M. Dastani, and L. Van Der Torre. AGM revision of beliefs about action and time. In *IJCAI-15*, pages 3250–3256, 2015.

41. Nikos Katzouris, Alexander Artikis, and Georgios Paliouras. Incremental learning of event definitions with inductive logic programming. *Machine Learning*, 100(2):555–585, Sep 2015.
42. Nikos Katzouris, Alexander Artikis, and Georgios Paliouras. Parallel online learning of event definitions. In *Inductive Logic Programming - 27th International Conference, ILP 2017, Orléans, France, September 4-6, 2017, Revised Selected Papers*, pages 78–93, 2017.
43. Adnan Darwiche and Judea Pearl. On the logic of iterated belief revision. In *Proceedings of the 5th conference on Theoretical aspects of reasoning about knowledge*, pages 5–23. Morgan Kaufmann Publishers Inc., 1994.