A Computationally Grounded Model for Goal Processing in BDI Agents

Sam Leask, Natasha Alechina, Brian Logan
University of Nottingham
{svl,nza,bsl}@cs.nott.ac.uk

Abstract

A fundamental aspect of Beliefs-Desires-Intentions (BDI) agents is deliberation over goals. We present GROVE, a model of goal processing that casts deliberation as a choice over possible future executions of the agent's plans that are consistent with the agent's beliefs. GROVE unifies existing work on deliberation based on goal life-cycles, and makes central the connection between changes to beliefs and their consequences for goals. We compare GROVE to previous approaches, and show conformity to rationality postulates for an abstract model of BDI agents, and a precise relationship to [Harland et al 2014] model.

1 Introduction

In a BDI agent, goals represent desirable outcomes that the agent should achieve, which guide and motivate the behaviour of the agent. Agents deliberate over their goals in order to determine which goals they should commit to. Intentions are goals that the agent is committed to achieving that act as a filter on the possible future courses of action the agent may take [Cohen and Levesque, 1990; Rao and Georgeff, 1991]. The deliberation process is informed by the agent's beliefs, and, as the environment evolves and the agent's beliefs change to reflect the evolving environment, the agent may need to reconsider its intentions. An agent may reconsider its intentions if it believes they are no longer possible to pursue, or if an alternative set of intentions are more preferable to its current intentions. A BDI agent should choose a set of intentions that will best satisfy its desires (design goals) [Bratman, 1987].

In previous work, this deliberation process is represented by a *goal life-cycle*, where each goal is assigned a *state* that determines how it may influence the agent's behaviour. Goals move from one state to another depending on the results of deliberation. For example, an *active* goal may be *suspended* to avoid conflicts with other *active* goals. While there has been significant progress toward identifying key components of the goal life-cycle, there has been less work on integrating the goal life-cycle with the deliberative processes that dictate when transitions between states should occur, e.g., when a goal should be suspended as a result of deliberation. For example, Harland et al. [2014] propose an operational semantics for goal life-cycles in which goals may be adopted, dropped, aborted, suspended and resumed that unifies prior work [Winikoff *et al.*, 2002; van Riemsdijk *et al.*, 2008; Morandini *et al.*, 2009] on goal semantics and goal types. However, although the Harland et al. approach provides a generic framework for operations on goals, it relies on an unspecified *deliberation function* to reason about goal interactions and trigger transitions between goal states.

In this paper we propose a new model of goal processing, GROVE (Goals Reified Over Valid Executions), that grounds rational behaviour in the possible execution traces of an agent's program. Our model has the advantages of unifying many features relating to rational behaviour that have been discussed separately in the literature, including belief and goal interactions [Castelfranchi and Paglieri, 2007], goal and plan conflicts [Clement and Durfee, 1999; van Riemsdijk *et al.*, 2009; Thangarajah *et al.*, 2003a], goal priorities [Vikhorev *et al.*, 2011], and plan preferences [Visser *et al.*, 2016]. At the same time it provides a more precise specification of what counts as rational goal processing.

The remainder of the paper is organised as follows. Section 2 introduces and defines the basic components of our model of goal processing. In Section 3 we present our model of goal processing in BDI agents informally, and give a formal specification in Section 4. In Section 5 we show how GROVE conforms to the rationality postulates proposed by Grant et al. [2010], and that GROVE executions are a subset of those generated by the goal life-cycle semantics of Harland et al. [2014]. Section 6 discusses related work, and we conclude in Section 7.

2 Preliminaries

In this section, we introduce and define the basic components that form the basis of our model of goal processing.

We assume a set P of atoms, and denote by L the set of literals L over $P : L = P \cup \{\neg l \mid l \in P\}$. A positive literal $l \in L$ is entailed by a set of atoms $P' \subseteq P$, denoted $P' \models l$, iff $l \in P'$, and a negative literal $\neg l$ is entailed by P', denoted $P' \models \neg l$, iff $l \notin P$, i.e., negation is interpreted as negation as failure. The complement of a literal l is denoted $\sim l$, and the complement of a set of literals L, $\sim L$, is defined as:

$$\sim L = \{\neg l \mid l \in L\} \cup \{l \mid \neg l \in L\}$$

2.1 Beliefs, Plans and Goals

The agent's *beliefs* $B \subseteq P$ represent the agent's information about the environment and itself. The agent's *possible goals* are denoted by $D \subseteq P$, where each $g \in D$ represents a state of affairs that the agent may want to bring about, and which it has the means to achieve. A goal g is considered achieved iff $B \vDash g$.

To achieve its goals, an agent executes actions. The set of actions available to the agent are denoted by Act. The *preconditions* of an action $e \in Act$ are a set of literals which must be true before the execution of the action, and the *postconditions* of the action are a set of literals that are expected to be true after the execution of the action. For an action e with preconditions pre(e) and postconditions pos(e), if $B \vDash$ pre(e) then $B \vDash$ pos(e) immediately after executing e. The pre- and postconditions of an action are assumed to be consistent, i.e., pre(e) does not contain $l, \sim l$ for any l (and the same for pos(e)).

Actions are organised into plans. The set of plans available to the agent are denoted by Π. Each plan $\pi \in \Pi$ consists of a sequence of plan steps. Each plan step is either an *action* $e \in Act$ or a *subgoal* $!\,g$, $g \in D$. Plans are defined by the following grammar

$$\begin{aligned} \text{plan-step} &= e \mid !\,g \\ \pi &= \text{plan-step}^{+} \mid \epsilon \end{aligned}$$

where ϵ denotes the empty plan. The function $plans : D \mapsto 2^{\Pi} \setminus \{\emptyset\}$ returns the (non-empty) subset of the agent's plans that achieve a goal. We stipulate that $plans$ always returns a non-empty set, i.e., for each $g \in D$ the agent has at least one plan to achieve it, and that the goal is in the set of postconditions for the last step in a plan to achieve it.[1] We further assume that plans are side-effect free in the sense that the plan steps following a subgoal $!\,g$ may depend on g, but not the effects of executing a plan in $plans(g)$, and that the goal g does not occur as a subgoal in any plan $\pi \in plans(g)$, or any plan to achieve a subgoal in π, recursively. (Since goals are interpreted declaratively, in a well-formed agent program g should not occur as subgoal in any means to achieve g.)

The relations between goals, plans and actions can be represented using *goal plan trees* (GPT) [Clement and Durfee, 1999; Thangarajah *et al.*, 2003a; Thangarajah and Padgham, 2011].[2] The root of a GPT is a top-level goal (goal-node), and its children are the plans that can be used to achieve the goal (plan-nodes). Usually there are several alternative plans to achieve a goal: hence, the child plan-nodes are viewed as 'OR' nodes. By contrast, plan execution involves performing all the steps in the plan: hence, the children of a plan-node are viewed as 'AND' nodes. As in Yao et al. [2016b; 2016], we consider goal-plan trees in which plans may contain primitive actions in addition to sub-goals. Each goal g induces a goal plan tree $\tau = gpt(g)$, where g is the root node of the tree. A goal plan tree thus represents all possible ways of achieving the goal g available to the agent.

[1] Plans in $plans(g)$ are termed *relevant plans* for the *triggering condition* g in the BDI literature.

[2] Goal-plan trees can be derived in a straightforward way from BDI agent programs [de Silva and Padgham, 2004].

2.2 Step Sequences

A *step sequence* is a sequence $\sigma = s_1, s_2, \ldots, s_n$ where each *step* s_i is a pair (A, e) where A is a set of *active goals* $A \subseteq D$, and e is an action $e \in Act$. The active goals for an action e can be thought of as the ends for which the action forms (part of) the means. Given a step $s_i = (A, e)$, the function $agoals(s_i)$ returns A, the active goals of s_i, and the function $act(s_i)$ returns the action e.

The set of *effectuated goals* for a step sequence $\sigma = s_1, \ldots, s_n$ is defined by

$$egoals(s_1, \ldots, s_n) = \bigcup_{i=1}^{k} agoals(s_i)$$

Concatenation for step sequences is denoted by $\circ$. We stipulate that ϵ is identity for $\circ$, i.e., $\sigma \circ \epsilon = \epsilon \circ \sigma = \sigma$. The prefix of a sequence $\sigma = s_1, s_2, \ldots, s_n$ is a finite subsequence of steps $s_1, s_2, \ldots, s_i$ where $i \leq n$. A suffix of a sequence $s_1, s_2, \ldots, s_i, s_{i+1}, \ldots, s_n$ is a subsequence of steps $s_i, s_{i+1}, \ldots, s_n$ where $i \leq n$.

The *projection* of a step sequence $\sigma = s_1, \ldots, s_n$ with respect to a set of atoms E, $\sigma \downharpoonleft E$, is defined by

$$\begin{aligned} &\epsilon \downharpoonleft E = \epsilon \\ &(A, e) \downharpoonleft E \circ \sigma = \sigma \downharpoonleft E \text{ where } A \cap E \neq \emptyset \\ &(A, e) \downharpoonleft E \circ \sigma = (A, e) \circ \sigma \downharpoonleft E \text{ where } A \cap E = \emptyset \end{aligned}$$

that is, the projection of σ wrt E is the subsequence σ' in which all steps that have a goal in E as an active goal are removed. Note that projection preserves the ordering of steps in σ.

The pre- and postconditions of step sequences are denoted by *prec* and *post* respectively:

$$\begin{aligned} \text{prec}(A, e) &= \text{pre}(e) \\ \text{prec}(s_1, \ldots, s_n) &= \text{prec}(s_1) \cup \\ &\bigcup_{i=2}^{n} \Big[\text{prec}(s_i) \setminus \text{post}(s_1, \ldots, s_{i-1}) \Big] \end{aligned}$$

$$\begin{aligned} \text{post}(A, e) &= \text{pos}(e) \\ \text{post}(s_1, \ldots, , s_{n-1}, s_n) &= \text{post}(s_1, \ldots, s_{n-1}) \uplus \text{post}(s_n) \end{aligned}$$

where $\uplus$ is a set union operator that removes negated literals in the left-hand argument from the result if the corresponding positive literal occurs in the right-hand argument, and removes positive literals in the left-hand argument from the result if there are negated instances of them in the right-hand argument:

$$X \uplus Y \ = \ \big(X \setminus {\sim}Y\big) \cup Y$$

Note that the precondition of a step sequence excludes preconditions established by steps earlier in the sequence (and not undone).[3] On the other hand, the postconditions of a step

[3] These are called *p-effects* in [Thangarajah *et al.*, 2003a]. However we extend their notion of p-effect to include the establishment of the precondition of an action by a previous action in the same plan.

sequence includes *all* literals that are established by actions in the sequence.

A step sequence $s_1, s_2, \dots, s_n$ is *coherent* if no step destroys the preconditions of later step(s) in the sequence, that is at no step s_i there exists $l \in \text{post}(s_1, \dots, s_i)$ such that $\sim l \in \text{prec}(s_{i+1}, \dots, s_n)$. A coherent step sequence $s_1, s_2, \dots, s_n$ is *executable* given beliefs B if its preconditions are true given B, that is if $B \vDash \text{prec}(s_1, s_2, \dots, s_n)$.

2.3 Traces

A *trace* is a step sequence corresponding to a possible execution of a goal-plan tree for a goal. Traces are generated by expanding the goals and plans comprising the tree recursively. Expanding a plan step s_i in the plan $\pi = s_1, s_2, \dots, s_i, \dots, s_n$, $\pi \in \mathit{plans}(g)$, $g \in D$, where s_i is an action $e \in \mathit{Act}$, results in a pair $(\{g\}, e)$. Expanding a plan step s_i in the plan $\pi = s_1, s_2, \dots, s_{i-1}, s_i, s_{i+1}, \dots, s_n$, $\pi \in \mathit{plans}(g_0)$, $g_0 \in D$ where s_i is a subgoal $!\,g$, results in the sequence $(\{g_0\}, s_1), (\{g_0\}, s_2), \dots, (\{g_0\}, s_{i-1}),$ $(\{g_0, g\}, s_j), (\{g_0, g\}, s_{j+1}), \dots, (\{g_0, g\}, s_k),$ $(\{g_0\}, s_{i+1}), \dots, (\{g_0\}, s_n)$, where $s_j, s_{j+1}, \dots, s_k \in \mathit{plans}(g), g \in D$.

The set of execution traces, $\mathit{traces}(g)$, induced by the goal-plan tree rooted at the goal g is given by:

$$\begin{aligned}
\mathit{traces}(g) &= \mathit{traces}(\{g\}, g)\\
\mathit{traces}(A, g) &= \{\epsilon\} \cup\\
&\quad \{\sigma \mid \sigma = \mathit{expand}(A \cup \{g\}, s_1), \dots,\\
&\qquad \mathit{expand}(A \cup \{g\}, s_k)\\
&\qquad \text{for some } s_1, \dots, s_k \in \mathit{plans}(g)\}\\
\mathit{expand}(A, e) &= (A, e)\\
\mathit{expand}(A, !\,g') &\in \mathit{traces}(A, g')
\end{aligned}$$

We stipulate that each trace $\sigma \in \mathit{traces}(g)$ is coherent. This can be seen as as a 'well-formedness' condition on goal-plan trees: the executability of a trace may depend on features of the environment, but an action may not destroy a precondition of a later action in the same trace.

2.4 Interleavings

An *interleaving* is a step sequence corresponding to a possible execution of the goal-plan trees induced by a set of goals $G \subseteq D$. The set of interleavings for a set of traces is generated by freely interleaving the steps comprising the traces whilst preserving the ordering of steps within the traces and their coherence. More precisely, the set of interleavings generated by a set of goals $G = \{g_1, g_2, \dots, g_n\}$ is given by

$$\begin{aligned}
\mathit{inters}(G) = \{\rho \mid \rho = \sigma_i \,||\, \dots \,||\, \sigma_j \;\wedge \{g_i, \dots, g_j\} \subseteq G\\
\sigma_i \in \mathit{traces}(g_i), \dots, \sigma_j \in \mathit{traces}(g_j) \wedge\\
\rho \text{ is coherent } \}
\end{aligned}$$

where $||$ is the interleaving operator. That is, each interleaving $\rho \in \mathit{inters}(G)$ is executable in some environment (for each $\rho \in \mathit{inters}(G)$ there is a set of beliefs $B' \subseteq P$ such that $B' \vDash \text{prec}(\rho)$). A set of goals G may have no coherent interleavings. For example, achieving goals $g, g' \in G$ may each require the consumption of some non-renewable resource such as time, energy or money, so that it is possible to achieve either g or g' but not both.

In general, some interleavings will be preferred to others. For example, interleavings that achieve higher priority goals, or goals requiring fewer resources to achieve etc., may be preferred. We assume a *preference ordering* on interleavings specified by a relation $\mathit{prf}(B, \rho, \rho')$ which is true when the interleaving ρ is strictly preferred to the interleaving ρ' given beliefs B. The prf relation incorporates all preferences over sets of goals and the means to achieve them relevant to a particular domain.[4] We assume prf is a strict partial order: a relation that is irreflexive, asymmetric and transitive.

The set of *most preferred interleavings* for a set of goals G given beliefs B, $\mathit{pref}(B, G)$ is then defined as

$$\begin{aligned}
\mathit{pref}(B, G) = \{\rho \mid \rho \in \mathit{inters}(G) \wedge\\
\neg\exists \rho' \in \mathit{inters}(G) \text{ such that } \mathit{prf}(B, \rho', \rho)\}
\end{aligned}$$

Note that, in general, the interleavings in $\mathit{pref}(B, G)$ may achieve different subsets of G using different plans with differing costs and execution times, however from the point of view of the agent, they are all equivalent. For example, an agent may consider an interleaving that achieves the (single) goal of working in the lab, and an interleaving that achieves the goal of going home and the goal of having dinner, equally preferable.

A *history* is an interleaving containing the steps executed by the agent so far. A history p is a *subhistory* of a history h if there exists $E \subseteq \mathit{egoals}(h)$ such that $p = h \downharpoonright E$. We call E the *elided goals* of h and the steps in h not appearing in p *elided steps*. A history p is *extendable* if it is a prefix of a most preferred interleaving in $\mathit{pref}(B, G)$, $p \circ \sigma \in \mathit{pref}(B, G)$, and the suffix of the interleaving, σ, is executable: $B \vDash \text{prec}(\sigma)$. We call the suffix σ induced by an extendable history an *executable suffix*.

Definition 1 (Conservative Elision)**.** *A history p is a* conservative elision *of a history h (or simply conservative) if*

1. *p is extendable, i.e., for some σ, $p \circ \sigma \in \mathit{pref}(B, G)$ and $B \vDash \mathit{prec}(\sigma)$; and*
2. *there is no other extendable history p' of h that elides strictly fewer steps of h than p, i.e., there is no E', p', σ' such that $E' \subseteq \mathit{egoals}(h), p' = h \downharpoonright E', p' \circ \sigma' \in \mathit{pref}(B, G)$, $B \vDash \mathit{prec}(\sigma')$ and $|p'| > |p|$, where $|p|$ is the length of p.*

The set of possible future executions of an agent with beliefs B, goals G and history h is the set of executable suffixes of interleavings in $\mathit{pref}(B, G)$ induced by the conservative (sub)histories of h. More precisely,

Definition 2 (Possible Future Executions)**.** *The set of* possible future executions $\mathit{pexecs}(B, G, h)$ *is given by:*

$$\begin{aligned}
\mathit{pexecs}(B, G, h) = \{\sigma \mid \exists p \;(p \textit{ is conservative and}\\
p \circ \sigma \in \mathit{pref}(B, G))\}
\end{aligned}$$

[4] Although how prf is defined is not part of the formal model, we can imagine prf being specified by an agent developer as an input to an agent architecture that implements the GROVE model of rationality.

The set of *effectuated goals* of a possible future execution $\rho \in pexecs(B, G, h)$, $egoals(\rho)$, are the unachieved goals of the interleaving of which ρ is a suffix, i.e., the agent has executed one or more actions in pursuit of the goal but has not yet achieved it, or has not yet executed any actions in pursuit of the goal.

3 Goal Model

In this section we introduce our model of goal processing in BDI agents. The presentation is informal, and aims to convey the main intuitions. A more formal treatment is deferred to Section 4.

At any point in time, a BDI agent has a set of goals $G \subseteq D$, a set of beliefs $B \subseteq P$, and a history of the steps executed so far in pursuit of its goals, h. We refer to I and h together as the agent's *plan state*. An agent is *rational* if it commits to an interleaving in $I = pexecs(B, G, h)$. If the environment is static,[5] then executing an interleaving in I is guaranteed to succeed (recall that actions are assumed to always bring about their postconditions) and result in a most preferred outcome. If the environment is dynamic and the agent has no knowledge of how the environment is likely to change, it can do no better than committing to an interleaving that is most preferred in the current environment. Note that, in GROVE, unless $I = pexecs(B, G, h)$ is a singleton, there is not necessarily any single set of goals to which the agent is committed, if the (top-level) goals achieved by one interleaving $\rho \in I$,$egoals(\rho) \cap G$, differs from the goals achieved by a different interleaving $\rho' \in I$, i.e., if $egoals(\rho) \cap G \neq egoals(\rho') \cap G$, $\rho, \rho' \in I, \rho \neq \rho'$. Rather, the agent commits to top-level goals and subgoals as execution proceeds. In this, our model differs fundamentally from approaches that posit a clear commitment to top-level goals and a lower level of commitment to subgoals.

In the remainder of this section, we describe how the execution of an agent achieves behaviour that is rational, and how it maintains such behaviour when its goals or the environment change. The execution model has some similarities with the execution or deliberation cycle found in many BDI architectures, however there are important differences.

Execution is cyclic. At each cycle, the agent updates its beliefs B to reflect the current state of the environment; updates its goals G to be consistent with its beliefs and to incorporate any changes in its goals requested by users or other agents; computes a set of most preferred interleavings $I = pexecs(B, G, h)$; and finally executes the next action from an interleaving in I and updates the history h. Below, we explain each of these steps in more detail.

Updating Beliefs

At each cycle, the agent senses the environment, and uses this sensory information to update its belief state. The agent's belief update is modelled as a function, $sense(B)$, that takes the agent's current beliefs B as an argument, and returns an updated set of beliefs B' reflecting the environment state at this cycle

$$B' = sense(B)$$

We stipulate that the set of beliefs B' returned by $sense$ is consistent.

Updating Goals

The agent's goals change when goals are achieved, and in response to requests from users or other agents to adopt or drop a goal. A goal may be achieved by the postcondition of the final step in a trace for the goal, 'inadvertently' by the postconditions of a step in a trace for a different goal, or 'serendipitously' by a spontaneous change in the state of the environment. (Goals may also be dropped when the plan to achieve a goal fails; we do not consider this case here.)

Requests to adopt or drop goals are modelled by a function, $mesg(G)$, that takes the current goals G as an argument and returns the set of goals to be adopted, G^+, and the set of goals to be dropped, G^-:

$$(G^+, G^-) = mesg(G)$$

We stipulate that $G^- \subseteq G$ and $G^+ \subseteq D$, $G^+ \cap G^- = G^+ \cap G = \emptyset$. The agent's updated goals for this cycle, G', are then given by

$$G' = ((G \cup G^+) \setminus G^-) \setminus B'$$

Updating Plan State

When the belief and goal states have been updated, the possible future executions, $I' = pexecs(B', G', h)$, are (re)computed. When an agent adopts one or more new goals giving a new set of goals $G \subseteq G'$, interleavings in the new set of possible future executions I' may or may not achieve the new goals in $G' \setminus G$ or the old goals in G, depending on the agent's preference relation prf. For example, one or more of the new goals in G^+ may not be jointly achievable with goals in G, and the agent may prefer interleavings that achieve goals in G. Conversely, the newly adopted goals $G' \setminus G$ may be of high priority, and not jointly achievable with goals in G. Similarly, when the agent drops one or more goals giving a new set of goals $G' \subseteq G$, the goals achieved by the interleavings in I', may or may not be a subset of the goals achieved by the interleavings in I. For example, if a high priority goal that is not jointly achievable with other goals is dropped, the agent may be able to pursue a larger number of goals.

If the agent has achieved or dropped one or more goals at the current cycle, a new conservative elision of h is computed and used to determine the set of executable suffixes of interleavings in $pref(B', G')$ by removing steps in h to achieve goals E such that $\rho = h \downharpoonleft E \circ \sigma$, $\rho \in pexecs(B', G', h)$ and there is no E', $\rho' = p \downharpoonleft E' \circ \sigma'$, $\rho' \in pexecs(B', G', h)$ such that $|h \downharpoonleft E'| > |h \downharpoonleft E|$. It is irrational for an agent to repeat steps already performed if these steps also occur in a prefix of an interleaving in $pexecs(B', G', h)$. Moreover, some steps may not be repeatable, e.g., if a step has consumed a non-renewable resource.[6] We therefore require that the agent removes the minimum number of steps from h so that $h \downharpoonleft E$

[5] That is, the environment state changes only as a result of the agent's actions.

[6] Recall that a possible future execution must be executable in the current environment, so steps elided from h cannot make σ non-executable.

becomes a prefix of an interleaving in $pexecs(B', G', h)$. In the worst case, $E = egoals(p)$ and $h \downharpoonright E = \epsilon$; in the best case, e.g., when the set of goals has not changed or when a goal is dropped that the agent was not actively pursuing, $h \downharpoonright E = h$, and h is a prefix of all interleavings in $pexecs(B', G', h)$.

As execution progresses and the history h increases in length, the cardinality of $pexecs(B, G, h)$ will typically decrease. This reflects the fact that, as the agent executes steps, it furthers its commitments to a particular course of action (which may include committing to top-level goals in G).

Action Execution

At each cycle, the agent selects an interleaving $\rho \in I' = pexecs(B, G, h)$ and executes the first action in ρ. That is, the agent selects an interleaving that is consistent with the actions it has executed so far and executes the next action after the conservative elision of h from the interleaving. The action executed is then added to the history to give a new history $h' = h \circ e$, where $\rho = e \circ \rho'$.

4 Semantics

An agent configuration is a 4-tuple $\langle B, G, h, f\rangle$ where B is a set of beliefs, G is a set of top-level goals, h is a history and f is a phase flag as explained below. We stipulate that in the initial configuration, $B \cap G = \emptyset$.

We model the evolution of the agent configuration and environment with an execution cycle. Each execution cycle is made up of linearly executed phases, starting with the belief update phase and ending in the execution phase. The current phase is represented in the configuration by a phase flag from the set $\{s, m, a\}$. In the *belief update* (s) phase the agent's beliefs are updated to reflect changes resulting from the agent's most recently executed action and changes that have happened independently of the agent. In the *goal update* (m) phase, the agent's goals are updated in response to requests from users or other agents to adopt or drop goals, when goals are achieved, and when they become unachievable. The goal update phase is followed by the *execution* (a) phase, in which the set of most preferred interleavings and their possible future executions are (re)computed, the first step of a possible future execution is executed, and the history of executed actions is extended with the executed step.

Belief update phase

$$\frac{B' = sense(B)}{\langle B, G, h, s\rangle \rightarrow \langle B', G, h, m\rangle} \quad (1)$$

Rule (1): update the belief set B and move to the goal update phase.

Goal update phase

$$\frac{(G^+, G^-) = mesg(G) \quad G' = ((G \cup G^+) \setminus G^-) \setminus B}{\langle B, G, h, m\rangle \rightarrow \langle B, G', h, a\rangle} \quad (2)$$

Rule (2): update the goal set G with the goals to be adopted G^+ and dropped G^- at this cycle, drop any achieved goals, and move to the execution phase.

Execution phase

$$\frac{\rho \in pexecs(B, G, h) \quad \rho = (A, e) \circ \sigma}{\langle B, G, h, a\rangle \rightarrow \langle B, G, h \circ e, s\rangle} \quad (3)$$

Rule (3): select a possible future execution ρ, execute the first action e in ρ, and extend the history of executed actions with e. Move to the sense phase.

$$\frac{pexecs(B, G, h) = \emptyset}{\langle B, G, h, a\rangle \rightarrow \langle B, G, h, s\rangle} \quad (4)$$

Rule (4): if there are no possible future executions, move to the sense phase.

Note that, in contrast to other approaches, we do not drop unachievable top-level goals. Rather we hope that the environment evolves in such a way to enable achievement of currently unachievable top-level goals.

5 Discussion

In this section, we consider two leading approaches to specifying rational behaviour from the BDI literature. We show that GROVE conforms to the rationality postulates proposed by Grant et al. [2010], and that GROVE executions are a subset of those generated by the goal life-cycle semantics of Harland et al. [2014] (i.e., GROVE agents are 'more rational' than than agents conforming to the Harland et al. model).

5.1 Rationality Postulates for Revising BDI Structures

In [Grant *et al.*, 2010], a high-level model of a mental state of BDI agents is proposed, called a BDI structure. A BDI structure S is a tuple $\langle B, D, I, v, (c, C)\rangle$, where B is a set of beliefs (all consequences of a finite belief base B_0), D is a set of declarative goals (in the same language as beliefs), I is a set of intentions (pairs (action, goal)), with functions $goals(I)$ and $actions(I)$ returning respectively the set of goals occurring in I and the set of actions occurring in I, v is a function from sets of goals to non-negative real numbers returning the value of a set of goals to the agent if it is achieved (satisfying the condition that a superset has at least the same value as its subset) and $C \supseteq actions(I)$, and c is a function from subsets of C to non-negative real numbers (cost of executing this set of actions). c also satisfies the condition that a superset of a set of actions costs at least as much. The postulates on a *rational* BDI structure are:

A1 B is consistent, i.e., $B \not\vdash \bot$

A2 I is feasible in the context of B (for every $(\alpha, \theta) \in I$, $B \vdash r_{\alpha,\theta}$, where $r_{\alpha,\theta}$ says that α's preconditions are true, and α terminates and makes θ true)

A3 $goals(I)$ is consistent

A4 For every $\theta \in goals(I)$, $B \not\vdash \theta$

A5 There is no I' such that $S' = \langle B, D, I', v, (c, C)\rangle$ satisfies A1 - A4 and $ben(I') > ben(I)$, where $ben(I) = v(goals(I)) - c(actions(I))$, in other words, there is no other set of intentions the agent can select which achieves more valuable goals by cheaper means.

Structures satisfying A1 - A4 are referred to in [Grant *et al.*, 2010] as weakly rational BDI structures (WRBDI) and structures satisfying A1 - A5 as rational BDI structures (RBDI). There are several complexity results stated concerning WRBDI and RBDI structures, but no algorithms given for revising the structures in a rational way. Our work can be seen as a step towards providing a computationally grounded approach to this problem.

Theorem 1. *GROVE is weakly rational in the sense of [Grant* et al.*, 2010], i.e., satisfies postulates A1 - A4.*

Proof. A1 holds because beliefs are atomic, A2 holds because all actions in a possible future execution are executable (from Definition 2) and guaranteed to achieve the goal if executed (since we require that a goal must be in the set of postconditions for the last step in a plan to achieve it). A3 holds because the effectuated goals of a possible future execution are consistent (from the coherence of interleavings). A4 holds because achieved goals are dropped (Rule (2) for top-level goals, and as a consequence of Definition 1 for subgoals). □

GROVE does not assume that a numerical value can be assigned to each set of goals or a cost to each set of actions; however, if this is possible, then these values can be used to derive a preference order on interleavings that chooses the optimal set of interleavings for execution.

5.2 HMTY Goal Life-Cycle Semantics

In [Harland *et al.*, 2014], an operational semantics for the life-cycles of goals is presented, that aims to unify the prior work on establishing goal semantics [van Riemsdijk *et al.*, 2008; Morandini *et al.*, 2009]. In this section, we give a high-level overview of the approach of the Harland et al. [2014] approach (hereinafter HMTY), and show that, while all GROVE executions are consistent with the HMTY operational semantics, some HMTY executions are not rational in the sense of GROVE (and we would argue, are not rational). HMTY consider both achievement and maintenance goals. As GROVE currently does not encompass maintenance goals, we focus on achievement goals here.

A configuration in HMTY is a tuple $\langle \mathcal{B}, \mathcal{G} \rangle$ where $\mathcal{B}$ is a set of beliefs and $\mathcal{G}$ is a set of goal contexts of the form $\langle I, \text{ach}(\kappa, S, F), Rules, State, \pi \rangle$, where I is a goal context identifier, κ is a goal context condition, S is a success condition, F is a failure condition, $Rules$ is a set of condition-action pairs for goal update, $State$ is a state flag, and π is a plan body.

Goals in the Pending state have no plan associated with them, and are not currently being executed. A Pending goal may be activated as a consequence of deliberation provided that the context condition κ is true.

Goals in the Active state must have a plan body associated with them, and are considered executable.

If either of a goal's success condition S or failure condition F are true, the goal is dropped. As each condition-action pair is triggered by the state of the beliefs, state transitions are triggered by changes to beliefs.

An unspecified deliberation process is assumed to add beliefs to the belief base which trigger state transitions for goals.

Plans are assigned to goals in the Active state by way of a means-end reasoning function, which allows for both prewritten plans and online generation of plans.

Next we prove a form of equivalence between the GROVE and HMTY models. We argue that, under certain background assumptions, all executions of a GROVE agent are valid executions in HMTY. (The converse is not the case; GROVE imposes more constraints on rational executions, for example it only adopts executable interleavings.) The background assumptions are as follows. In the interests of brevity, we assume that the environment is static, known (beliefs are complete), that actions are infallible and have no duration. In addition, we consider only achievement goals.

Theorem 2. *Let the environment be static and actions infallible. Then for any GROVE agent with initial configuration $\langle B, G, \epsilon, s \rangle$ and an execution history h', there is an HMTY agent with the same goals and plans which produces the same execution history.*

Proof. The idea of the proof is as follows. Given a GROVE agent with the initial configuration $\langle B, G, \epsilon, s \rangle$, and an execution history h' starting from this configuration, we define an HMTY agent with a 'matching' initial configuration $\langle \mathcal{B}, \mathcal{G} \rangle$ which will produce the same execution h'. We first define 'matching' precisely, and then show that if two configurations are matching, then if an execution step is possible in the GROVE configuration, then the same step is possible in the HMTY configuration, and the resulting configurations (possibly after some number of internal transitions) are matching.

Consider a GROVE configuration $\langle B, G, \epsilon, a \rangle$ (reached by internal transitions from $\langle B, G, \epsilon, s \rangle$) and a history h' which is generated from this initial configuration. Since we are assuming that the environment is static and actions are infallible, then without loss of generality we can assume that the agent is executing a single interleaving σ_0 such that $pexecs(B, G, \epsilon) = \{\sigma_0\}$. (In fact, it is sufficient to assume that we can identify σ_0 which the agent is executing to generate h', as an element of $pexecs(B, G, \epsilon)$.)

For the sake of simplifying the proof, we assume that $expand$ is implemented so that the effectuated goal set and action in each step are annotated with plans and goals respectively (this information can clearly be added when an interleaving is generated). In particular, for each step (A, e), each effectuated goal $a \in A$ is annotated with a plan π_a which corresponds to the plan that was selected to achieve a in the expansion of the goal-plan tree; the action e is annotated with the goal g_e that it achieves. We denote an annotated goal by $g{:}\pi_g$ and an annotated action by $e{:}g_e$.

The matching initial HMTY configuration is $\langle \mathcal{B}, \mathcal{G} \rangle$ where $\mathcal{B} = B$, and $\mathcal{G}$ contains tuples of the form $\langle I, \text{ach}(\top, g, \bot), Rules, Pending, \epsilon \rangle$ for each $g \in G$. The $Rules$ are the 'standard' HMTY rules for adopting and dropping goals; we assume that they allow making each of the goals in $\mathcal{G}$ Active and associating a plan π with it.

The annotations of steps in σ_0 are used to determine what should happen in the HMTY agent. When its goals are activated and plans are assigned to them, then the annotations on

A inform the choice of plan for the goal. That is, for a goal g that is being activated, $mer = \pi_g$ where $g{:}\pi_g \in A$.

Subgoal steps that are executed are replaced by SGP, according to the HMTY semantics. The subgoal step $!g'$ is replaced by $g' \vee \text{drop}(I')$ $:?S$ for the subgoal g' with id I' in $\mathcal{G}$. A plan with SGP at the head cannot be progressed until its subgoal g' has been dropped (by deliberation fact) or achieved.

For an arbitrary GROVE configuration $\langle B, G, h, a\rangle$ with a history h which is a prefix of h', where the first $|h|$ steps of σ_0 are executed and $pexecs(B, G, h) = \{\sigma_h\}$, the matching HMTY configuration intuitively corresponds to removing the actions in h from the plans for goals and subgoals adopted so far. More precisely, it is a configuration $\langle \mathcal{B}, \mathcal{G}\rangle$ where $\mathcal{B} = B \cup DF$ (where DF are deliberation facts of the form activate(I) that are added in HMTY in order to activate goals), and $\mathcal{G}$ contains tuples corresponding to (g, π) where $g \in G$ or g is a subgoal of some plan for $g' \in G$ and π is a suffix of a plan for g where the prefix of the plan is a subsequence of h.

In order to show that at any point in the history h', the HMTY agent can execute the same action as the one executed by the GROVE agent, we need to show that in the matching configurations, the action in the first element of σ_h is executable by the HMTY agent, and the resulting configurations again match. In the initial situation the action $e{:}g_e$ in the first element of σ_0 is the first action of a plan $e \circ \pi$ for some $g_e \in G$. It can be executed by the HMTY agent by assumption, since there is a goal $\langle I, \text{ach}(\top, g, \bot), Rules, Pending, \epsilon\rangle$ in the HMTY configuration, which can be activated and assigned a plan $e \circ \pi$ to become $\langle I, \text{ach}(\top, g, \bot), Rules, Active, e \circ \pi\rangle$.

For the inductive step, we need to consider two cases for the action $e{:}g_e$ in the first element $(A, e{:}g_e)$ in σ_h. The first case is when e belongs to a plan for $g_e \in G$, which is as before. The second case is when e is the first action in a remaining plan for a subgoal g_e. Since by definition of a matching configuration there is a goal of the form $\langle I, \text{ach}(\top, g_e, \bot), Rules, Active, e \circ \pi\rangle$ in the HMTY configuration, it can be chosen for execution and in the resulting configuration, there is a goal $\langle I, \text{ach}(\top, g_e, \bot), Rules, Active, \pi\rangle$, so the HMTY configuration again matches the GROVE configuration corresponding to $\sigma_{h \circ e}$.

□

By showing that GROVE executions are a subset of those in HMTY, we reveal that there are some executions permitted by HMTY that are not permitted by GROVE. These executions are those that correspond to irrational behaviour permitted by HMTY.

6 Related Work

In this section, we discuss previous approaches to handling interactions between goals and goal dynamics that are most closely related to GROVE.

Castelfranchi & Paglieri [Castelfranchi and Paglieri, 2007] describe a model of goal processing in which goals are on a scale between desire and intention, and desire and intention are both expressed in terms of (states of) goals. Goals progress from desire to intention and are "filtered" by beliefs that characterise the support required to progress to the next stage. As in GROVE, beliefs filter the future executions which in turn dictate the degree to which goals are intended.

There has been considerable work on goal dynamics, goal lifecycles, and rational commitment. Van Riemsdijk et al. [van Riemsdijk *et al.*, 2008] define two goal types: achievement and performative, and an give an abstract architecture for goals. Once adopted, goals may move between suspended and activated states, until they are finally dropped. In the suspended state, a goal is simply inactive, while in the activated state, a plan is is assigned to the goal and the plan is executed. Morandini et al. [Morandini *et al.*, 2009] extend the semantics for goals given by Riemsdijk et al. [van Riemsdijk *et al.*, 2008] to define an operational semantics for the behaviour of leaf and non-leaf goals in goal models.

As discussed in Section 5.2, Harland et al. [2014] give an operational semantics for a goal lifecycle model that encompasses the behaviour of goals of monitoring and goals of accomplishment. Their semantics includes states and transitions for aborting, suspending and resuming goals, handling the execution of plans, and subgoaling. Their work unifies the existing semantics for goals of monitoring and accomplishment [van Riemsdijk *et al.*, 2008; Morandini *et al.*, 2009; Duff *et al.*, 2006] with the authors' previous work on aborting, resuming and suspending goals [Thangarajah *et al.*, 2007; Thangarajah *et al.*, 2008; Thangarajah and Padgham, 2011]. In [Harland *et al.*, 2014] decisions about which transitions to perform and when are made by a *deliberation function*. The deliberation function is assumed to be consistent with the HMTY operational semantics, but is not further specified. We address this issue in GROVE by defining rational deliberation in terms of preferences over possible future executions.

In [Harland *et al.*, 2017] Harland et al. develop a generic BDI-based execution model for handling the aborting, suspending, and resuming goals and (potentially parallel) subgoals, offering finer control over the agent's tasks and subtasks in the presence of these operations. The semantics they give ensure that all sub-tasks are aborted before the task is aborted, and similarly that sub-tasks are suspended before suspension of the parent task. The authors distinguish between cases in which aborting a plan or goal imply failure, and cases in which aborting is possible without leading to failure of the plan or goal. For instance, they note that aborting a plan's subgoals necessarily causes it to fail, however aborting that plan is not necessarily indicative of failure. The authors also note the desirability of storing resumption conditions for suspended tasks, especially in the case that a task has been suspended in order to avoid conflicts. We address the issue of managing the suspension and resumption of goals and plans in our model using interleavings.

A parallel strand of work in the Goal Reasoning literature casts reasoning about goals (and ultimately goal semantics, including a goal life-cycle model) as goal refinement, an extension of the notion of plan refinement [Roberts *et al.*, 2014; Roberts *et al.*, 2015]. Goals are transitioned between modes (which are analogous to state flags) by application of refinement strategies. The refinements that can be applied to a goal are subject to a set of constraints on that goal; for example, a plan selected to achieve a goal must satisfy its formulated constraints. If a goal's constraints are violated while it is executing, a resolution strategy is applied that determines whether the goal can be recovered and does so if possible. Each mode in the lifecycle represents a decision to be made in the deliberative process, from the formulated goal about which little has been decided, through to the dispatched goal which has a plan being executed for it. In [Roberts *et al.*, 2015], the authors integrate their goal lifecycle model with an existing model of online planning and execution.

In [Johnson *et al.*, 2016], the goal reasoning lifecycle model is extended with *information measures* that specify domain-specific information about the current and expected progress of goals and plans, and which enables prediction of failure i.e., due to constraints not being satisfied.

The goal reasoning lifecycle models are similar to GROVE in that they provide a framework for reasoning about the deliberation process by which an agent selects a set of goals to achieve. However, they frame a solution to the goal reasoning problem at the individual goal level, and it is unclear how interactions between goals are managed. The hard constraints in the goal reasoning lifecycle model constrain the options the agent can consider, for instance the plans that may be selected to achieve a goal. The notion of executability in GROVE is analogous to a hard constraint, as inexecutability (a violation of the constraint) corresponds to failure.

There has been considerable work on reasoning about conflicts [Clement and Durfee, 1999; Thangarajah *et al.*, 2003a; Pokahr *et al.*, 2005; van Riemsdijk *et al.*, 2009; Zatelli *et al.*, 2016; Yao *et al.*, 2016a] and synergies [Thangarajah *et al.*, 2003b; Yao *et al.*, 2016b] between goals. For example, Thangarajah et al. [2003a] describe an approach based on *summary information* that avoids conflicts by reasoning about necessary and possible pre- and post-conditions of different ways of achieving a goal. They also present mechanisms to determine whether a newly adopted (sub)goal will definitely be safe to execute without conflicts, or will definitely result in conflicts, or may result in conflicts. If the goal cannot be executed safely, execution of the intention is deferred. Yao et al. [2016a] present a stochastic approach based on Single-Player Monte-Carlo Tree Search (SP-MCTS) in which pseudorandom simulations of different interleavings of the plans in each intention are used to determine which intention to progress. Zatelli et al. [2016] consider the detection and avoidance of conflicts between goals. Their solution relies on developer-specified annotations for plans in order to determine when there is a conflict. This is in contrast to our approach where conflicts are detected using condition sets derived from traces through goal-plan trees, as in [Clement and Durfee, 1999; Thangarajah *et al.*, 2003a; Yao *et al.*, 2016a; Yao *et al.*, 2016b] Our approach does not require that the developer foresee conflicting executions of their plans, and instead leaves the task of detecting conflicts to the agent.

There has also been work on the use of preferences to inform the choice of which goals to commit to and what means should be used to achieve them. For example, Visser et al. [2016] develop an approach in which goal-plan trees are annotated with preference information from which summaries are derived and further refined by user input. We have made the notion of preference central to our model as a means to define a rational choice over interleavings, however the preference function itself is abstract.

7 Conclusions

We present GROVE, a model of goal processing that casts deliberation as a choice over possible future executions of the agent's plans that are consistent with the agent's beliefs. GROVE unifies existing work on deliberation based on goal life-cycles, and makes central the connection between changes to beliefs and their consequences for goals. We showed that GROVE conforms to the rationality postulates for BDI agents given by Grant et al. [2010] the the operational semantics for goal life-cycles given by Harland et al. [2014].

We have focussed on *perfect rationality* in this paper. One direction for future work would be to consider approximations to perfect rationality or bounded rationality, by bounding the lookahead of possible future executions. Other avenues for future work include incorporating from [Harland *et al.*, 2014] that GROVE lacks, such as support for abort methods and maintenance goals.

References

[Bratman, 1987] M. Bratman. *Intention, Plans, and Practical Reason*. Center for the Study of Language and Information, 1987.

[Castelfranchi and Paglieri, 2007] C. Castelfranchi and F. Paglieri. The role of beliefs in goal dynamics: prolegomena to a constructive theory of intentions. *Synthese*, 155(2):237–263, 2007.

[Clement and Durfee, 1999] B. J. Clement and E. H. Durfee. Theory for coordinating concurrent hierarchical planning agents using summary information. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence*, pages 495–502, 1999.

[Cohen and Levesque, 1990] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2-3):213–261, 1990.

[de Silva and Padgham, 2004] L. de Silva and L. Padgham. A Comparison of BDI Based Real-Time Reasoning and HTN Based Planning. In *AI 2004: Advances in Artificial Intelligence, 17th Australian Joint Conference on Artificial Intelligence, Proceedings*, pages 1167–1173. Springer, 2004.

[Duff *et al.*, 2006] S. Duff, J. Harland, and J. Thangarajah. On proactivity and maintenance goals. In *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006)*, pages 1033–1040, 2006.

[Grant *et al.*, 2010] J. Grant, S. Kraus, D. Perlis, and M. Wooldridge. Postulates for revising BDI structures. *Synthese*, 175(Supplement-1):39–62, 2010.

[Harland *et al.*, 2014] J. Harland, D. N. Morley, J. Thangarajah, and N. Yorke-Smith. An operational semantics for the goal life-cycle in BDI agents. *Autonomous Agents and Multi-Agent Systems*, 28(4):682–719, 2014.

[Harland *et al.*, 2017] J. Harland, D. N. Morley, J. Thangarajah, and N. Yorke-Smith. Aborting, suspending, and resuming goals and plans in BDI agents. *Autonomous Agents and Multi-Agent Systems*, 31(2):288–331, 2017.

[Johnson *et al.*, 2016] B. Johnson, M. Roberts, T. Apker, and D. Aha. Goal reasoning with information measures. In *Proceedings of the Fourth Annual Conference on Advances in Cognitive Systems*, 2016.

[Morandini *et al.*, 2009] M. Morandini, L. Penserini, and A. Perini. Operational semantics of goal models in adaptive agents. In *8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, pages 129–136, 2009.

[Pokahr *et al.*, 2005] A. Pokahr, L. Braubach, and W. Lamersdorf. A goal deliberation strategy for BDI agent systems. In *Multiagent System Technologies, Third German Conference, MATES 2005, Proceedings*, pages 82–93, 2005.

[Rao and Georgeff, 1991] A. S. Rao and M. P. Georgeff. Modeling rational agents within a bdi-architecture. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 473–484, 1991.

[Roberts *et al.*, 2014] M. Roberts, S. Vattam, R. Alford, B. Auslander, J. Karneeb, M. Molineaux, T. Apker, M. Wilson, J. McMahon, and D. W. Aha. Iterative goal refinement for robotics. In *Planning and Robotics: Papers from the ICAPS Workshop*. AAAI Press, 2014.

[Roberts *et al.*, 2015] M. Roberts, S. Vattam, R. Alford, B. Auslander, T. Apker, B. Johnson, and D. W. Aha. Goal reasoning to coordinate robotic teams for disaster relief. In *Planning and Robotics: Papers from the ICAPS Workshop*. AAAI Press, 2015.

[Thangarajah and Padgham, 2011] J. Thangarajah and L. Padgham. Computationally effective reasoning about goal interactions. *Journal of Automated Reasoning*, 47(1):17–56, 2011.

[Thangarajah *et al.*, 2003a] J. Thangarajah, L. Padgham, and M. Winikoff. Detecting & avoiding interference between goals in intelligent agents. In *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 721–726, 2003.

[Thangarajah *et al.*, 2003b] J. Thangarajah, L. Padgham, and M. Winikoff. Detecting & exploiting positive goal interaction in intelligent agents. In *The Second International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2003, Proceedings*, pages 401–408, 2003.

[Thangarajah *et al.*, 2007] J. Thangarajah, J. Harland, D. N. Morley, and N. Yorke-Smith. Aborting tasks in BDI agents. In *6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007)*, page 6, 2007.

[Thangarajah *et al.*, 2008] J. Thangarajah, J. Harland, D. N. Morley, and N. Yorke-Smith. Suspending and resuming tasks in BDI agents. In *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, pages 405–412, 2008.

[van Riemsdijk *et al.*, 2008] M. B. van Riemsdijk, M. Dastani, and M. Winikoff. Goals in agent systems: a unifying framework. In *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Volume 2*, pages 713–720, 2008.

[van Riemsdijk *et al.*, 2009] M. B. van Riemsdijk, M. Dastani, and J. C. Meyer. Goals in conflict: semantic foundations of goals in agent programming. *Autonomous Agents and Multi-Agent Systems*, (3):471–500, 2009.

[Vikhorev *et al.*, 2011] K. Vikhorev, N. Alechina, and B. Logan. Agent programming with priorities and deadlines. In *Proceedings of the Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 397–404, 2011.

[Visser *et al.*, 2016] S. Visser, J. Thangarajah, J. Harland, and F. Dignum. Preference-based reasoning in BDI agent systems. *Autonomous Agents and Multi-Agent Systems*, 30(2):291–330, 2016.

[Winikoff *et al.*, 2002] M. Winikoff, L. Padgham, J. Harland, and J. Thangarajah. Declarative & procedural goals in intelligent agent systems. In *Proceedings of the Eighth International Conference on Principles and Knowledge Representation and Reasoning (KR-02)*, pages 470–481, 2002.

[Yao and Logan, 2016] Y. Yao and B. Logan. Action-level intention selection for bdi agents. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016)*, pages 1227–1235. IFAAMAS, 2016.

[Yao *et al.*, 2016a] Y. Yao, L. de Silva, and B. Logan. Reasoning about the executability of goal-plan trees. In *Engineering Multi-Agent Systems - 4th International Workshop, EMAS 2016*, pages 176–191, 2016.

[Yao *et al.*, 2016b] Y. Yao, B. Logan, and J. Thangarajah. Robust execution of BDI agent programs by exploiting synergies between intentions. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2558–2565, 2016.

[Zatelli *et al.*, 2016] M. R. Zatelli, J. F. Hübner, A. Ricci, and R. H. Bordini. Conflicting goals in agent-oriented programming. In *Proceedings of the 6th International Workshop on Programming Based on Actors, Agents, and Decentralized Control, AGERE 2016*, pages 21–30, 2016.