Refinement of intentions

Andreas Herzig[1], Laurent Perrussel[1], Zhanhao Xiao[1,2], Dongmo Zhang[2]

[1] University of Toulouse, IRIT, France
[2] Dept. of CS, Western Sydney University, Australia

Abstract. According to Bratman's influential theory, intentions should be viewed as more or less detailed plans. Such plans are typically made up of high-level actions that cannot be executed directly: they have to be progressively refined to basic actions in order to be executable. Inspired by Shoham's database perspective, we view basic and high-level intentions as organized in a database that specifies the temporal intervals within which the corresponding actions have to be performed. This database also contain beliefs about the environment and its change. Actions and events are defined in terms of their pre- and postconditions. Higher intentions can be refined by choosing several possible lower-level intentions to implement them. Based on the refinement of intentions, by adding those lower-level intentions, the database may be further refined. A kind of means-end relation on intentions follows the refinement of intentions, alias instrumentality relation, linking higher- and lower-level intentions.

1 Introduction

Bratman highlighted the fundamental role of an agent's future-directed intentions: they are *high-level plans* to which the agent is committed [6, 8]. Such high-level plans typically cannot be executed directly: they have to be *refined* as time goes by, resulting in more and more elaborate plans. At the end of the refinement process there are *basic actions*, which are the actions the agent can perform intentionally. The aim of this paper is to provide a logical analysis of this refinement process.

The refinement operation of intentions is of fundamental importance and should be central in any theory of autonomous agents as pointed out by Rao and Georgeff [18]. The literature on BDI logics only contains very few such theories [23, 20, 3, 14, 17]. The refinement of intentions is notably missing in Cohen and Levesque's logic [9] and in Shoham's *database perspective* [21, 15, 25]. This latter perspective considers intentions as a list of actions and beliefs indexed by time: this rather simple perspective is actually a promising basis for a logical analysis of intentions. Hereafter, we commit to this perspective as starting point.

Up to now Shoham's framework only contains basic actions; we first introduce *high-level actions* that cannot be executed directly. Second, we add environment actions, which are not only nature's actions, but also actions of other agents; taking the perspective of the planning agent we call such actions *events*.

While high-level actions have arbitrary boolean formulas as pre- and postconditions, we consider (just as Shoham does) that basic actions are STRIPS-like. We make the same hypothesis for basic events. In our framework the preconditions of actions and events play different roles: as usual in reasoning about actions, when the precondition of an action is true then the action is execut*able*; in contrast, when the precondition of an event is true then —according to the planning agent's beliefs—the event is execut*ed*. This hypothesis is necessary in order to allow our agent to include the reactions of the environment in her plans. For example, when planning my JELIA submission I suppose that when I click the 'upload' button then the upload event is triggered. So the planning agent is proactive while, from her perspective, the environment is reactive.

Hereafter, we focus on the central role that is played by the refinement operation of intentions: we define the proper refinement of intentions database by adding lower-level intentions to refine the high-level intention. While Bratman [7] pointed out the fundamental significance of *instrumentality* on intentions, it is involved in the refinement of intentions: the refining lower-level intentions should be instrumental for the refined high-level intention i, that is as a decomposition of i.

Our paper is organized as follows. Section 2 recalls Shoham's database perspective and its shortcoming and motivates our main concepts. Section 3 defines action theories, event theories, and database and gives their semantics. Section 4 defines refinement of intentions. In Section 5 we define refinement of intention database as well as their completion. We defined instrumentality based on refinement of intentions in Section 6. Section 7 discusses related work and Section 8 concludes.

2 Shoham's database perspective

Cohen and Levesque's commitment-based logic provided a seminal logical modeling of Bratman's theory [9]. While this approach is much cited, it is fair to say that it is rather complicated. None of the BDI logics that were introduced subsequently—starting with [18]—fully adapted Cohen and Levesque's definition of intention.

Almost 20 years later, Shoham argued for a simpler approach that he baptized the *database perspective* [21, 22]. The central idea is to organize beliefs and intentions, which are intentions-to-do, in two temporal databases. The belief base is a set of time-indexed propositional variables p_t, read "p is true at t", while the intention base is a set of time-indexed actions a_t, read "the agent intends to do action a at time t". Shoham also sketched an operation of joint revision of the two databases where belief change can trigger intention change and vice versa. Shoham's perspective was subsequently worked out by Icard *et al.* [15], who provided a semantics for belief-intention databases in terms of paths as well as AGM-like postulates for the joint revision of beliefs and intentions. Van Zee *et al.* [25, 24] recently proposed a modification of Icard *et al.*'s logic, moving to a semantics of CTL*-like tree structures and using time-indexed modalities.

The common point between all these contributions is the assumption that intentions are organized in a flat way: there is no notion of high- and low-level intentions. This is clearly a major shortcoming w.r.t. Bratman's theory [8]. Intentions might be defined at high level and next refined in order to obtain executable actions. As already pointed out by [19] in the planning context, defining actions too early is expensive and rigid and leads few years later to consider Hierarchical Tasks Networks for planning [11]. In that context, revising the intentions base consists at first in refining intentions.

The database approach up to now did not also cater for environment actions, alias events. For that reason, the existing approaches—despite their STRIPS-like action theories—fail to solve the frame problem: contrarily to what one may expect, the agent's beliefs at time point t together with her action a at t do not determine her beliefs at $t{+}1$ (only belief related to the effects of a can be determined). In this paper we go beyond the initial Shoham's database perspective by not only considering the actions of the agent under concern, but also the environment's actions. Taking the perspective of the planning agent, we call the latter 'events' and the former just 'actions'. We only consider basic events, each of which takes one time unit. We allow for several events to occur simultaneously, allowing thus for environments with multiple agents. In other words, we have a planning agent who is proactive and who has beliefs about a reactive environment: with respect to her beliefs, she believes the environment will react accordingly.

3 From actions to intentions

The first thing we have to do is to define *dynamic theories* and *beliefs, events and intentions database* and to equip them with a semantics.

3.1 Dynamic theories

Let $\mathsf{Evt}_0 = \{e, f, \ldots\}$ be a set of basic events and $\mathsf{Act}_0 = \{a, b, \ldots\}$ a set of basic actions. Basic events and basic actions take one time unit. Basic actions can be directly executed by the planning agent. The set Act_0 is contained in the set of all actions $\mathsf{Act} = \{\alpha, \beta, \ldots\}$ which also contains non-basic, high-level actions. The set of propositional variables is $\mathbb{P} = \{p, q, \ldots\}$. The language of boolean formulas built on $\mathbb{P}$ is denoted by $\mathscr{L}_{\mathbb{P}}$. We suppose that the sets $\mathbb{P}$, Evt_0, and Act are all finite. The behavior of actions and events is described by dynamic theories.

Definition 1. ***(Dynamic theory)*** *A dynamic theory is a tuple $\mathcal{D} = \langle pre, post \rangle$ with $pre, post : \mathsf{Act} \cup \mathsf{Evt}_0 \longrightarrow \mathscr{L}_{\mathbb{P}}$, such that the effects of basic actions and events are conjunctions of literals: there are functions $\mathit{eff}^+, \mathit{eff}^- : \mathsf{Act}_0 \cup \mathsf{Evt}_0 \longrightarrow 2^{\mathbb{P}}$ such that for every $x \in \mathsf{Act}_0 \cup \mathsf{Evt}_0$,*

$$\models post(x) \leftrightarrow \Big(\bigwedge_{p \in \mathit{eff}^+(x)} p \Big) \land \Big(\bigwedge_{p \in \mathit{eff}^-(x)} \neg p \Big) \tag{1}$$

where $\mathit{eff}^+(x) \cap \mathit{eff}^-(x) \neq \emptyset$.

In other words, basic actions and events are STRIPS-like. For example, for the basic empty action wait we have $\mathit{pre}(\mathsf{wait}) = \top$ and $\mathit{eff}^+(\mathsf{wait}) = \mathit{eff}^-(\mathsf{wait}) = \emptyset$. The functions $\mathit{pre}, \mathit{post}, \mathit{eff}^+$ and eff^- are naturally extended to sets, e.g. $\mathit{pre}(X) = \bigwedge_{x \in X} \mathit{pre}(x)$ for $X \subseteq \mathsf{Act}_0 \cup \mathsf{Evt}_0$.

Definition 2. ***(Coherence)*** *A dynamic theory* $\mathcal{D}$ *is coherent if and only if for every basic action* $a \in \mathsf{Act}_0$ *and event set* $E \subseteq \mathsf{Evt}_0$*, if* $\mathit{pre}(\{a\} \cup E)$ *is consistent then* $\mathit{post}(\{a\} \cup E)$ *is consistent.*

Notice that the exponential number of pairs of basic action and set of events entails a potential exponential number of consistency tests. To avoid this issue, we introduce a formula checking coherence with a polynomial length of dynamic theory and propositional variables.

Proposition 1. *A dynamic theory* $\mathcal{D}$ *is coherent iff the following formula, denoted by* $\mathit{Coh}(\mathcal{D})$*, is valid:*

$$\bigwedge_{\substack{e \in \mathsf{Evt}_0, x \in \mathsf{Act}_0 \cup \mathsf{Evt}_0, \\ \mathit{eff}^+(e) \cap \mathit{eff}^-(x) \neq \emptyset \text{ or } \mathit{eff}^-(e) \cap \mathit{eff}^+(x) \neq \emptyset}} \mathit{pre}(e) \land \mathit{pre}(x) \to \bot$$

Proof. "⇒": Suppose dynamic theory $\mathcal{D}$ is coherent and $\mathit{post}(\{a\} \cup E)$ is inconsistent. Because all basic actions and events have a consistent postcondition in form of a conjunction of literals, Only a pair of an action or event $x \in \{a\} \cup E$ and an event $e \in E$ such that one has a positive effect on propositional variable p and the other has a negative effect on p, would make $\mathit{post}(\{x, e\})$ inconsistent and further $\mathit{post}(\{a\} \cup E)$ inconsistent. According to the definition of coherence their jointly precondition $\mathit{pre}(\{x, e\})$ is inconsistent. Thus $\mathit{Coh}(\mathcal{D})$ is valid.

"⇐": Suppose $\mathit{Coh}(\mathcal{D})$ is valid and suppose there exists some action a and event set E such that $\mathit{pre}(\{a\} \cup E)$ is consistent while $\mathit{post}(\{a\} \cup E)$ is inconsistent. As $\mathit{post}(a)$ and $\mathit{post}(E)$ can be rewritten into a conjunction of literals as formula (1), there must exist a pair of p and $\neg p$ occurring in $\mathit{post}(\{a\} \cup E)$. Then there are $x, y \in \{a\} \cup E$ such that $x \neq y$ and $p \in \mathit{eff}^+(x) \cap \mathit{eff}^-(y)$. Due to $\mathit{Coh}(\mathcal{D})$, we have $\mathit{pre}(x) \land \mathit{pre}(y) \to \bot$, which entails $\mathit{pre}(a) \land \mathit{pre}(E) \to \bot$, contradicting that $\mathit{pre}(\{a\} \cup E)$ is consistent. This ends the proof.

Let us now prove that checking coherence is co-NP-complete.

Theorem 1 (Complexity of Coherence). *Given any a dynamic theory* $\mathcal{D}$*, to decide whether* $\mathcal{D}$ *is coherent is co-NP-complete.*

Proof. As the formula $\mathit{Coh}(\mathcal{D})$ has length $O(|\mathbb{P}| \times \mathit{len}(\mathcal{D}))$, deciding coherence is in co-NP.

To establish hardness, consider a propositional formula φ. Let $\mathsf{Act}_0 = \{a, \mathsf{wait}\}$, $\mathsf{Evt}_0 = \{e\}$ and let $\mathcal{D}$ be a dynamic theory with $\mathit{pre}(a) = \mathit{pre}(e) = \varphi$, $\mathit{post}(a) = p$ and $\mathit{post}(e) = \neg p$. As $\mathit{post}(a) \land \mathit{post}(e)$ is inconsistent, φ is inconsistent iff $\mathcal{D}$ is coherent. It follows that deciding coherence is co-NP-hard.

Example 1. Alice has a high-level action **buy** of buying a movie ticket and the basic actions of buying a ticket online buyWeb, going to the cinema gotoC, and buying a ticket at the cinema counter buyC. Moreover, there is an event of the website delivering the electronic ticket $\mathsf{deliver}$. Let the propositional variables $\mathsf{PaidWeb}$, Ticket and InC respectively stand for "Alice has paid online", "Alice has a ticket" and "Alice is in the cinema". The actions and events obey the following coherent dynamic theory $\mathcal{D}$:

$$
\begin{array}{ll}
pre(\mathsf{wait}) = \top & post(\mathsf{wait}) = \top \\
pre(\mathsf{buyWeb}) = \top & post(\mathsf{buyWeb}) = \mathsf{PaidWeb} \\
pre(\mathsf{gotoC}) = \top & post(\mathsf{gotoC}) = \mathsf{InC} \\
pre(\mathsf{buyC}) = \mathsf{InC} & post(\mathsf{buyC}) = \mathsf{Ticket} \\
pre(\mathbf{buy}) = \top & post(\mathbf{buy}) = \mathsf{Ticket} \\
pre(\mathsf{deliver}) = \mathsf{PaidWeb} \land \neg\mathsf{Delivered} & post(\mathsf{deliver}) = \mathsf{Ticket} \land \mathsf{Delivered}
\end{array}
$$

3.2 Beliefs, Events and Intentions Database

We extend database perspective with events: an agent's database then contains her intentions plus her beliefs about the states and event occurrences. Her beliefs about the latter two may be incomplete.

Occurrence of an event $e \in \mathsf{Evt}_0$ at time point t is noted (t, e). We also want to be able to talk about the non-occurrence of events. To that end we define the set $\overline{\mathsf{Evt}_0} = \{\bar{e} : e \in \mathsf{Evt}_0\}$ of event complements. Non-occurrence of e is noted $(t, \bar{e})$.

An *intention* is a triple $i = (t, \alpha, d) \in \mathbb{N}^0 \times \mathsf{Act} \times \mathbb{N}$ with $t < d$. It represents that the agent wants to perform α in the time interval $[t, d]$: action α should start after t and end before d. When $\alpha \in \mathsf{Act}_0$ then i is a *basic* intention. We use $i, j, \ldots$ to denote intentions and $J, J_1, \ldots$ to denote sets thereof.

Definition 3 (Belief, Event, Intention Database). *A belief, event, intention database $\mathcal{D}b$ (BEI Database) is a finite set*

$$\mathcal{D}b \subseteq (\mathbb{N}^0 \times \mathscr{L}_{\mathbb{P}}) \cup (\mathbb{N}^0 \times \mathsf{Evt}_0) \cup (\mathbb{N}^0 \times \overline{\mathsf{Evt}_0}) \cup (\mathbb{N}^0 \times \mathsf{Act} \times \mathbb{N}).$$

We often partition *BEI* database into belief base, event base and intention base by means of the following functions:

$$
\begin{aligned}
\mathcal{B}(\mathcal{D}b) &= \mathcal{D}b \cap (\mathbb{N}^0 \times \mathscr{L}_{\mathbb{P}}) \\
\mathcal{E}(\mathcal{D}b) &= \mathcal{D}b \cap (\mathbb{N}^0 \times (\mathsf{Evt}_0 \cup \overline{\mathsf{Evt}_0})) \\
\mathcal{I}(\mathcal{D}b) &= \mathcal{D}b \cap (\mathbb{N}^0 \times \mathsf{Act} \times \mathbb{N})
\end{aligned}
$$

Given an intention $i = (t, \alpha, d)$, we define $\mathsf{end}(i) = d$. For a database $\mathcal{D}b$, we let $\mathsf{end}(\mathcal{D}b)$ be the greatest time point occurring in $\mathcal{D}b$. This is well defined as database are finite. When $\mathcal{E}(\mathcal{D}b) = \mathcal{I}(\mathcal{D}b) = \emptyset$ then we set $\mathsf{end}(\mathcal{D}b) = 0$.

Example 2. Let us reconsider our previous example. Alice's initial *BEI* database only contains $\mathcal{D}b_c = \{(0, \mathbf{buy}, 2)\}$, i.e., Alice intends to buy a movie ticket within the temporal interval $[0, 2]$. According to the dynamic theory $\mathcal{D}$ there are two ways to achieve this intention: either perform buyWeb at 0 and then wait (believing event $\mathsf{deliver}$ will occur at 1); or perform gotoC at 0 and buyC at 1.

3.3 Semantics

The semantics of dynamic theories and *BEI* databases is in terms of *paths*. A path defines for each time point which propositional variables are true, which basic actions the agent will perform, and which events will occur.

Definition 4 (path). *A* $\mathcal{D}$-path *is a triple* $\pi = \langle V, H, D\rangle$ *with* $V : \mathbb{N}^0 \longrightarrow 2^{\mathbb{P}}$, $H : \mathbb{N}^0 \to 2^{\mathsf{Evt}_0}$*, and* $D : \mathbb{N}^0 \to \mathsf{Act}_0$*.*

A path π associates to every time point t a valuation $V(t)$ (alias a state), a set of basic events $H(t)$ happening at t, and a basic action $D(t)$ that the agent does at t. *BEI* database is interpreted given a background dynamic theory.

Definition 5 ($\mathcal{D}$-model). *A* model of $\mathcal{D}$*, or* $\mathcal{D}$-model*, is a path* $\pi = \langle V, H, D\rangle$ *such that for every time point* $t \in \mathbb{N}^0$*,*

$$\mathit{eff}^+\big(H(t)\cup\{D(t)\}\big) \cap \mathit{eff}^-\big(H(t)\cup\{D(t)\}\big) = \emptyset \tag{2}$$

and

$$\begin{aligned} V(t{+}1) &= \big(V(t) \cup \mathit{eff}^+\big(H(t)\cup\{D(t)\}\big)\big) \setminus \mathit{eff}^-\big(H(t)\cup\{D(t)\}\big) \\ H(t) &= \{e \in \mathsf{Evt}_0 \mid V(t) \models \mathit{pre}(e)\} \\ D(t) &\in \{a \in \mathsf{Act}_0 \mid V(t) \models \mathit{pre}(a)\} \end{aligned}$$

So a path is a $\mathcal{D}$-model when (1) the action $D(t)$ to be performed and the events $H(t)$ to happen are consistent; (2) a minimal change condition is satisfied: the state at t and the basic action and events occurring at t determine the state at $t{+}1$; (3) the environment is reactive: event e occurs *iff* $\mathit{pre}(e)$ is true; (4) the agent is autonomous: if $\mathit{pre}(a)$ is true then the agent *can* perform a, but does not necessarily do so. Note that when $\mathcal{D}$ is coherent then the constraint (2) in Definition 5 is always satisfied.

We are now ready to define the satisfaction relation $\Vdash_{\mathcal{D}}$ between a path and an intention or a database.

Definition 6 (Satisfaction of an intention). *Intention* $i = (t, \alpha, d)$ *is satisfied at a path* $\pi = \langle V, H, D\rangle$*, noted* $\pi \Vdash_{\mathcal{D}} i$*, if there exist* t', d' *such that* $t{\le}t'{<}d'{\le}d$*,* $V(t') \models \mathit{pre}(\alpha)$*,* $V(d') \models \mathit{post}(\alpha)$*, and* $\alpha \in \mathsf{Act}_0$ *implies* $D(t') = \alpha$*.*

An intention $i = (t, \alpha, d)$ is satisfied at π if the intended action α can start at some $t' \geq t$ where the precondition of α holds and can end at some $d' \leq d$ where the postcondition of α holds. Moreover, when α is basic then α is indeed performed at the starting point t' according to the ‘do’-function D of π.

Definition 7 (Satisfaction of a *BEI* database). *A $\mathcal{D}$-model $\pi = \langle V, H, D \rangle$ is a $\mathcal{D}$-*model *of $\mathcal{D}b$, noted $\pi \Vdash_{\mathcal{D}} \mathcal{D}b$, if*

- *for every $(t, \varphi) \in \mathcal{B}(\mathcal{D}b)$: $V(t) \models \varphi$;*
- *for every $(t, e) \in \mathcal{E}(\mathcal{D}b)$: $e \in H(t)$;*
- *for every $(t, \bar{e}) \in \mathcal{E}(\mathcal{D}b)$: $e \notin H(t)$;*
- *for every $i \in \mathcal{I}(\mathcal{D}b)$: $\pi \Vdash_{\mathcal{D}} i$.*

So when $\pi \Vdash_{\mathcal{D}} \mathcal{D}b$ then the agent's beliefs about the state $\mathcal{B}(\mathcal{D}b)$ and about the (non-)occurrence of events $\mathcal{E}(\mathcal{D}b)$ are correct w.r.t. π, and all intentions in $\mathcal{I}(\mathcal{D}b)$ are satisfied on π. A database $\mathcal{D}b$ is $\mathcal{D}$-*satisfiable* when $\mathcal{D}b$ has a $\mathcal{D}$-model.

$\mathcal{D}b$ is a $\mathcal{D}$-*consequence of* $\mathcal{D}b'$, noted $\mathcal{D}b' \models_{\mathcal{D}} \mathcal{D}b$, if every $\mathcal{D}$-model of $\mathcal{D}b'$ is also a $\mathcal{D}$-model of $\mathcal{D}b$. When $\mathcal{D}b$ is a singleton $\{i\}$ we write $\mathcal{D}b' \models_{\mathcal{D}} i$ instead of $\mathcal{D}b' \models_{\mathcal{D}} \{i\}$.

Proposition 2. *For any dynamic theory $\mathcal{D}$, the satisfiability problem and consequence problem of database are decidable.*

Proof. (sketch) Because there is a finite number of propositional variables in $\mathbb{P}$, there is at most $2^{|\mathbb{P}|}$ states with different valuations. Observe that valuation at $t{+}1$ only depends of the valuation at t and the action performed at t, paths become Markov chains. There is going to be a loop of length at most $2^{|\mathbb{P}|} \times |\mathsf{Act}_0|$. Consequently there is a finite number of paths and the satisfiability problem and the consequence problem can be decided by checking a finite number of models.

If dynamic theory $\mathcal{D}$ is actually coherent, checking whether a *BEI* database is $\mathcal{D}$-satisfiable can be performed by translating the dynamic theory and database into a propositional formula where literals are time-indexed symbols. Using a similar technique, it is also easy to check the consequence problem $\mathcal{D}b' \models_{\mathcal{D}} \mathcal{D}b$. When it comes to arbitrary dynamic theories, such translations become unavailable because the constraint (2) can not be guaranteed to be satisfied and we have to consider a model with infinite time points.

4 Refining an intention

A high-level intention cannot be executed directly by the agent: it can only be refined into lower-level intentions, until basic intentions are produced. For example, my high-level intention i to submit a paper to JELIA'16 before its deadline Jun. 30 is decomposed into intention i_1 to log in Easychair before Jun. 30 and then intention i_2 to upload a paper as a PDF file; etc.

Definition 8. *We say that an intention $i \in \mathcal{D}b$ is* refined *in $\mathcal{D}b$ when $\mathcal{D}b \setminus \{i\} \models_{\mathcal{D}} i$; else we say that i has not been refined yet.*

Intuitively, to refine an intention i means to add a minimal set of new intentions J to the database which, together with other intentions but i, suffice to guarantee satisfaction of i.

Definition 9 (Intention refinement). *Let $\mathcal{D}b$ be a database. Let $i \in \mathcal{I}(\mathcal{D}b)$ and let J be some set of intentions. Then i is refinable to J in $\mathcal{D}b$, noted $\mathcal{D}b \models_{\mathcal{D}} i \lhd J$, iff*

1. *there is no $j \in J$ such that $\mathcal{D}b \models_{\mathcal{D}} j$;*
2. *$\mathcal{D}b \cup J$ has a $\mathcal{D}$-model;*
3. *$(\mathcal{D}b \cup J) \setminus \{i\} \models_{\mathcal{D}} i$;*
4. *$(\mathcal{D}b \cup J') \setminus \{i\} \not\models_{\mathcal{D}} i$ for every $J' \subset J$;*
5. *$\mathsf{end}(J) \leq \mathsf{end}(i)$.*

Intuitively, Condition 1 states that refinement consists of adding new intentions, Condition 2 enforces consistent refinement, Condition 3 states that new added intentions must satisfy the refined intention i, Condition 4 enforces minimality of refinement. The last condition checks that time constraints are actually satisfied.

Example 3. Suppose Alice's current database is $\mathcal{D}b_c$ and she decides to buy a ticket online. Let $i = (0, \mathbf{buy}, 2)$ and $j = (0, \mathsf{buyWeb}, 1)$. We have that $\mathcal{D}b_c \not\models_{\mathcal{D}} j$ and $(\mathcal{D}b_c \cup \{j\}) \setminus \{i\} \models_{\mathcal{D}} i$ and $\mathcal{D}b_c \setminus \{i\} \not\models_{\mathcal{D}} i$. Therefore satisfying j can guarantee the satisfaction of i: we have $\mathcal{D}b_c \models_{\mathcal{D}} i \lhd \{j\}$.

The following proposition shows that refinement is minimal:

Proposition 3. *Let $\mathcal{D}b \models_{\mathcal{D}} i \lhd J$. Then:*

1. *$\mathcal{D}b$ has a $\mathcal{D}$-model;*
2. *$J \cap \mathcal{I}(\mathcal{D}b) = \emptyset$ and $i \notin J$;*
3. *there is no $J' \subset J$ such that $\mathcal{D}b \models_{\mathcal{D}} i \lhd J'$;*
4. *$J = \emptyset$ iff i is already refined in $\mathcal{D}b$.*

Proof. Let $\mathcal{D}b \models_{\mathcal{D}} i \lhd J$.

1. $\mathcal{D}b$ must be $\mathcal{D}$-satisfiable because of Condition 1 of Def. 9 (and also because of Condition 2).
2. Suppose $j \in J \cap \mathcal{I}(\mathcal{D}b)$. Let $J' = J \setminus \{j\}$. Then $\mathcal{D}b \cup J'$ equals $\mathcal{D}b \cup J$, and as $(\mathcal{D}b \cup J) \setminus \{i\} \models_{\mathcal{D}} i$ we also have $(\mathcal{D}b \cup J') \setminus \{i\} \models_{\mathcal{D}} i$. The latter violates condition 4 of minimality of Def. 9.
3. Suppose there is a $J' \subset J$ such that $\mathcal{D}b \models_{\mathcal{D}} i \lhd J'$. Then by the definition of refinement we would have $(\mathcal{D}b \cup J') \setminus \{i\} \models_{\mathcal{D}} i$, contradicting $\mathcal{D}b \models_{\mathcal{D}} i \lhd J$.
4. $J = \emptyset$ implies that i is already refined in $\mathcal{D}b$, i.e., $\mathcal{D}b \setminus \{i\} \models_{\mathcal{D}} i$, by condition 3 of Def. 9. The other way round, if $\mathcal{D}b \setminus \{i\} \models_{\mathcal{D}} i$ then all the conditions for $\mathcal{D}b \models_{\mathcal{D}} i \lhd J$ will be satisfied.

Proposition 4. *Given a dynamic theory $\mathcal{D}$, to decide $\mathcal{D}b \models_{\mathcal{D}} i \lhd J$ is decidable.*

Proof. By Proposition 2, $\mathcal{D}b \models_{\mathcal{D}} i \lhd J$ can be decided by checking a finite number of models.

5 Proper refinement and completion of a *BEI* database

Let us show how the refinement relation is used for refining a database. Intuitively, a given *BEI* database $\mathcal{D}b$ can be refined by means of the following procedure:

1. select an intention i of $\mathcal{D}b$;
2. find a set of intentions J such that $\mathcal{D}b \models i \lhd J$;
3. add J to $\mathcal{D}b$.

The refinement relation between intentions can be extended to database:

Definition 10 (Database refinement). *A database $\mathcal{D}b$ is one-step refinable to $\mathcal{D}b'$, noted $\mathcal{D}b \lhd \mathcal{D}b'$, iff there is an i in $\mathcal{D}b$ and a nonempty set of intentions J such that $\mathcal{D}b \models i \lhd J$ and $\mathcal{D}b' = \mathcal{D}b \cup J$.*

For $n \geq 0$, we write $\mathcal{D}b \lhd^n \mathcal{D}b'$ when there exist $\mathcal{D}b_1, \ldots, \mathcal{D}b_n$ such that $\mathcal{D}b = \mathcal{D}b_1 \lhd \mathcal{D}b_2, \ldots, \mathcal{D}b_{n-1} \lhd \mathcal{D}b_n = \mathcal{D}b'$. (For $n = 0$ we suppose that $\mathcal{D}b = \mathcal{D}b'$.)

Example 4. Let $\mathcal{D}b'_c$ be the result of adding to $\mathcal{D}b_c$ intention set $\{(0, \mathsf{buyWeb}, 1)\}$ to refine intention $i = (0, \mathbf{buy}, 2)$. We have $\mathcal{D}b_c \lhd \mathcal{D}b'_c$.

The following proposition shows that model of a refined database is also a model of the original database, but the converse does not necessarily hold. Moreover, refinement preserves satisfiability.

Proposition 5. *Let $\mathcal{D}b \lhd^n \mathcal{D}b'$ and $n \geq 1$. Then:*

1. *$\mathcal{D}b' \models \mathcal{D}b$;*
2. *$\mathcal{D}b \not\models \mathcal{D}b'$;*
3. *if $\mathcal{D}b$ is $\mathcal{D}$-satisfiable then $\mathcal{D}b'$ is $\mathcal{D}$-satisfiable;*

Proof. The proof is by induction on n. For $n = 1$ we have $\mathcal{D}b \lhd \mathcal{D}b'$. So there is an $i \in \mathcal{D}b$ and a set of intentions J such that $\mathcal{D}b \models_{\mathcal{D}} i \lhd J$ and $\mathcal{D}b' = \mathcal{D}b \cup J$. The first item is clear because $\mathcal{D}b \subset \mathcal{D}b'$. It also follows that there is at least one such $j \in \mathcal{D}b'$ such that $\mathcal{D}b \not\models_{\mathcal{D}} j$ by condition 1 in Def. 9. So we cannot have $\mathcal{D}b \models_{\mathcal{D}} \mathcal{D}b'$. The third item is trivial due to Condition 2 in Def. 9.

Let $n = k+1, k \geq 1$ and $\mathcal{D}b \lhd^k \mathcal{D}b^k$, $\mathcal{D}b^k \lhd \mathcal{D}b'$. Suppose the three conditions are satisfied for $\mathcal{D}b^k$. Then $\mathcal{D}b' \models_{\mathcal{D}} \mathcal{D}b$ and $\mathcal{D}b' \models_{\mathcal{D}} \mathcal{D}b^k$ because $\mathcal{D}b \subset \mathcal{D}b^k \subset \mathcal{D}b'$. From assumption we get $\mathcal{D}b \not\models_{\mathcal{D}} \mathcal{D}b^k$, if $\mathcal{D}b \models_{\mathcal{D}} \mathcal{D}b'$ then due to $\mathcal{D}b' \models_{\mathcal{D}} \mathcal{D}b^k$ we get $\mathcal{D}b \models_{\mathcal{D}} \mathcal{D}b^k$, which contradicts the assumption. So $\mathcal{D}b \not\models_{\mathcal{D}} \mathcal{D}b'$. If $\mathcal{D}b^k$ is $\mathcal{D}$-satisfiable then $\mathcal{D}b'$ is $\mathcal{D}$-satisfiable. Item 3 is then also proved.

Our notion of refinement is strict (or proper): the models of the refined database are a strict subset of the models of the original database. We otherwise may *complete* the database: intentions which are entailed by a database but not belonging to it are explicitly added.

Definition 11 (Database completion). *An intention i* completes *a BEI database $\mathcal{D}b$ if $\mathcal{D}b \models i$, $i \notin \mathcal{D}b$ and $\mathsf{end}(i) \leq \mathsf{end}(\mathcal{D}b)$. $\mathcal{D}b'$ is a completion of $\mathcal{D}b$ if there exists an i completing $\mathcal{D}b$ such that $\mathcal{D}b' = \mathcal{D}b \cup \{i\}$.*

The completed database is clearly equivalent to the original one. Moreover, as database and actions are finite, only a finite number of completion steps can be made.

Let us now give a sufficient condition for the elaboration of a database.

Proposition 6. *If a $\mathcal{D}$-satisfiable $\mathcal{D}b$ contains a non-basic intention that is not refined then $\mathcal{D}b$ can be either completed or refined.*

Proof. Consider a path $\pi = \langle V, H, D\rangle$ such that $\pi \Vdash_{\mathcal{D}} \mathcal{D}b$. Let $J_0 = \{(s, D(s), s+1) \mid s < \mathsf{end}(i)\}$. So the non-basic and non-refined intention i is not in the set of basic intentions J_0 and J_0 is finite. We have $(\mathcal{D}b \setminus \{i\}) \cup J_0 \models i$. Let $J \subseteq J_0$ be an inclusion-minimal set such that $(\mathcal{D}b \cup J) \setminus \{i\} \models i$. (When $J = \emptyset$ then i has already been refined by item 4 of Proposition 3.) If there is an intention $j \in J$ with $\mathcal{D}b \models_{\mathcal{D}} j$, then $\mathcal{D}b$ can be completed by j (because $j \notin \mathcal{D}b$ due to minimality of J). Otherwise $\mathcal{D}b$ has more than one model $\mathcal{D}b \models_{\mathcal{D}} i \lhd J$ and $\mathcal{D}b \lhd \mathcal{D}b \cup J$.

As we require refinements to be proper, when all $\mathcal{D}$ models of a *BEI* database share the same fragment from 0 to $\mathsf{end}(\mathcal{D}b)$, the database cannot be refined further. So every database can be refined in a finite number of steps.

Proposition 7. *For every satisfiable database $\mathcal{D}b$ there is an $n \leq |\mathcal{I}(\mathcal{D}b)|$ and a satisfiable database $\mathcal{D}b'$ such that $\mathcal{D}b \lhd^n \mathcal{D}b'$ and $\mathcal{D}b'$ cannot be refined.*

Proof. We follow the reasoning in the proof of Proposition 6, refining one by one all those intentions in $\mathcal{D}b$ that are non-basic and have not been refined yet. Each of these refinement steps only adds basic intentions, therefore we terminate after at most $|\mathcal{I}(\mathcal{D}b)|$ steps.

6 Instrumentality from refinement

In the previous sections we have described how higher-level intention could be refined by lower-level intentions. The higher- and lower-level intentions should stand in naturally a kind of means-end relation: the lower-level means contributes to the higher-level end. Indeed this relation is also called *instrumentality relation* [2, 10, 16, 7].

Instrumentality cannot be defined from an action theory alone, for several reasons. First, the time point of action execution matters. For example, let us take up our intention of attending JELIA in November. Suppose I also have to go to the conference host city, Larnaca, in February, for some other reason. The postcondition of that action—to be in Larnaca— entails one of the preconditions of the attending JELIA action. However, as I am going to come back from Larnaca by the end of February, my February intention does not contribute to my November intention. So the former is not necessarily instrumental for the latter. It may actually happen that j is instrumental for i although the postconditions of j are inconsistent with the preconditions of i. Second, the preconditions of the means are typically more demanding than the preconditions of the end; similarly, the postconditions of the means are more detailed than the effects of the end. For example, buying a movie ticket requires cash in your pocket while buying a ticket online may require an account. (The precondition of former action should *a priori* not involve the online account because I can choose another way to buy a ticket.)

Formally, the instrumentality relation relates a refined high-level intention to a set of lower-level intentions, given a background database.

Definition 12 (Instrumentality). *Let $\mathcal{D}b$ be a $\mathcal{D}$-satisfiable database. Let $i \in \mathcal{I}(\mathcal{D}b)$ and let $J \subseteq \mathcal{I}(\mathcal{D}b)$. Then J is* instrumental *for i in $\mathcal{D}b$, noted $\mathcal{D}b \models_{\mathcal{D}} J \gtrdot i$, iff*

1. $\mathcal{D}b \setminus J \not\models_{\mathcal{D}} i$;
2. $(\mathcal{D}b \setminus J) \cup \{j\} \models_{\mathcal{D}} i$ *for every* $j \in J$;
3. $\mathsf{end}(J) \leq \mathsf{end}(i)$.

When $\mathcal{D}b \models_{\mathcal{D}} J \gtrdot i$ then J is a minimal set of intentions satisfying the counterfactual "if J was not in $\mathcal{D}b$ then i would no longer be guaranteed by $\mathcal{D}b$" (Condition 1). Next, the presence of all intentions of J in $\mathcal{D}b$ is mandatory for satisfying i (Condition 2). Moreover, the intentions of J have to be terminated before or together with i (Condition 3).

Example 5. For Alice's database $\mathcal{D}b'_c$ of Example 3, the only set of intentions that is instrumental for $(0, \mathbf{buy}, 2)$ in $\mathcal{D}b'_c$ is $\{(0, \mathbf{buy}, 2), (0, \mathsf{buyWeb}, 1)\}$. That is, $\mathcal{D}b'_c \models_{\mathcal{D}} \{(0, \mathbf{buy}, 2), (0, \mathsf{buyWeb}, 1)\} \gtrdot (0, \mathbf{buy}, 2)$.

Note that it is decidable whether $\mathcal{D}b \models_{\mathcal{D}} J \gtrdot i$, because instrumentality results from checking satisfiability and checking consequence of databases. When $\mathcal{D}b \models_{\mathcal{D}} J \gtrdot i$ then clearly $\mathcal{D}b \models_{\mathcal{D}} i$ (because $i \in \mathcal{D}b$). The followings are some properties of instrumentality.

Proposition 8. *Let $\mathcal{D}b \models_{\mathcal{D}} J \gtrdot i$. Then $i \in J$ and:*

- $J = \{i\}$ *iff* $\mathcal{D}b \setminus \{i\} \not\models_{\mathcal{D}} i$;
- $J = \{i, j\}$ *iff* $\mathcal{D}b \setminus \{i\} \models_{\mathcal{D}} i$, $\mathcal{D}b \setminus \{i, j\} \not\models_{\mathcal{D}} i$, *and* $\mathsf{end}(j) \leq \mathsf{end}(i)$.

Proof. It is easy to check by Definition 12.

Consequently when $\mathcal{D}b \models_{\mathcal{D}} J \gtrdot i$ then J cannot be empty and $i \in J$. We now relate intention refinement to instrumentality: when $\mathcal{D}b \models_{\mathcal{D}} i \lhd J$ then every element of J is instrumental for i in the new database $\mathcal{D}b \cup J$.

Theorem 2. *If $\mathcal{D}b \models_{\mathcal{D}} i \lhd J$ then $\mathcal{D}b \cup J \models_{\mathcal{D}} \{i, j\} \gtrdot i$ for every $j \in J$.*

Proof. Let $\mathcal{D}b \models_{\mathcal{D}} i \lhd J$ and $j \in J$. We have to show that $\mathcal{D}b \cup J \models_{\mathcal{D}} \{i, j\} \gtrdot i$:

1. $(\mathcal{D}b \cup J) \setminus \{i, j\} \not\models_{\mathcal{D}} i$ holds because $\mathcal{D}b \models_{\mathcal{D}} i \lhd J$ implies $(\mathcal{D}b \cup (J \setminus \{j\})) \setminus \{i\} \not\models_{\mathcal{D}} i$.
2. $(\mathcal{D}b \cup J) \setminus J' \models_{\mathcal{D}} i$ holds for every $J' \subset \{i, j\}$:
   - $(\mathcal{D}b \cup J) \setminus \{i\} \models_{\mathcal{D}} i$ follows from $\mathcal{D}b \models_{\mathcal{D}} i \lhd J$;

  – $(\mathcal{D}b \cup J) \setminus \{j\} \models_{\mathcal{D}} i$ holds because $\mathcal{D}b$ contains i and $i \neq j$ by the above Proposition 3.

3. $\mathsf{end}(\{i, j\}) \leq \mathsf{end}(i)$ holds as $\mathcal{D}b \models_{\mathcal{D}} i \lhd J$ implies $\mathsf{end}(J) \leq \mathsf{end}(i)$.

The converse does not hold: instrumentality can not guarantee that the added intentions are new, contradicting item 1 of Definition 9.

7 Related work

Our work is based on Shoham's database perspective on beliefs and intentions [21]. Our contribution differs in two key points. First, in line with Cohen and Levesque [9] who distinguish different notions of goals and intentions, we introduce high-level actions. It enables to consider high-level intentions in a flexible way: in our running example, Alice intends to buy a ticket (**buy**) during some time interval. While in the initial Shoham's framework, Alice can only intend at a specific time point to buy a ticket on line (buyWeb) or at the counter (buyC). Second, by considering in an explicit way the environment we solve the frame problem: our semantics enables Alice to predict that the online system will give a ticket while she waits. Let us stress that this last issue is also unsolved in other existing frameworks related to Shoham's database perspective. For instance, [24] which proposes an AGM-like revision of the belief about action and time in a temporal logic does not bring any solution for the frame problem.

As far as we are aware there are only a few approaches analyzing intention refinement. The closest domain concerns Hierarchical Task Networks (HTN) [11]. HTN have decomposition methods which are nothing but refinements of high-level actions into lower-level actions. The decomposition is primitive: all methods have to be defined by hand by the designer of the planning domain. In our running example, it means that Alice should already set before buying a ticket the decomposition: either buying a ticket online (and wait) or at the counter. In a complex domain it may be difficult to consider all ways in which actions can be decomposed. This contrasts with our approach, which does not require such work: given the action theory and the current database, the refinement relation is defined at a semantic level (satisfiability of logical statements). Our work is also related to [4, 5] which propose an ASP-based system for reconsidering high-level intentions for the unexpected environment change where high-level intentions are predefined in an explicit and sequential way. Again, we avoid this hard coding stage by rooting refinement at a semantic level.

Finally, let us mention Hunsberger and Ortiz [14] work which includes a refinement-like relation and formalized in dynamic logic. It captures a kind of means-end relation on intentions and represents a high-level intention in first order logic by the conjunction of means actions. When elaborating intentions further, abstract means actions will either get a more concrete specification or will be changed by other means actions. They recently presented a dynamic logic-based syntactical operation of intention revision that basically works by the instantiation of variables by constants [17]. However, different from this approach, our approach provides a broader perspective on instrumentality and means-end reasoning where a means only contributes to the end, without guaranteeing it.

8 Conclusion

We have extended Shoham's database view by the fundamental concept of high-level intentions with a flexible duration. The integration of STRIPS-like environment events and high-level actions does not cause undecidability for checking satisfiability and consequence relation. Our *BEI* database view is about high- and low-level intentions that are related by the refinement operator. The instrumentality relation between intentions follows the refinement of intentions.

In this paper, we focus on refinement which is, in some way, a well-founded *BEI* database expansion. More general expansion may lead to unsatisfiable database and raises the issue about withdrawal or revision of intentions.

The proposed notion of instrumentality paves the way for revision of intentions caused by the environment change. When $\mathcal{D}b \models_{\mathcal{D}} J \gg i$ then the end intention i is deeper entrenched in the *BEI* database $\mathcal{D}b$ than the means J to achieve i: the agent should only abandon i when all possible ways of refining i have been explored to be unavailable. One possible relational postulate for revision of intentions is that the end intentions in the revised database should be a subset of those end intentions in the original database. There is currently few work on linking intention revision with instrumentality. One exception should be [12] which formalizes the 'rational' change of belief-desire-intention by quantifying the benefits of desires and gives formal postulates on the change of these mental states. In [24], the authors propose an AGM-like revision of the belief about action and time in a temporal logic. However, these contributions are still preliminary as many issues are not yet solved (eg. the frame problem, basic vs non-basic actions).

Let us also mention another research avenue that we would like to pursue in future work. Our introduction of environment actions open the road towards a more general, game-theoretic setting with multiple agents and semantics in terms of Alternating-time Transition Systems [1]. This naturally comes with the hypothesis that the actions and events are independent, which is the coherence assumption. We would in particular like to investigate how the database perspective can be combined with logics of propositional control [13].

References

1. Alur, R., Henzinger, T.A., Kupferman, O.: Alternating-time temporal logic. Journal of the ACM (JACM) 49(5), 672–713 (2002)
2. Audi, R.: A theory of practical reasoning. Journal of American Philosophical Quarterly 19, 25–39 (1982)
3. Baral, C., Gelfond, M.: Reasoning about intended actions. In: Proc. of the 20th National Conference on Artificial Intelligence (AAAI). vol. 20, pp. 689–694. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999 (2005)

4. Blount, J., Gelfond, M., Balduccini, M.: Towards a theory of intentional agents. In: AAAI Stanford Spring Symposium on Knowledge Representation and Reasoning in Robotics. pp. 10–17 (2014)
5. Blount, J., Gelfond, M., Balduccini, M.: A theory of intentions for intelligent agents. In: Proc. of the 13th intelnational conf. on Logic Programming and Nonmonotonic Reasoning (LPNMR). pp. 134–142. Springer (2015)
6. Bratman, M.: Intention, plans, and practical reason. Cambridge: Harvard University Press (1987), reedited 1999 with CSLI Publications
7. Bratman, M.: Intention, belief, and instrumental rationality. Reasons for action pp. 13–36 (2009)
8. Bratman, M.E., Israel, D.J., Pollack, M.E.: Plans and resource-bounded practical reasoning. Journal of Computational Intelligence 4, 349–355 (1988), `http://dx.doi.org/10.1111/j.1467-8640.1988.tb00284.x`
9. Cohen, P.R., Levesque, H.J.: Intention is choice with commitment. Journal of Artificial intelligence 42(2), 213–261 (1990)
10. Dignum, F., Conte, R.: Intentional agents and goal formation. In: Singh, M.P., Rao, A., Wooldridge, M. (eds.) Agent Theories, Architectures, and Languages: Agent Theories, Architectures, and Languages (ATAL-97), pp. 231–243. Springer Verlag, Berlin (1998)
11. Erol, K., Hendler, J., Nau, D.S.: HTN planning: Complexity and expressivity. In: Proc. of the 12th National Conference on Artificial Intelligence (AAAI). vol. 94, pp. 1123–1128 (1994)
12. Grant, J., Kraus, S., Perlis, D., Wooldridge, M.: Postulates for revising BDI structures. Synthese 175(1), 39–62 (2010)
13. van der Hoek, W., Walther, D., Wooldridge, M.: Reasoning about the transfer of control. Journal of Artificial Intelligence Research (JAIR) 37, 437–477 (2010), `http://dx.doi.org/10.1613/jair.2901`
14. Hunsberger, L., Ortiz Jr., C.: Dynamic intention structures I: a theory of intention representation. Proc. of Autonomous Agents and Multi-Agent Systems (AAMAS) 16(3), 298–326 (2008)
15. Icard, T., Pacuit, E., Shoham, Y.: Joint revision of belief and intention. In: Proc. of the 12th International Conference on Principles of Knowledge Representation and Reasoning (KR). pp. 572–574 (2010)
16. Lorini, E., Herzig, A.: A logic of intention and attempt. Synthese 163(1), 45–77 (2008)
17. Ortiz Jr, C.L., Hunsberger, L.: On the revision of dynamic intention structures. In: Proc. of the 11th International Symposium on Logical Formalizations of Commonsense Reasoning (2013)
18. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a BDI-architecture. Proc. of the 2nd International Conf. on Principles of Knowledge Representation and Reasoning (KR) 91, 473–484 (1991)
19. Sacerdoti, E.D.: The nonlinear nature of plans. In: Proc. of the 4th International Joint Conference on Artificial Intelligence (IJCAI). pp. 206–214. Morgan Kaufmann Publishers Inc. (1975)
20. Sardina, S., de Silva, L., Padgham, L.: Hierarchical planning in BDI agent programming languages: A formal approach. In: Proc. of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS). pp. 1001–1008. ACM (2006)
21. Shoham, Y.: Logical theories of intention and the database perspective. Journal of Philosophical Logic 38(6), 633–647 (2009)

22. Shoham, Y.: Why knowledge representation matters. Journal of Communications of the ACM 59(1), 47–49 (2015)
23. de Silva, L., Padgham, L.: A comparison of BDI based real-time reasoning and HTN based planning. In: AI 2004: Advances in Artificial Intelligence, pp. 1167–1173. Springer (2005)
24. Van Zee, M., Doder, D., Dastani, M., Van Der Torre, L.: AGM revision of beliefs about action and time. In: Proc. of the 24th International Joint Conference on Artificial Intelligence (IJCAI). pp. 3250–3256 (2015)
25. van Zee, M., Dastani, M., Doder, D., van der Torre, L.: Consistency conditions for beliefs and intentions. In: The 12th International Symposium on Logical Formalizations of Commonsense Reasoning. pp. 152–158 (2015)