A Framework for Belief Update

Paolo Liberatore

Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, I-00198 Roma, Italy
Email: liberato@dis.uniroma1.it

Summary. In this paper we show how several different semantics for belief update can be expressed in a framework for reasoning about actions. This framework can therefore be considered as a common core of all these update formalisms, thus making it clear what they have in common. This framework also allows expressing scenarios that are problematic for the classical formalization of belief update.

1 Introduction

Belief update and reasoning about actions are two well studied areas of research about the evolution of knowledge over time. The similarities between these two fields have already been pointed out by some researchers: for example del Val and Shoham [4] use a theory of action to derive a semantics for belief update; Li and Pereira [8] use a Ginsberg-like semantics for updating a theory of actions.

In this paper we present a very simple action description language [6] with narratives that allows expressing several different update semantics. The basic principles of this language has already been investigated in the literature. Indeed, the basic semantics of this language can be seen as a proper restriction of the language $\mathcal{L}$ by Baral et al. [1]. What is new in this paper is not the language itself, but rather the way it is able to express update semantics.

To introduce the language, we consider an example similar to the evergreen Yale Shooting Problem.

initially $Loaded$
initially $Alive$
$Alive$ holds at 3
$Shoot$ happens at 2
$Unload$ causes $\neg Loaded$
$Shoot$ causes $\neg Alive$ if $Loaded$

Short explanation of the syntax: at time 0 Fred is alive, and the gun is loaded. Fred is still alive at time 3. This is the meaning of the initially and holds at propositions. The last two propositions specify the effect of actions: the action *Unload* causes the gun to be loaded no longer, while the action *Shoot* causes Fred to die, if the gun is loaded.

According to the original semantics of the basic action description language $\mathcal{A}$ [6], this domain description is inconsistent. This can be intuitively explained as follows: at time 0 the gun is loaded. Since nothing happens between time 0 and 2, the gun remains loaded. As a result, the effect of shooting at Fred at time 2 causes him to die, since the gun is still loaded.

Such inconsistent scenarios are very common in the field of belief revision and update. Suppose for example we have loaded the gun at time 0. Then, we have done nothing modifying the domain of interest (e.g. we go out for a walk, we have a nap, we just do nothing at all, etc.) When we shoot the gun, Fred does not die. This is surprising, since we expected the gun to be still loaded. However, it is very easy to find an explanation: someone unloaded the gun while we was not looking at it. Such conclusion can be drawn assuming that some actions may take place at some time points, and this is initially not known.

In languages with narratives, such that the language $\mathcal{AU}$ introduced in this paper, such a deduction is possible. Note that it is not only a matter of finding an explanation of already known facts. For example, we can conclude $\neg$*Loaded* holds at 2 from the domain description above. Such an inference is clearly impossible in the basic action description language $\mathcal{A}$.

The example describes a prototypical scenario of belief update: we have a set of facts which are known to holds at a certain time point (e.g. the gun is loaded and Fred is alive at time 0). In a subsequent time point something is observed (e.g. Fred is alive at time 3). The possible inconsistency between the facts and the observation is explained as due to changes happened in the world. In this paper, the assumption is that all changes are caused by actions.

The formalization of change given in belief update is very simple. If T is a set of known facts, and P is an observation, $T * P$ denotes the result of updating T with P, that is, our knowledge after the observation of P. The use of this notation seemed the natural choice to the first researchers in the field, since what we want to formalize is indeed the update of T with P.

This notation is very simple, but sometimes it does not allow to express enough information. The example of the gun contains information that cannot be formalized using the star notation. For example, there is no way to express the fact that it is impossible that Fred becomes alive, once it is dead. Such information cannot be represented using the notation $T * P$, since the only information expressed in this way is the old set of facts T and the observation P. Another problem is the impossibility of deciding what is true in time points before the update. In the example, *Loaded* is false at time 2. However, $T * P$ only expresses the result of the update, that is, what is known at the time of the observation (in this case, at time 3). As a result, there is no way to even ask what is true at time 1, or 2, etc. Finally, there are problems in formalizing the process of iterated update. For example, $(T * P_1) * P_2$ is different from the intuitive result of incorporating two observations P_1 and P_2 (for an explanation of why, we refer the reader to the borrowed car example [5]).

All these issues have already been pointed out by researchers in the field, for example Boutilier [2,3] and Li and Pereira [8]. However, most of these formalisms employ an ad-hoc syntax and semantics. The framework introduced in this paper allows for formalizing all those forms of update. The semantics of the language generalizes many semantics given for belief update, and thus it allows for a better comparison between them.

The benefits of $\mathcal{AU}$ are twofold: it is at the same time a useful extension of action theories with narratives, and it allows an easy and intuitive formalization (in a standard way) of theories of belief update.

The paper is organized as follows: in the next section we describe the syntax and the semantics of the language $\mathcal{AU}$. The syntax of $\mathcal{AU}$ is similar to that of action description languages with narratives. As a result, we can define a "classical" semantics for it, as well as a semantics that formalizes actions that are not known to be happened. We prove that many belief update semantics can be captured this way. Finally, we compare our approach with other ones dealing with updates and action theories, and discuss possible extensions of this work.

2 The Language $\mathcal{AU}$

2.1 Syntax

The alphabet of the language is composed by three mutually disjoint sets: the set of actions, the set of fluents, and the set of time points. In this paper we assume that the set of time points is the set of non-negative integers.

A fluent literal is a fluent possibly preceded by the negation symbol $\neg$. A fluent expression is a propositional formula over the alphabet of fluents. Thus, all the fluent literals are also fluent expressions, and if E_1 and E_2 are fluent expressions, so are $E_1 \wedge E_2$, $E_1 \vee E_2$, and $\neg E_1$.

A domain description is composed of three parts: behavioral, historical, and actual. If D is a domain description then D_B, D_H, and D_A are its behavioral, historical, and actual parts, respectively.

Behavioral Part. Is the set of effect propositions, and is the part of the domain that specifies how the domain behaves in response to actions. An effect proposition has the syntax

$$A \textsf{ causes } F \textsf{ if } P_1, \ldots, P_m$$

where F is a fluent *literal*, $P_1, \ldots, P_m$ are fluent expressions, and A is an action. The meaning is that the action A causes the fluent literal F to become true, if the fluent expressions $P_1, \ldots, P_m$ are currently true. For this reason, the fluent expressions $P_1, \ldots, P_m$ are called the preconditions of the proposition, and F is called the effect.

Historical Part. Is the specification of the actions that are known to have been executed. A happens proposition is a statement of the form

A happens at t

where A is an action and t is a time point. The meaning is clear: the action A is executed at time point t.

Actual Part. Is the set of propositions that specify the status of a fluent at a certain time point.

E after $A_1; \ldots ; A_m$ from t

where E is a fluent expression, $A_1; \ldots ; A_m$ are actions and t is a time point. The meaning is that the fluent expression E is true after executing the actions $A_1; \ldots ; A_m$ in sequence starting from the time point t. This propositions allows to specify both the status of a "real" time point, and the status of hypothetical situations. When $m = 0$ (i.e. no actions) the proposition is written E holds at t, its meaning being that the fluent expression E is true in the time point t. On the other hand, when $t = 0$, we write E after $A_1; \ldots ; A_m$. What is the difference between propositions like E holds at t and E after $A_1; \ldots ; A_m$? The first one refers to a specific time point t. The second one refers to a sequence of actions. It is possible that the actions executed from 0 are *not* the sequence $A_1; \ldots ; A_m$. If this is the case, E after $A_1; \ldots ; A_m$ is a form of conditional knowledge: if the actions $A_1; \ldots ; A_m$ *were* executed then E would be true. On the other hand, E holds at t refers to the real status of the world at a certain time point.

2.2 Classical Semantics

In this section we present the semantics of the language, according to the hypothesis that all the actions that are executed are known.

A state is a set of fluent names. A fluent literal without negation F is true in the state σ if $F \in \sigma$, false otherwise. A fluent expression $\neg E$ is true in σ if and only if E is false in σ. A fluent expression $E_1 \wedge E_2$ is true in σ if both E_1 and E_2 are true in σ. A fluent expression $E_1 \vee E_2$ is true in σ if either E_1 is true in σ or E_2 is true in σ.

A transition function Φ is a function from the set of pairs (A, σ), where A is an action and σ a state, to the set of states. With $\Phi(A, \sigma)$ we want to represent the state obtained performing the action A in the state σ. We abbreviate $\Phi(A_m, \Phi(A_{m-1}, \ldots, \Phi(A_1, \sigma) \ldots))$ as $\Phi(A_1; \ldots ; A_m, \sigma)$. This is the state obtained after executing the sequence of actions $A_1; \ldots ; A_m$ in σ.

Let $V^+_{D_B}(A,\sigma)$ be the set of the fluent names F (i.e. positive fluent literals) such that there exists an effect proposition A causes F if $P_1,\ldots,P_m$ in the behavioral part of the domain description D and $P_1,\ldots,P_m$ are true in σ. Intuitively, $V^+_{D_B}(A,\sigma)$ represents the set of fluents whose value must became true when the action A is performed in the state σ.

In a similar manner, $V^-_{D_B}(A,\sigma)$ is the set of fluents whose value must became false, and thus is defined as the set of fluent names F such that there exists an effect proposition A causes $\neg F$ if $P_1,\ldots,P_m$ in D_B and $P_1,\ldots,P_m$ are true in σ.

The transition function associated to a behavioral part D_B is the (partial) function Ψ_{D_B} defined as

$$\Psi_{D_B}(A,\sigma) = \begin{cases} (\sigma \cup V^+_{D_B}(A,\sigma)) \backslash V^-_{D_B}(A,\sigma) & \text{if } V^+_{D_B}(A,\sigma) \cap V^-_{D_B}(A,\sigma) = \emptyset \\ \text{undefined} & \text{otherwise} \end{cases}$$

We used the subscript D_B here to stress the fact that the transition function of a domain description is determined by its behavioral part only. We assume that the transition function associated to a domain description D is always total. This can be verified in polynomial time.

The sequence of actions associated to a time point t is defined as the sequence of actions $B_1;\ldots;B_k$ that have been happened before t. Formally, given a set of happens propositions H, we define

$S(H,t) = B_1;\ldots;B_k$ such that

1. $\{B_1$ happens at $t_1,\ldots,B_k$ happens at $t_k\} \subseteq H$
2. $0 \leq t_1 < t_2 < \cdots < t_k < t$
3. there is no other proposition C happens at t'
   in H such that $0 \leq t' < t$

$S(H,t)$ is the sequence of actions that have took place in the time interval between the time points 0 and t.

We define interpreted structures and models as follows.

Definition 1. An interpreted structure is a 3-tuple $M = (\sigma_0, \Phi, H)$, where σ_0 is a state, Φ is a transition function, and H is a set of happens at propositions.

Definition 2. An interpreted structure $M = (\sigma_0, \Phi, H)$ is a *model* of a domain description $D = D_B \cup D_H \cup D_A$ (written $M \models D$) if and only if

1. $\Phi = \Psi_{D_B}$
2. $H = D_H$
3. for each pair of actions A_1 and A_2, and each time point t, it does not hold A_1 happens at $t \in H$ and A_2 happens at $t \in H$ (non-concurrency).

4. for each proposition E after $A_1; \ldots; A_m$ from t in D_A, the fluent expression E is true in the state $\Phi(S(H,t); A_1; \ldots; A_m, \sigma_0)$.

A domain description is consistent if it has models. A domain description entails a proposition E after $A_1; \ldots; A_m$ from t if and only if, for each $M = (\sigma_0, \Phi, H)$ such that $M \models D$, the fluent expression E is true in the state $\Phi(S(H,t); A_1; \ldots; A_m, \sigma_0)$. If this is the case, we write $D \models E$ after $A_1; \ldots; A_m$ from t.

2.3 Update Semantics

The semantics of the previous section does not take into account actions that happened, but of which we have no knowledge. For example, the domain description

$$D = \{A \text{ causes } F,\ \neg F \text{ holds at } 0,\ F \text{ holds at } 1\}$$

is not consistent. This is because the value of a fluent remains unchanged if there is no action modifying it. Since there is no happens proposition specifying that an action happened in the time point 0, the value of the fluent F at 1 should be the same of that at 0. Instead, the truth value of the fluent is changed.

Intuitively, it is clear that the action A happens at 0, and this causes the fluent F to become true. However, such an inference is not allowed in the semantics of the previous section, which assumes that the only actions that have been happened are those specified in the domain description.

In this section we present a semantics that allows the inference of statements about actions which are not known to be happened. First of all, we define a model with abduced actions as follows.

Definition 3. An interpreted structure $M = (\sigma_0, \Phi, H)$ is a model with abduced actions for the domain description $D = D_B \cup D_H \cup D_A$ (written $M \models_A D$) if and only if:

1. $\Phi = \Psi_{D_B}$
2. $D_H \subseteq H$
3. for each pair of actions A_1 and A_2, and each time point t, it does not hold A_1 happens at $t \in H$ and A_2 happens at $t \in H$ (non-concurrency).
4. for each proposition E after $A_1; \ldots; A_m$ from t in D_A, the fluent expression E is true in the state $\Phi(S(H,t); A_1; \ldots; A_m, \sigma_0)$.

The only difference between this definition and the one given in the previous section is the fact that H can be a superset of D_H, rather than D_H itself. Of course, this way arbitrarily large sets of happens propositions are allowed to be part of H. To this extent, a definition of minimality is needed. We assume that there is an ordering $\preceq$ between interpreted structures.

Definition 4. A minimal model M of a domain description D is a minimal (w.r.t. $\preceq$) model with abduced actions of D.

Thus, "minimal model" is indeed a shorthand. We define a domain description D to be consistent if it has at least one minimal model. A domain description D entails a proposition E after $A_1; \ldots; A_m$ from t if and only if, for each minimal model M of D, the fluent expression E is true in the state $\Phi(S(H,t); A_1; \ldots; A_m, \sigma_0)$. If this is the case, we write $D \models_A E$ after $A_1; \ldots; A_m$ from t.

The last point to be defined is the ordering $\preceq$. The choice of $\preceq$ depends on the knowledge about the domain. A general principle is that a model with less happens statements should be preferred (i.e. should be lower than, according to $\preceq$) over models with more happens statements. This leads to the following definition.

Definition 5. The standard ordering $\preceq_S$ is defined as:

$$(\sigma_0, \Psi_0, H_0) \preceq_S (\sigma_1, \Psi_1, H_1) \quad \text{iff} \quad \begin{cases} \sigma_0 = \sigma_1 \\ \Psi_0 = \Psi_1 \\ H_0 \subseteq H_1 \end{cases}$$

Using this specific ordering, $\mathcal{AU}$ can be seen as a fragment of the logic $\mathcal{L}$ by Baral et al. [1]. What makes $\mathcal{AU}$ interesting is the fact that, using different orderings, it allows for expressing different update semantics, thus characterizing a number of natural processes of abducting execution of actions.

The entailment relation $\models_A$ obtained from the standard ordering $\preceq_S$ can be used to express the scenario of the example described in the introduction. Indeed, one can prove that the domain description entails for example $\neg$*Loaded* holds at 2, which is intuitively the only possible reason of why Fred is still alive. Note that it is also possible to formalize the similar scenario in which we *know* that nothing happens between time 0 and 2: just add an action *Nop*, without effects, and two happens propositions *Nop* happens at 0 and *Nop* happens at 1 to the domain description. This new domain description is inconsistent: in this case, this is the intuitive outcome.

3 Belief Update using $\mathcal{AU}$

In this section we show how several definitions of belief update can be formalized in a domain of actions using the language $\mathcal{AU}$. There are two reasons to do this. The first is that this formalization allows a better understanding of the definitions of update. For example Winslett's update can be expressed by introducing an action that change the value of a variable, and minimizing the set of actions happened.

Moreover, by giving possible definitions of the ordering $\preceq$, we solve the problem of not complete specification of the entailment relation $\models_A$. Indeed, the ordering defined could be used for domain descriptions different from those given from the formalization of update.

We consider the following update definitions: Winslett's update [14], Katsuno and Mendelzon's updates [7], and Boutilier's abduction-based update [2]. We do not consider Boutilier's event based update [3] due to the lack of space, but this update can be expressed in our formalism.

We use the following notations: if P is a propositional formula, then $Mod(P)$ is the set of its models. Conversely, if A is a set of models, then $Form(A)$ is a propositional formula whose set of models is A. Thus, *Form* is a multi-valued function, since there are many formulas sharing the same set of models. This is not a problem in this work.

3.1 Winslett's Update

Consider a propositional formula T representing the state of the world. This information is assumed to be correct, but not (necessarily) complete. When a change in the world occurs, this description of the world must be modified. The assumption behind belief update is that what we know about the change is a propositional formula P that is true in the new situation. Winslett's approach is model-based, that is, the result of the update $T *_W P$ is defined in terms of the sets of models of T and P.

The underlying assumption in belief revision and update is that of minimal change: the knowledge base T should be changed as little as possible, in the process of incorporation of the update P.

Winslett's update [14] operates on a model by model base. Let I be an interpretation, and let $\leq_I$ be the ordering on interpretations defined as

$$J \leq_I Z \text{ iff } Diff(I,J) \subseteq Diff(I,Z)$$

where $Diff(I,J)$ is the set of variable to which I and J disagree. Intuitively, $J \leq_I Z$ means that, since J and I have more literals assigned to the same truth value than Z and I, the interpretation J must be considered to be closer to I that Z.

The update of the k.b. T when a new formula P becomes true after a change is defined considering each model of T separately.

$$Mod(T *_W P) = \bigcup_{I \in Mod(T)} \min(Mod(P), \leq_I))$$

We show that Winslett's update can be easily expressed in our framework. Let X be the alphabet of T and P. We define a domain description as follows. The set of fluents is the set of variables X. The intuitive explanation is: the set of fluent is the set of facts that may change over time, and this is also the meaning of the fluents in reasoning about actions. For each variable x_i there is an actions A_i. This action formalizes the change of value of the variable x_i between time points.

The domain description is built as follows. For each variable x_i there are two effect propositions:

$$D_B = \bigcup_{x_i \in X} \{A_i \ \mathsf{causes}\ x_i \ \mathsf{if}\ \neg x_i,\ A_i \ \mathsf{causes}\ \neg x_i \ \mathsf{if}\ x_i\}$$

The historical part of the domain is empty: $D_H = \emptyset$. Let $n = |X|$, that is, the number of variables. The actual part of the domain description is composed of two propositions:

$$D_A = \{T \ \mathsf{holds\ at}\ 0,\ P \ \mathsf{holds\ at}\ n\}$$

Thus, $D = D_B \cup D_A$. This formalization is a very intuitive one: the fluents are facts, and each action changes the value of a fact. This definition captures Winslett's semantics of update.

Theorem 1. *Let D be the domain description corresponding to T and P. Then, for each propositional formula Q over the alphabet X, it holds $T *_W P \models Q$ if and only if $D \models_A Q$ holds at n (using the standard ordering $\preceq_S$).*

P is assumed to hold at time n because we do not allow concurrent action.

3.2 Katsuno and Mendelzon's Update

Katsuno and Mendelzon [7] defined a family of updates, rather than a specific operator. They also proved that Winslett's operator is a sub-case of their definition.

Let $O = \{\leq_I \mid I \text{ is an interpretation}\}$ be a family of partial orderings over the set of the interpretations, one for each interpretation I. In other words, for each interpretation I there is a partial ordering $\leq_I$ over the set of the interpretations. An interpretation I represents a complete description of the world. $J \leq_I Z$ means that the situation represented by the interpretation J is considered more plausible than the situation of Z. As a result, assuming that there has been a transition from I to J requires less change than the change from I to Z. Thus, assuming that I represents the current state, the result of the update should be:

$$Mod(Form(I) *_{KM} P) = \min(Mod(P), \leq_I)$$

If the current k.b. is not composed of a single interpretation, this must be done for each $I \in Mod(T)$:

$$Mod(T *_{KM} P) = \bigcup_{I \in Mod(T)} \min(Mod(P), \leq_I)$$

Note that Katsuno and Mendelzon define a set of update operators rather than a single one: indeed, each family of orderings define a specific KM operator. As a result, in order to specify an actual update, a family of orderings must be defined.

There is a simple way to capture and Katsuno and Mendelzon's update in our framework. Given a family of orderings (one for each interpretation) we define the domain description as the one given in the previous section. The ordering used is defined as follows.

Definition 6. Given a family of partial ordering $O = \{\leq_I\}$, one for each interpretation I, we define an ordering over interpreted structures $\preceq_{KM}$ as $(\sigma_0, \Phi_0, H_0) \preceq_{KM} (\sigma_1, \Phi_1, H_1)$ if and only if

1. $\sigma_0 = \sigma_1$.
2. $\Phi_0 = \Phi_1$.
3. $\Phi_0(S(H_0, n), \sigma_0) \leq_{\sigma_0} \Phi_1(S(H_1, n), \sigma_1)$.

Note that there is an ordering $\preceq_{KM}$ for each family of orderings over the interpretations. Thus, the formally correct notation should be $\preceq_O$, but we use $\preceq_{KM}$ for simplicity. The following theorem shows that we are indeed formalizing the Katsuno and Mendelzon updates.

Theorem 2. *For each Katsuno and Mendelzon update, and for each 3-tuple of propositional formulas T, P, and Q, it holds $T * P \models Q$ if and only if $D \models_A Q$* holds at *n, using the ordering $\preceq_{KM}$ as in Definition 6.*

3.3 Abduction-Based Update

The rationale of the abduction-based update [3] is that the events that change the world can be modeled by an abductive semantics. Some of these events may be more plausible than others. In order to explain the change, we choose only the ones we consider to be more plausible.

Since this update requires the specification of the outcome of events, and their plausibility, the current knowledge base T and the update P do not suffice to evaluate the updated k.b.. This kind of updates, in which some extra information is required is called *update schema*. It can be view as a family of updates, one of each set of events and their plausibility. Giving the events and their plausibility is equivalent to selecting a specific update of the family.

We now give the formal definition of the update. A more detailed explanation can be found in the paper where this update is introduced [3]. In order to explain the changes, we have a set of *events* E. Each event e is a function from interpretations to sets of interpretations. Thus, for each interpretation I, $e(I)$ is a set of interpretations. The meaning of $J \in e(I)$ is that the possible world represented by the interpretation J is one of the possible outcomes of the event e, if this event occur in the world represented by the interpretation I. An event e is said to be deterministic if $e(I)$ is always composed of a single interpretation.

As seen in the informal explanation above, not all the events are considered equally plausible. To represent the relative plausibility of events we have a family of preorders $O = \{\leq_I \mid I \in \mathcal{M}\}$, one for each interpretation I. When $e \leq_I s$ the event e is considered more likely to happen that s, in the world represented by the interpretation I. We denote by $e <_I s$ the fact that e is *strictly* more likely than s; formally, that $e \leq_I s$ but not $s \leq_I e$.

Let T be the current k.b. and P the update. The set of explanations of P is the set of events whose occurrence can explain the fact that P is now true. There are two possible definitions.

Definition 7. The set of weak explanations of P is

$$Expl(I, P) = \min(\{e \mid e(I) \cap Mod(P) \neq \emptyset\}, \leq_I)$$

The set of predictive explanations of P is

$$Expl_p(I, P) = \min(\{e \mid e(I) \subseteq Mod(P)\}, \leq_I)$$

The outcome of the update is defined in terms of the *progression* of a possible world I.

Definition 8. The progression of an interpretation I is the set

$$Prog(I, P) = \bigcup \{e(I) \cap Mod(P) \mid e \in Expl(I, P)\}$$

The progression of an interpretation can be defined also for predictive explanations. The updated k.b. is defined as the union of all the progressions.

Definition 9. The result of updating T with P is[1]

$$Mod(T *_{ABD} P) = \bigcup \{Prog(I, P) \mid I \in Mod(T)\}$$

In this definition we assume the use of the weak explanations. A similar definition can be given using predictive explanations instead.

We show how the abduction based update can be expressed in $\mathcal{AU}$. We assume that all the events are deterministic. This is a natural assumption, since the actions of the language $\mathcal{AU}$ are always deterministic. Under the assumption of determinism, weak and predictive explanations are the same.

Let $E = \{e_1, \ldots, e_m\}$ be the set of events. The corresponding action theory has m actions $A_1, \ldots, A_m$. The behavioral part of the domain is determined by the events in the following manner. For each event e_j and interpretation I, if x_i is true in $e_j(I)$ we have the effect proposition

$$A_j \textsf{ causes } x_i \textsf{ if } \left(\bigwedge_{x_k \in I} x_k \wedge \bigwedge_{x_k \notin I} \neg x_k \right)$$

[1] In the original Boutilier's definition, the update is inconsistent if there is an $I \in Mod(T)$ such that $Prog(I, P)$ is empty. For simplicity, we do not consider this case.

otherwise the effect proposition to add is

$$A_j \textsf{ causes } \neg x_i \textsf{ if } \left(\bigwedge_{x_k \in I} x_k \wedge \bigwedge_{x_k \notin I} \neg x_k \right)$$

The behavioral part D_B of the domain description is the union of all these effect propositions, for each event e, interpretation I and atom x_i.

The actual part is composed by two propositions only:

$$D_A = \{T \textsf{ holds at } 0,\ P \textsf{ holds at } 1\}$$

The historical part of the domain description is empty: $D_H = \emptyset$. The ordering $\preceq_A$ is defined as follows.

Definition 10. The ordering $\preceq_A$ is defined as: $(\sigma_0, \Phi_0, H_0) \preceq (\sigma_1, \Phi_1, H_1)$ if and only if

1. $\sigma_0 = \sigma_1$
2. $\Phi_0 = \Phi_1$
3. it holds e_0 happens at $0 \in H_0$, e_1 happens at $0 \in H_1$, and $e_0 \leq_{\sigma_0} e_1$.

About the correctness of this definition, the following theorem relates the entailment in $\mathcal{AU}$ and the inference of $*_{ABD}$.

Theorem 3. *For each 3-tuple of propositional formulas T, P, and Q, it holds $T *_{ABD} P \models Q$ if and only if $D \models_A Q$ holds at 1 (using the ordering $\preceq_A$), where D is the domain description defined above.*

4 Related Work

In this section we compare our approach with others that use the similarities between reasoning about actions and update. The approach that is most similar to ours is the Possible Causes Approach (PCA) proposed by Li and Pereira [8]. Although it is based on similar principles, is different from our proposal in two aspects. First of all, update is not embedded into the temporal logic. Rather, given a domain description and an update, the aim of PCA is to consistently incorporate the update in the domain. In our semantics, the updating formula is expressed as a proposition of the domain description.

Another difference regards the KM postulates. Li and Pereira's approach does not respect the basic principle that models of the initial knowledge base must be updated separately. This implies, for example, that Winslett's update cannot be easily expressed into Li and Pereira's formalism.

The KM postulates, as our framework, provide a generalization of Winslett's approach to update. Due to the lack of space, we cannot make a detailed comparison between these two frameworks. Let us only say that, while KM postulates only generalizes Winslett's semantics, our approach is more general, as other update methods can be encoded in it.

Another approach which is somewhat related to use is due to Peppas [10], which shows how epistemic entrenchment (a well-known notion in belief revision) can be used in the update framework as well.

The relationship between belief update and reasoning about actions have been also analyzed by del Val and Shoham. The key idea of their work can be summarized by the following quotation [4].

> The initial database is taken to describe a particular situation, and the update formula is taken to describe the effect of a particular action. A formal theory of action is then used to infer facts about the result of taking the particular action in the particular situation [...]. Finally, anything inferred about the resulting situation can be translated back to the timeless framework of belief update.

Their framework is used to derive a semantics for belief update. In order to do this, they translate a specific initial base and an update into a specific theory of actions. A single update is translated into a single action. From this point of view, our framework is exactly the opposite: we derive a semantics of a possibly inconsistent theory of actions by employing the idea of update. An update is indeed a fact that holds in some time point, and changes are caused by actions.

Winslett's update, as it was initially defined [13], was used in a similar way: the initial knowledge base is the state of the world at a certain time point, and the update is the effect of a complex action. The result of Winslett's update is used to determine the state of the world after that the action is performed. This way the frame problem is solved, if the effect of the action is a conjunction of literals.

In this context, an action is formalized by an update: as have shown, updates can be in turns formalized as the result of a number of simpler actions. Following this approach, and using Winslett's update, a complex action is considered to be equivalent to a set of elementary actions, each changing the truth value of a variable. In the case of non-disjunctive actions, having yet another solution from the frame problem is not really intersting at this point, as many other solutions already exist [11,12]. The point of view offered by this approach may be of interest in the case of disjunctive actions.

5 Discussion

In this paper we have introduced the language $\mathcal{AU}$, that formalizes scenarios in which actions may take place, and of which the agent has no knowledge. The language $\mathcal{AU}$ is essentially a dialect of the language $\mathcal{A}$ for reasoning about actions [6] with narratives.

This formalism is also useful for the field of belief update. Indeed, the definitions given by Boutilier, Katsuno and Mendelzon, and Winslett can be easily encoded in $\mathcal{AU}$. This provides a way for comparing the semantics of these formalisms. For example, Winslett's update can be expressed in $\mathcal{AU}$ by assuming that the change that caused the updating formula to hold in a successive state is due to the effect of a sequence of simple actions, each causing the truth value of a variable to change. The actions we used to formalize Boutilier's abduction-based update are more complicated (i.e. involving more that one variable).

Regarding Boutilier's update, we also note that the translation given here is exponential-size. This can be explained by observing that Boutilier's events may be arbitrarily complicated. In real scenarios, there should be a simple rule to determine the effect of events.

The language $\mathcal{AU}$ allows the integration of many features that are recognized by many researchers as fundamental in expressive theories of belief update.

1. It is possible to express which changes may take place (for example, the fact that Fred cannot become alive, once he is dead is formalized by the absence of actions that makes Fred alive, if he is dead).
2. In some situation, the observation at time 1 leads to modify our knowledge about time 0. This can be expressed in $\mathcal{AU}$.
3. It is possible to express multiple observations at different time points (iterated belief update).

An interesting feature of $\mathcal{AU}$ is that it allows inference of happens statements: a domain description D implies an happens proposition A happens at t if and only if the A happens at t is contained in all the models of D. This issue is of course trivial in classical action description languages, in which an happens proposition is implied by a domain description if and only if it is in the domain. In $\mathcal{AU}$ (with the update semantics) it is possible to infer that an action took place at time t if A happens at t is in all the models of the domain description.

Finally, the principles of $\mathcal{AU}$ can be used to extend the system BReLS [9] to deal with complex actions. BReLS has been introduced to deal with domains in which both revision and update are necessary. The semantics of BReLS are based on the principle of combining a measure of reliability of sources of information with the likeliness of events. The way in which events are formalized is so far quite simple: the only possible actions are those setting the value of a variable to a given value (true or false). The user can decide the likeliness of such actions, but cannot define more complex actions. Syntactically, this is done with a statement like `change(i)` : l, which means that the penalty (degree of unlikeliness) of the literal l becoming true is i. Extending the syntax is quite straightforward: `change(i)` : A means that the penalty of the action A to take place is i. The extension of the semantics is also quite easy: a model is composed by a set of static models (propositional interpretations), one for each time point, and a set of actions for any pair of consecutive time points. This model is consistent with the domain description if and only if the static model at time $t+1$ is the result of applying the actions relative to the pair $\langle t, t+1 \rangle$ to the static model of time t. The ordering between models can also be obtained by combining the degree of reliability of sources with the penalty associated to changes, as usual. Extending the implemented algorithms, on the other hand, seems to be not as simple.

References

1. C. Baral, M. Gelfond, and A. Provetti. Representing actions: laws, observations, and hypothesis. *J. of Logic Programming*, 31, 1997.
2. C. Boutilier. Generalized update: belief change in dynamic settings. In *Proc. of IJCAI'95*, pages 1550–1556, 1995.
3. C. Boutilier. Abduction to plausible causes: an event-based model of belief update. *Artificial Intelligence*, 83:143–166, 1996.
4. A. del Val and Y. Shoham. Deriving properties of belief update from theories of action. *J. of Logic, Language and Information*, 3:81–119, 1994.
5. N. Friedman and J. Y. Halpern. A knowledge-based framework for belief change, part II: Revision and update. In *Proc. of KR'94*, pages 190–200, 1994.
6. M. Gelfond and V. Lifschitz. Representing action and change by logic programs. *J. of Logic Programming*, 17:301–322, 1993.
7. H. Katsuno and A. O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52:263–294, 1991.
8. R. Li and L. Pereira. What is believed is what is explained. In *Proc. of AAAI'96*, pages 550–555, 1996.
9. P. Liberatore and M. Schaerf. BReLS: A system for the integration of knowledge bases. In *Proc. of KR 2000*, pages 145–152, 2000.
10. P. Peppas. PMA epistemic entrenchment: The general case. In *Proc. of ECAI'96*, pages 85–89, 1996.
11. E. Sandewall. *Features and Fluents.* Oxford University Press, 1994.
12. M. Thielscher. Introduction to the fluent calculus. *ETAI*, 3, 1998.
13. M. Winslett. Reasoning about actions using a possible models approach. In *Proc. of AAAI'88*, pages 89–93, 1988.
14. M. Winslett. *Updating Logical Databases.* Cambridge University Press, 1990.