A Framework for Plan Library Evolution in BDI Agent Systems

Mengwei Xu, Kim Bauters, Kevin McAreavey, Weiru Liu
School of Computer Science, Electrical and Electronic Engineering, and Engineering Maths
University of Bristol, Bristol, UK
{mengwei.xu, kim.bauters, kevin.mcareavey, weiru.liu}@bristol.ac.uk

***Abstract*—The Belief-Desire-Intention (BDI) paradigm is a flexible framework for representing intelligent agents. Practical BDI agent systems rely on a static plan library to reduce the planning problem to the simpler problem of plan selection. However, fixed pre-defined plan libraries are unable to adapt to fast-changing environments pervaded by uncertainty. In this paper, we advance the state-of-the-art in BDI agent systems by proposing a plan library evolution architecture with mechanisms to incorporate new plans (plan expansion) and drop old/unsuitable plans (plan contraction) to adapt to changes in a realistic environment. The proposal follows a principled approach to define plan library expansion and contraction operators, motivated by postulates that clearly highlight the underlying assumptions, and quantified by decision-support measures of temporal information. In particular, we demonstrate the feasibility of the proposed contraction operator by presenting a multi-criteria argumentation based decision making to remove plans exemplified in a planetary vehicle scenario.**

***Index Terms*—BDI Agent Systems, Agent Reasoning, Plan Library Expansion/Contraction, Multi-criteria Decision Making**

I. Introduction

The Belief, Desire, and Intentions (BDI) architecture [1] is a popular and well-studied framework for developing intelligent agents. Most of the languages based on BDI – such as AgentSpeak [2], CAN [3], CANPLAN [4] – rely on a pre-defined plan library to react to events. Each plan in the plan library prescribes under which conditions the plan is valid, and which steps should be taken to respond to the event. This (computationally) efficient design has been proven very successful in areas such as business [5], healthcare [6], industry [7].

However, intelligent agents should also be able to adapt to a changing environment. The aforementioned approaches are not able to do so, given that their plan libraries are fixed and pre-defined. Nevertheless, real-world environments do often change over time. Furthermore, realistic environments can be non-deterministic, making it particularly difficult for an agent designer to foresee all eventualities, and hence to create plans in advance to deal with all obscure situations. When an intelligent agent ends up in a situation where its pre-defined response is inadequate or incorrect because the environment changed, or in a situation that was not foreseen at design time, it should be able to augment the range of behaviours (i.e. the BDI plans) in order to cope with changes in the environment.

To illustrate the problem, consider the example of a Mars Rover exploring the surface of Mars. The Rover, which is pre-designed with tasks (e.g. carrying out science experiments), must utilise a high degree of autonomy due to the high-latency communication channels to Earth. For example, when an off-nominal event (e.g. a blue rock) is detected, it would increase science exploration if the Rover can respond to such "science opportunities". This autonomous behaviour implies that new plans (e.g. navigation plan) may need to be generated. Furthermore, since the Rover will always return to the lander to deliver samples to the ascent vehicle, this lends an opportunity to the Rover to (potentially) use knowledge obtained during navigation to a sampling site on its return. If the Rover has safely navigated to a location, "remembering" the route it took (i.e. adding such a navigation plan to its plan library) and then returning by the same path are promising features highlighted in [8]. However, some of these "*remembered*" plans may fail because of the changing Martian surface. Therefore, future planetary Rovers demand more adaptive agent systems which can both add and delete plans intelligently.

Over the years, many enhancements have been proposed to provide existing BDI agent systems with powerful autonomous planning capabilities. An overview of these is given in the survey by Meneguzzi and De Silva [9]. However, most of these planning extensions overlooked the potential adoption of new plans, and continue to treat the plan library as a fixed and pre-defined set. For example, to compensate for the inadequacy of a plan library in an uncertain environment, the authors of [10] proposed the AgentSpeak+ framework, which extends AgentSpeak with a mechanism for probabilistic planning named Partially Observable Markov Decision Processes (POMDPs) [11]. However, once the goal was met, the (potentially valuable) plan obtained from the POMDP was simply forgotten. Other promising works, such as [12], [13], [14], proposed the integration of classical planners and BDI agents to generate new plans. Unfortunately, none of them considers expanding the set of pre-defined plans by adopting these new plans. To the best of our knowledge, the only work considering the reuse of new plans (achieved by adding them to the plan library) is found in [15]. Still, this work solely focuses on leveraging the new plans in a classical setting and approaches the plan library expansion in an ad-hoc manner.

In this paper, we investigate the structure of a pre-defined plan library and define a generic framework that allows a BDI agent to incorporate new plans from automated planning tools for unforeseen situations. We will refer to this step as plan library expansion. However, merely adding plans is not enough for an agent. As the agent ages, some plans may become unsuitable, hampering its reactive nature which is crucial to the success of BDI agents. For instance, an approach to an event (e.g. the need to enter another room) which worked in the past (e.g. turning a handle) may no longer work in the future (e.g. the handle has been removed, and a button needs to be pressed instead). Therefore, there is a need for plan contraction as well. Plan library contraction is an altogether more significant – albeit challenging – problem than expansion because it relies on both qualitative and quantitative measures associated with each plan in the library to determine which plans are no longer deemed valuable and so can be removed. For example, a plan may be flagged for deletion because it became obsolete (e.g. a low number of calls) or because it became incorrect (e.g. a high failure rate). However, due to the nature of a plan library, care must be taken when deleting plans to avoid undesirable side-effects (e.g. deleting a (sub)plan relied upon by another highly successful plan).

To achieve these objectives, we follow a principled approach to a plan library expansion and contraction, motivated by postulates that clearly highlight the underlying assumptions, and supported by measures which are able to characterise plans in the library. The contributions of this paper are therefore threefold. *Firstly*, we provide a systematic specification of domain independent characteristics (e.g. the quality of plans) of the plan library as the basis for plan library expansion and contraction reasoning. *Secondly*, we define a plan library expansion operator and formally shows the benefits of expansion regarding relevant characteristics. *Thirdly*, we introduce an operator for plan library contraction which takes the earlier characteristics into account, and which balances the need for reactivity, the fragility of the plan library, and the correctness and overall performance of the agent.

The remainder of the paper is organised as follows. Preliminaries on BDI are given using CAN language in Section II. The plan library analysis is given in Section III. A novel BDI evolution architecture is presented in Section IV and Section V. Section VI discusses related work and concluded.

II. Preliminaries

An agent in the BDI framework is defined by its beliefs, desires, and intentions. The beliefs encode the understanding of the environment, the desires are those goals that an agent would like to accomplish, and the intentions those desires that the agent has committed to achieving.

CAN formalises the behaviour of a classical BDI agent, which is defined by a belief base $\mathcal{B}$ and a plan library Π. The belief base $\mathcal{B}$ is a set of formulas encoding the current beliefs (i.e. what the agent believes to be true)[1]. The plan library Π encodes a collection of plans of the form $e : \varphi \leftarrow P$ with e the event-goal, φ the pre-condition, and P the plan-body program. The plan $e : \varphi \leftarrow P$ encodes a strategy (i.e. plan body P) of reacting to the event-goal e if φ is believed to be true. When no ambiguity arises, we use the plan-body P to refer the whole plan rule, e_P the event-goal of the plan P, and φ_P the pre-condition of the plan P. The language used in the plan-body program P has the following

$$nil \mid act \mid !e \mid e : (|\psi_1 : P_1, \cdots, \psi_n : P_n|)$$

where nil is an empty program, act a primitive action, $!e$ a subgoal to achieve the event e, which is simply an event-goal combined with the syntactic label "!". We sometimes blur the distinction between event-goals e and subgoals $!e$. A set of relevant plans for an event-goal e is encoded by $e^P = \{\psi_1 : P_1, \cdots, \psi_n : P_n\}$. It means that, given an event-goal e, the agent will select the plan-body program P_i to resolve e when ψ_i is believed true ($1 \leq i \leq n$). When there are multiple plan-body programs P_i satisfied to achieve event-goal e, the agent will select one of them to pursue and only drop it when it either succeeds or fails. For simplicity, we consider a BDI agent system that is programmed relative to some finite propositional language. We also use standard mathematical symbols $\mathbb{N}$ to refer to the set of natural numbers and $\mathbb{R}_{\geq 0}$ the set of non-negative real numbers throughout this work.

[1]The belief base of a BDI agent is a classical belief base rather than e.g. a probabilistic belief base. Nonetheless, our ideas can extend to the latter (e.g. applying [16]).

III. Plan Library Analysis

In this section, we establish some measures to capture the characteristics of plans (e.g. the performance of plans, and the relationships between them) in a BDI agent system. This section will provide the foundations for understanding both how to compute them, and how they can be used for the library expansion in Section IV and contraction reasoning in Section V.

A. Measuring Performance of Plans

In this work, we use $\mathcal{P}$ to stand for a set of plans and $\mathcal{T}$ a set of time points. We start by introducing notation for plan execution as follows:

Definition 1. . *A function $\mathcal{S} : \mathcal{P} \times \mathcal{T} \rightarrow \{\top, \bot, \varnothing\}$ is called a status function.*

A status function records the success and failure of plans during agent execution while the agent is running. For example, $\mathcal{S}(P_1, 3) = \top$ means that plan P_1 succeeded at time point 3 while $\mathcal{S}(P_2, 5) = \bot$ says P_2 failed at time point 5. Finally, $\mathcal{S}(P, t) = \varnothing$ if it didn't succeed or fail at time point t (e.g. still in execution).

We now introduce the *execution frequency* of plans to measure how many times a plan has been completely executed.

Definition 2. *An execution frequency function $\Delta : \mathcal{P} \times \mathcal{T} \times \mathcal{T} \rightarrow \mathbb{N}$ is defined for each $P \in \mathcal{P}$ and each $t_1, t_n \in \mathcal{T}$ such that $t_1 \leq t_n$ as follows:*

$$\Delta(P, t_1, t_n) = |\{\mathcal{S}(P, t_i) \neq \varnothing \cdot\ i = 1, \cdots, n\}|$$

The *execution frequency* quantifies the number of times a plan has led to either success of failure over a given set of time points between t_1 and t_n. The *success rate* defined below, which is based on the *execution frequency*, captures the relative performance quality of each individual plan.

Definition 3. *A success rate for plan P is defined as:*

$$\Phi(P, t_1, t_n) = \frac{\Delta^s(P, t_1, t_n)}{\Delta(P, t_1, t_n)}$$

where $\Delta^s(P, t_1, t_n) = |\{\mathcal{S}(P, t_i) = \top \cdot\ i = 1, \cdots, n\}|$ stands for the successful execution frequency.

We stress that argument $\mathcal{T} \times \mathcal{T}$ allows the agent to have a capture of the quality of plans which can be based on both overall performance (i.e. $\Delta(P, t_0, t_n)$) and latest performance (e.g. $\Delta(P, t_{n-2}, t_n)$). Such timely capacity is vital as the realistic environment is highly dynamic. Simply having the overall *success rate* may prevent the agent from being aware of the recent abruptly growing failure of some plans.

B. Relationships Between Plans

We have introduced *execution frequency* and *success rate* to provide a performance abstraction of plans in BDI agents. However, it says nothing about the inherent structural characteristics (i.e. the relationships between plans) of these plans.

Recall that e^P is a set of relevant plans $\{P_1, \cdots, P_n\}$ for achieving an event-goal e. The first thing we are interested in is to know, and to compute the *relevancy* of individual plan P, i.e. how many alternative relevant plans a BDI agent possesses to responds to an event-goal e in different situations.

Definition 4. *A relevancy function $\Upsilon : \mathcal{P} \to \mathbb{N}$ is defined as:*

$$\Upsilon_{\mathcal{P}}(P) = |e^P| - 1$$

where $P \in e^P \subseteq \mathcal{P}$.

Besides the concept of *relevancy*, we are also interested in *replaceability*, which is when there are two or more plans applicable in the same situation to get the same result (i.e. post-effects). Intuitively, the "greater" the *replaceability* of a plan P is, the higher the chance that such a plan P can be recovered in the event of its failure, thus providing flexibility and robustness to the whole system.

We introduce the concept of what it means by being replaced and the degree of *replaceability* based on the work of [17] on overlapping and [18] on summary information. The overlap of P and $\{P_1, \cdots, P_n\}$, denoted as $\mathcal{O}(\{P, P_1, \cdots, P_n\})$ in [17], measures the number of situations (i.e. possible worlds) that both P and $\{P_1, \cdots, P_n\}$ can be applicable. It tells whether two or more plans can be applicable in the some same situations. For example, $\mathcal{O}(\{P, P_1, \cdots, P_n\}) \neq 0$ shows that the situation for both P and a set of plans $\{P, P_1, \cdots, P_n\}$ to be applicable exists. On the other hand, the summarised post-effects of a plan P (i.e. $post(P)$ denoted in [18]) provides a means to check if some plans can achieve the same result w.r.t. necessary and possible post-effects. The summarised necessary post-effects are those which are always true after successfully executing any decomposition of plans while possible post-effects are those that may result from some decomposition of plans. We use $post(\{P_1, \cdots, P_n\}) \models post(P)$ to represent that the post-effects of executions of $S = \{P_1, P_2, \cdots, P_n\}$ can ensure the post-effects of executions of P to be true. Therefore, we have

Definition 5. *A plan P can be replaced by a set of plans $S = \{P_1, P_2, \cdots, P_n\}$, denoted as $P \rhd_r S$, if the overlap of P and $\{P_1, \cdots, P_n\}$, $\mathcal{O}(\{P, P_1, \cdots, P_n\}) \neq 0$ and $post(\{P_1, \cdots, P_n\}) \models post(P)$. We also say that S can minimally replace P, denoted as $P \rhd_{mr} S$, iff $P \rhd_r S$ and $\forall P' \in S,\ P \not\rhd_r (S \setminus P')$.*

Finally, we define the degree of *replaceability* for P to be the number of sets of plans S that can *minimally* replace P.

Definition 6. *A degree of replaceability for plan P is a function $\Gamma_{\mathcal{P}} : \mathcal{P} \to \mathbb{N}$, defined as $\Gamma_{\mathcal{P}}(P) = |\{S \cdot P \rhd_{mr} S\}|$.*

We close the section by noting that what we have done so far is to define the relevant measures of BDI plans at the individual plan level. The extended measures for a given set of plans (e.g. a plan library) are the subject of following section.

C. Summary Information

We can now summarise both performance and structural information of an individual plan to characterise the plan library of a BDI agent system as a whole.

Firstly, we describe how the performance information (e.g. *execution frequency*) of each plan can be summarised to indicate the performance of a plan library. We apply a mean aggregation method to provide an average performance of a plan library.

Definition 7. *An execution frequence is a function $\Delta : 2^{\mathcal{P}} \times T \times T \to \mathbb{R}_{\geq 0}$ defined as follows:*

$$\Delta(\Pi, t_1, t_n) = \frac{\sum_{P \in \Pi} \Delta(P, t_1, t_n)}{|\Pi|}.$$

where $\Pi \subseteq 2^{\mathcal{P}}$ and $\Delta(P, t_1, t_n)$ refers to the execution frequency of plan P between time points t_1 and t_n.

Definition 8. *A success rate is a function $\Delta : 2^{\mathcal{P}} \times T \times T \to \mathbb{R}_{\geq 0}$ defined as follows:*

$$\Phi(\Pi, t_1, t_n) = \frac{\sum_{P \in \Pi} \Delta^s(P, t_1, t_n)}{|\{P \in \Pi \cdot \Delta(P, t_1, t_n) \neq 0\}|}$$

where $\Pi \subseteq 2^{\mathcal{P}}$, $\Phi(P, t_1, t_n)$ refers to success rate of plan P between time points t_1 and t_n, and $\Delta(P, t_1, t_n) \neq 0$ means P has to be executed at least once between time t_1 and t_n.

Secondly, we summarise the structural information of a plan library by counting how many event-goals a plan library accounts for. The intuition of it is that the capability of a BDI agent is essentially the amount of different types of event-goals it can handle. Therefore, we define the degree of *functionality* to formalise this intuition as follows:

Definition 9. *A degree of the functionality is a function $\Delta : 2^{\mathcal{P}} \times T \times T \to \mathbb{N}$ defined as follows:*

$$\mathcal{F}(\Pi) = |\{e_P \cdot P \in \Pi\}|$$

where $\varPi \in 2^{\mathcal{P}}$ and e_P is an event-goal of plan P.

Finally, we introduce four ordering relations corresponding to the three summaries we established above (i.e. *execution frequency*, *success rate*, and *functionality*), and one extra ordering relation based on *relevancy* and *replaceability* (recall that *relevancy* quantifies how many relevant plans to respond to the event-goal of a plan, while *replaceability* counts the number of sets of plans which are available to replace such a plan). The orderings are given as follows:

Definition 10. *A set of binary relations $\succeq$ over $2^{\mathcal{P}}$ w.r.t. execution frequency $\varDelta$, success rate $\varPhi$, functionality $\mathcal{F}$, and relevancy $\varUpsilon$ and replaceability $\varGamma$ measure, is a 4-tuple*

$$\langle \succeq_{activeness}, \succeq_{success}, \succeq_{functionality}, \succeq_{robustness} \rangle$$

where $\forall \varPi, \varPi' \in 2^{\mathcal{P}}$

- *$\varPi \succeq_{activeness} \varPi'$ iff $\varDelta(\varPi, t_1, t_n) \geq \varDelta(\varPi', t_1, t_n)$;*
- *$\varPi \succeq_{success} \varPi'$ iff $\varPhi(\varPi, t_1, t_n) \geq \varPhi(\varPi', t_1, t_n)$;*
- *$\varPi \succeq_{functionality} \varPi'$ iff $\mathcal{F}(\varPi) \geq \mathcal{F}(\varPi')$;*
- *$\varPi \succeq_{robustness} \varPi'$ iff $\nexists P \in \varPi$ s.t. $P \in \varPi'$, $\varUpsilon_{\varPi}(P) \leq \varUpsilon_{\varPi'}(P)$, and $\varGamma_{\varPi}(P) \leq \varGamma_{\varPi'}(P)$;*

We have that $\varPi \simeq \varPi'$ if $\varPi \succeq \varPi'$ and $\varPi' \succeq \varPi$ while $\varPi \succ \varPi'$ if $\varPi \succeq \varPi'$ and $\varPi \not\preceq \varPi'$. For a plan library $\varPi$, if $\varPi$ has a higher *execution frequency* than $\varPi'$, denoted as $\varPi \succeq_{activeness} \varPi'$, then it is interpreted as that $\varPi$ is believed to be more active than $\varPi'$. The second ordering $\varPi \succeq_{success} \varPi'$ means that $\varPi$ has a higher *success rate* than $\varPi'$, and $\varPi$ is believed to be more successful than $\varPi'$. The third ordering $\varPi \succeq_{functionality} \varPi'$ means that $\varPi$ can respond to more types of event-goals than $\varPi'$ can. The fourth ordering $\varPi \succeq_{robustness} \varPi'$ shows that for every plan $P \in \varPi$, $\varPi$ has both more relevant and replaceable plans for P than $\varPi'$ does.

Now that we have defined all relevant measures, we look into how we can expand and contract a plan library in a sensible and predictable way.

IV. Plan Library Expansion

In this section, we propose some postulates for a plan library expansion. The approach of postulates has amounted to many seminal works in the community of belief revision and contraction such as [19] and [20], and belief merging [21]. For illustration and simplicity, we will first consider using a single plan to represent inputs, and then extend to the general case where we use any set of plans to represent general inputs.

A. Formal Expansion Framework

We start with the definition of an expansion operator $\circ$:

Given a plan library $\varPi$ and a plan P, $\varPi \circ P$ denotes the expansion of $\varPi$ by P with $\circ$ if and only if it satisfies the following postulates:

EO1 $\varPi \circ P$ is a plan library.

This postulate ensures that the expansion is still a plan library.

EO2 $P \in \varPi \circ P$ and $\varPi \subseteq \varPi \circ P$.

This postulate states that the new plan is obtained after the expansion and the result of plan library expansion $\varPi \circ P$ indeed subsumes the knowledge of the previous plan library $\varPi$.

EO3 If $P \in \varPi$, then $\varPi \circ P = \varPi$.

This postulate indicates that the plan library expansion $\varPi \circ P$ should only consider a new plan P which is initially not included in $\varPi$.

EO4 $(\varPi \circ P) \circ P' = (\varPi \circ P') \circ P$ for any plan P and P'.

This postulate indicates that the order of inputs (received) should not influence the outcome of the expansion.

Proposition 1. *If an operator $\circ$ satisfies **EO1**, **EO2**, and **EO4**, we have $\varPi \circ \{P, P'\} = (\varPi \circ P) \circ P' = (\varPi \circ P') \circ P$.*

This proposition shows that the expansion of a set of plans is equivalent to a sequence of expansions by a single plan. Now we give the following representation theorem for these postulates.

Theorem 1. *Given an operator $\circ$, $\varPi \circ P$ satisfies **EO1-E04** precisely when*

$$\varPi \circ P \succeq_{functionality} \varPi \text{ and } \varPi \circ P \succeq_{robustness} \varPi.$$

This theorem formally confirms that the expansion of a plan library $\varPi$ by P will never cause a decrease of *functionality* or *robustness* (i.e. more event-goals can be responded to, and more relevant or replaceable plans may be adopted).

Finally, in order to extend these postulates to the case that new input is not restricted to only one plan, we simply need to replace a single input plan P with a set of plans $\mathcal{P}$.

V. Plan Library Contraction

In this section, we give a principled definition of a plan library contraction operator. We then present a concrete instantiation of such an operator in a Mars Rover scenario. We then close this section by showing that this instantiation satisfies the postulates of a contraction operator.

A. Formal Contraction Framework

We start with the definition of a contraction operator ∇: Given a plan library $\varPi$, $\nabla(\varPi)$ denotes the contraction of $\varPi$ by ∇ iff it satisfies the following postulates:

CO1 $\nabla(\varPi)$ is a plan library.

This postulate ensures the result of contraction is a plan library.

CO2 $\nabla(\varPi) \subseteq \varPi$.

This postulate states that the result of a contraction operator is a subset of the original plan library.

CO3 Given a set of plans $\mathcal{P}$, if $\mathcal{P} \subseteq \varPi \setminus \nabla(\varPi)$ and $\mathcal{P} \subseteq \varPi' \subseteq \varPi$, then $\mathcal{P} \subseteq \varPi' \setminus \nabla(\varPi')$.

This relativity postulate states that if a set of plans $\mathcal{P}$ are contractible in the plan library $\varPi$ (i.e. $\mathcal{P} \subseteq \varPi \setminus \nabla(\varPi)$), then they must be deemed as contractible in any subset $\varPi'$ (i.e. $\mathcal{P} \subseteq \varPi' \setminus \nabla(\varPi')$) which includes them (i.e. $\mathcal{P} \subseteq \varPi' \subseteq \varPi$).

CO4 $\nabla(\varPi) \succeq \varPi$ where $\succeq \in \{\succeq_{activeness}, \succeq_{success}\}$.

This postulate restricts the behaviour of the contraction by saying that the contraction $\nabla(\Pi)$ should not witness the decrease of both *execution frequency* and *success rate* of Π.

CO5 $\forall P \in \Pi \setminus \nabla(\Pi)$, then $\Gamma_{\nabla(\Pi)}(P) > 0$.

This postulate takes care of the fragility of the plan library by ensuring that there are still plans left in $\nabla(\Pi)$ which can replace deleted plan P.

With these postulates, we have the following results that characterise contraction operators that satisfy some of postulates **CO1-CO5**.

Proposition 2. *If an operator ∇ satisfies **CO1-CO3**, then the following holds:*

(1) $\nabla(\Pi') \subseteq \nabla(\Pi)$ *if* $\Pi' \subseteq \Pi$ — *ordered set inclusion*

(2) $\nabla(\Pi \cap \Pi') \subseteq \nabla(\Pi) \cap \nabla(\Pi')$ — *intersection set inclusion*

(3) $\nabla(\Pi \setminus \Pi') \subseteq \nabla(\Pi) \setminus \nabla(\Pi')$ — *difference set inclusion*

(4) $\nabla(\Pi) \cup \nabla(\Pi') \subseteq \nabla(\Pi \cup \Pi')$ — *union set inclusion*

Proof. The first condition shows that the contraction preserves the set inclusion order. If $(\Pi \setminus \nabla(\Pi)) \cap \Pi' = \emptyset$, then we have $\Pi' \subseteq \nabla(\Pi)$. Given **CO2** (i.e. $\nabla(\Pi') \subseteq \Pi'$), we have $\nabla(\Pi') \subseteq \nabla(\Pi)$. If $(\Pi \setminus \nabla(\Pi)) \cap \Pi' = \mathcal{P} \neq \emptyset$, we have $\Pi' \setminus \mathcal{P} \subseteq \nabla(\Pi)$. Given **CO3**, we have $\mathcal{P} \subseteq \Pi' \setminus \nabla(\Pi')$, which means $\nabla(\Pi') \subseteq \Pi' \setminus \mathcal{P}$. Hence we have $\nabla(\Pi') \subseteq \nabla(\Pi)$. In the second condition, it shows that the contraction on intersection of Π and Π' are a subset of the intersection of the contraction results of $\nabla(\Pi)$ and $\nabla(\Pi')$. Notice $\Pi \cap \Pi'$ is a subset of both Π and Π'. Therefore, $\nabla(\Pi \cap \Pi') \subseteq \nabla(\Pi)$ and $\nabla(\Pi \cap \Pi') \subseteq \nabla(\Pi')$ according to the first condition. Hence $\nabla(\Pi \cap \Pi') \subseteq \nabla(\Pi) \cap \nabla(\Pi')$. Similar arguments can be give to the third and fourth condition by noticing $\Pi \setminus \Pi' \subseteq \Pi$ (i.e. difference set inclusion) and $\Pi' \subseteq \Pi$ in third condition, and $\Pi \subseteq \Pi \cup \Pi'$ (i.e. union set inclusion) and $\Pi' \subseteq \Pi \cup \Pi'$ in fourth condition. $\square$

B. Instantiation of Plan Library Contraction

In this section, a concrete multi-criteria argumentation-based decision making is proposed to instantiate the abstract contraction operator presented in Section V-A. We stress though that the purpose of this instantiation is not to signify its supremacy over other potential instantiations, but rather to verify the existence and feasibility of our contraction operator. Also, the benchmark comparison of different instantiated contraction operators is beyond the scope of this work.

The multi-criteria argumentation-based decision making is a general-purpose decision framework which combines the multi-criteria decision [22] with knowledge based qualitative argumentation theory [23]. Argumentation serves to support or attack whether a particular candidate is better than another based on knowledge processed by an agent. The framework employed in this work is formally stated in [24] and is conceptually composed by three components, namely $\langle X, \mathcal{K}, \mathcal{R}\rangle$. The first component X is the set of all possible candidates (e.g. a set of plans) presented to the decision maker. The second component is epistemic knowledge $\mathcal{K}$, denoted as a 5-tuple $\langle \mathcal{C}, >_{\mathcal{C}}, \lambda, \varpi, ACC\rangle$. It allows the agent to reason and compare candidates among each other and decide which is/are best candidate/s to be chosen. A set of non-cyclic (i.e. linear) criteria $\mathcal{C}$ (e.g. *success rate*) is used to compare the elements in X. The strict order of the element of $\mathcal{C}$, denoted as $>_{\mathcal{C}}$, is given such that $(C_i, C_j) \in \mathcal{C}$ means that the criteria C_i is preferred than C_j. In order to quantify the linear preference, each $C_i \in \mathcal{C}$ has a subinterval that states the preference among all criteria (e.g. $C_1 =$ *execution frequency* $\sim [0, 0.25]$). A set of clauses (ζ, α) is computed where ζ in the form of $q \leftarrow p_1 \wedge \cdots \wedge p_k$ $(k \geq 0)$ says the conclusion q is supported by $p_1 \wedge \cdots \wedge p_k$, where $q, p_1, \cdots, p_k$ are literals, and $\alpha \in [0, 1]$ which express a low bound for the necessity degree of ζ. A set of uncertain clauses (i.e. $\alpha \in (0, 1)$) is denoted as λ while the set of certain clauses (i.e. $\alpha = 1$) denoted as ϖ. Uncertain clauses with the same conclusion will be combined to form arguments. A use-specified aggregation function ACC aggregates necessity degrees of arguments which support a same conclusion q to build accrued structures. Finally, decision rules $\mathcal{R}$ will be used to select final candidates, denoted as $\Omega(\langle X, \mathcal{K}, \mathcal{R}\rangle)$, based on those accrued structures. There are two decision rules[2]:

DR1 : $\{W\} \overset{X}{\Leftarrow} \{$*ws(W,Y)*$\}$, *not*$\{$*ws(Z, W)*$\}$.

A candidate $W \in X$ will be chosen if *W* is worse (ws) than another candidate *Y* and no *Z* exists which is worse than *W*.

DR2 : $\{W, Y\} \overset{X}{\Leftarrow} \{$*sm(W,Y)*$\}$, *not*$\{$*ws(Z, W)*$\}$.

Both *W* and $Y \in X$, deemed as equivalently bad i.e. same (sm), will be chosen if there is no *Z* which is worse than *W* and *Y*.

Following the methodology of the formalism in [24], we consider a BDI agent which has a set of plan $\mathcal{P}$ and a set of criteria $\mathcal{C} = \{C_1, C_2, C_3, C_4\}$ where $C_1 = \Delta(P, t_0, t_{current})$ is the overall *execution frequency* from initial time point t_0 to current time point $t_{current}$ (i.e. the moment the plan contraction starts), $C_2 = \Delta(P, t', t_{current})$ the latest *execution frequency* of P from a chosen recent time point t' to $t_{current}$, C_3 the overall *success rate*, and C_4 the the latest *success rate*.

In this work, we assume that the agent prioritises the *success rate* criteria over *execution frequency* criteria and prefer the latest information. Therefore, the criteria order $>_{\mathcal{C}} = \{(C_4, C_3), (C_2, C_1), (C_4, C_2), (C_3, C_1)\}$. Since plans that are without any replaceable plans cannot be deleted and the formalism in [24] is not concerned with how the possible candidates are obtained to present to the decision-maker, we will employ postulate **CO5** to filter these plans out. Therefore, we define a specific contraction operator, denoted as ∇^{abm}, and propose a generic algorithm which implements ∇^{abm} as shown in Algorithm 1:

Definition 11. *Let $\mathcal{P}$ be a set of plans. We define a contraction operator $\nabla^{abm} = \Omega(\langle X, \mathcal{K}, \mathcal{R}\rangle)$ where $\Omega(\langle X, \mathcal{K}, \mathcal{R}\rangle)$ is the selected candidates of decision problem $\langle X, \mathcal{K}, \mathcal{R}\rangle$ defined previously, and $X = \{P \in \mathcal{P} \mid \Gamma_{\mathcal{P}}(P) > 0, C_4(P) < \tau\}$*

[2] We modify the rules to choose the worse (ws) plans compared to the original work on finding the better ones.

where τ represents the success rate tolerance threshold and $C_4(P)$ the value of criteria C_4 (i.e. latest success rate) of P.

The set of all possible candidates X will not include any plans which do not have replaceable plans (i.e. $\mathit{\Gamma}_{\mathcal{P}}(P) > 0$) and only have plans with success rates lower than τ for potential removal (i.e. $C_4(P) < \tau$) shown in step 2 of Algorithm 1. In the following section, all the concepts involved in the multi-criteria argumentation based decision making in [24] will be exemplified in a Mars Rover example to demonstrate how it can effectively assist the selection of plans.

Algorithm 1 Computation for Contraction Operator ∇^{abm}

1: **function** $\nabla^{abm}(\langle X,\ \mathcal{C}, >_{\mathcal{C}},\ \mathcal{R}\rangle)$
2: $\quad X = \{P \in \mathcal{P} \mid \mathit{\Gamma}_{\mathcal{P}}(P) > 0, C_4(P) < \tau\}$ ▷ filtering
3: $\quad \mathcal{C} = \{C_1, C_2, C_3, C_4\}$ defined previously
4: $\quad >_{\mathcal{C}} = \{(C_4, C_3), (C_2, C_1), (C_4, C_2), (C_3, C_1)\}$
5: $\quad$ Compute uncertain and certain clauses (λ, ϖ)
6: $\quad$ Build arguments (defined in [24])
7: $\quad$ Apply rules $\mathcal{R}$ to select the acceptable candidates
8: **return** solution of selection

C. Planetary Vehicle Example

One of the missions of Mars Rover is to use scientific instruments mounted to the robotic arm of the Rover to investigate and analyse Martian terrain. This requires the Rover to drive up to a designated target (i.e. terrain navigation plan), position themselves to reach the target (i.e. Rover positioning plan), and deploy the arm onto the target to perform the investigation (i.e. arm deployment plan). After remembering several routes from the navigation planner it took to a designated crater wall, the Rover has plans P_1, P_2, and P_3 to navigate to it again if needed. Plan P_4 and P_5 are Rover positioning and arm deployment plan, respectively. Consider the set of plans $\mathit{\Pi} = \{P_1, P_2, P_3, P_4, P_5\}$ where *replaceability* $\mathit{\Gamma}_{\mathit{\Pi}}(P_1) = \mathit{\Gamma}_{\mathit{\Pi}}(P_2) = \mathit{\Gamma}_{\mathit{\Pi}}(P_3) = |\{P_1, P_1, P_3\}| - 1 = 2$, $\mathit{\Gamma}_{\mathit{\Pi}}(P_4) = \mathit{\Gamma}_{\mathit{\Pi}}(P_5) = |\{P_4\}| - 1 = |\{P_5\}| - 1 = 0$, and $\mathcal{C} = \{C_1, C_2, C_3, C_4\}$ where C_1 is the overall *execution frequency* (**oef**), C_2 the latest *execution frequency* (**lef**), C_3 overall *success rate* (**osf**), and C_4 latest *success rate* (**lsr**). We set the lower bound tolerant success rate threshold $\tau = 0.85$.

Table I shows the possible candidates (P_4 and P_5 are filtered out due to no replaceable plans available) and their respective values for each criterion (in C_1, C_2, C_3, and C_4) and their respective values normalised to interval $[0, 1]$ (in C'_1, C'_2, C'_3, and C'_4). Likewise, following the approach from [24], a set of uncertain and certain clauses $\langle\lambda, \varpi\rangle$ can be computed (explained shortly after) and shown as follow:

Candidates	C_1	C_2	C_3	C_4	C'_1	C'_2	C'_3	C'_4
P_1	80	70	0.8	0.5	1	1	1	0.63
P_2	20	5	0.6	0.8	0.25	0.07	0.75	1
P_3	70	10	0.7	0.75	0.88	0.14	0.88	0.94

TABLE I: Criterion Values and Normalised Criterion Values

$$\lambda = \left\{ \begin{array}{c} (\mathit{oef}(P_2, P_1), 0.19) \quad (\mathit{ws(W,Y)} \leftarrow \mathit{oef}(W, Y), 0.24) \\ (\mathit{oef}(P_2, P_3), 0.16) \quad (\neg\mathit{ws(W,Y)} \leftarrow \mathit{oef}(W, Y), 0.24) \\ (\mathit{oef}(P_3, P_1), 0.03) \quad (\mathit{ws(W,Y)} \leftarrow \mathit{lef}(W, Y), 0.49) \\ (\mathit{lef}(P_2, P_1), 0.48) \quad (\neg\mathit{ws(W,Y)} \leftarrow \mathit{lef}(W, Y), 0.49) \\ (\mathit{lef}(P_2, P_3), 0.27) \quad (\mathit{ws(W,Y)} \leftarrow \mathit{osf}(W, Y), 0.74) \\ (\mathit{lef}(P_3, P_1), 0.47) \quad (\neg\mathit{ws(W,Y)} \leftarrow \mathit{osf}(W, Y), 0.74) \\ (\mathit{osr}(P_2, P_1), 0.56) \quad (\mathit{ws(W,Y)} \leftarrow \mathit{lsr}(W, Y), 0.99) \\ (\mathit{osr}(P_3, P_1), 0.53) \quad (\neg\mathit{ws(W,Y)} \leftarrow \mathit{lsr}(W, Y), 0.99) \\ (\mathit{osr}(P_2, P_3), 0.53) \quad (\mathit{lsr}(P_1, P_2), 0.84) \\ (\mathit{lsr}(P_1, P_3), 0.83) \quad (\mathit{lsr}(P_3, P_2), 0.77) \end{array} \right\}$$

$$\varpi = \{(\neg\mathit{ws}(W, Y) \leftarrow \mathit{sm}(W, Y), 1)\ (\neg\mathit{ws}(W, Y) \leftarrow \mathit{sm}(Y, W), 1)\}$$

The necessity degrees of the clauses belonging to (λ, ϖ) above were calculated as follows. **Step 1**: Normalise the criteria values (see C_i) to interval $[0, 1]$ for all of the criteria (see C'_i). **Step 2**: Compare the candidates among each other regarding the normalised criteria. The candidate which appears as first argument has a worse criteria value than the one that appears as second argument. The necessity degree of the clause is calculated as the absolute value of the remainder of the normalised criteria values. **Step 3**: Divide the necessity degree obtained in previous step by the number of criteria provided to the decision maker, i.e. by 4 in this case. **Step 4**: Assign the subinterval to each criteria according to $>_{\mathcal{C}}$, i.e. $C_1 = \mathit{oef} \sim [0, 0.25]$, $C_2 = \mathit{lef} \sim [0.25, 0.5]$, $C_3 = \mathit{osr} \sim [0.5, 0.75]$, and $C_4 = \mathit{lsr} \sim [0.75, 1]$. **Step 5**: Map the necessity degrees obtained in the previous step to the subinterval assigned to the criteria in the clause. **Step 6**: For each clause (ζ, α) such that ζ is a rule of either $\mathit{ws}(W, Y) \leftarrow C_i(W, Y)$ or $\neg\mathit{ws}(W, Y) \leftarrow C_i(W, Y)$, we set α to be the upper bound value of the subinterval assigned to C_i where $i \in \{1, 2, 3, 4\}$.

The arguments of the form $\mathcal{A} = \langle u, h, \alpha\rangle$ is presented in Figure 1 and is built from the uncertain clause program in which u is a set of uncertain clauses from λ, h the conclusion it supports (e.g. *ws(W,Y)*), and α its necessity degree. Finally, we aggregate the arguments which support the same conclusion h into accrued structures. For example, $\mathcal{A}_1$, $\mathcal{A}_7$, and $\mathcal{A}_{13}$ support the same conclusion $\mathit{ws}(P_2, P_1)$ to build the accrued structure $[\Psi_1, \mathit{ws}(P_2, P_1), 0.82]$ in which $\Psi_1 = \mathcal{A}_1 \cup \mathcal{A}_7 \cup \mathcal{A}_{13}$. To calculate the necessity values for accrued structures, it will use the ACC function defined below, with $K = 0.1$:

$$ACC(\alpha_1, \cdots, \alpha_n) = [1 - \prod(1 - \alpha_i)] + K max(\alpha_1, \cdots, \alpha_n) \prod(1 - \alpha_i)$$

As it can be observed, 12 aggregated arguments shown in Figure 1 can be built to support the reasons by which a candidate should be deemed worse than another one. Those aggregated arguments warranted (shown in bold) because of their greater necessity values than their negated counterparts will be used to compute the final chosen candidate(s) with decision rules $\mathcal{R}$. In this particular case, only decision rule *DR1* can be applied. For candidate P_1, precondition of *DR1* can be warranted and there is no warranted accrued structure supporting a conclusion of the kind $\mathit{ws}(Z, P_1)$ to warrant the restriction of rule *DR1*, hence P_1 becomes the selected candidate for deletion.

Choosing plan P_1 is quite evident for the so-called human common sense reasoning since it has the worst latest *success rate* (i.e. 0.5) which is most important preference criterion, despite having a best overall *success rate*. The worst latest *success rate* may imply that P_1 is no longer suitable for the current Martian surface navigation task, thus ready to be dropped by the Rover.

$$\mathcal{A}_1 = \langle\{(\mathit{ws}(P_2, P_1) \leftarrow \mathit{oef}(P_2, P_1), 0.24), (\mathit{oef}(P_2, P_1), 0.19)\}, \mathit{ws}(P_2, P_1), 0.19\rangle$$

$$\mathcal{A}_2 = \langle\{(\neg\mathit{ws}(P_1, P_2) \leftarrow \mathit{oef}(P_1, P_2), 0.24), (\mathit{oef}(P_1, P_2), 0.19)\}, \neg\mathit{ws}(P_1, P_2), 0.19\rangle$$

$$\mathcal{A}_3 = \langle\{(\mathit{ws}(P_2, P_3) \leftarrow \mathit{oef}(P_2, P_3), 0.24), (\mathit{oef}(P_2, P_3), 0.19)\}, \mathit{ws}(P_2, P_3), 0.16\rangle$$

$$\mathcal{A}_4 = \langle\{(\neg\mathit{ws}(P_3, P_2) \leftarrow \mathit{oef}(P_3, P_2), 0.24), (\mathit{oef}(P_3, P_2), 0.19)\}, \neg\mathit{ws}(P_3, P_2), 0.16\rangle$$

$$\mathcal{A}_5 = \langle\{(\mathit{ws}(P_3, P_1) \leftarrow \mathit{oef}(P_3, P_1), 0.24), (\mathit{oef}(P_3, P_1), 0.03)\}, \mathit{ws}(P_3, P_1), 0.03\rangle$$

$$\mathcal{A}_6 = \langle\{(\neg\mathit{ws}(P_1, P_3) \leftarrow \mathit{oef}(P_1, P_3), 0.24), (\mathit{oef}(P_1, P_3), 0.03)\}, \neg\mathit{ws}(P_1, P_3), 0.03\rangle$$

$$\mathcal{A}_7 = \langle\{(\mathit{ws}(P_2, P_1) \leftarrow \mathit{lef}(P_2, P_1), 0.49), (\mathit{lef}(P_2, P_1), 0.48)\}, \mathit{ws}(P_2, P_1), 0.48\rangle$$

$$\mathcal{A}_8 = \langle\{(\neg\mathit{ws}(P_1, P_2) \leftarrow \mathit{lef}(P_1, P_2), 0.49), (\mathit{lef}(P_1, P_2), 0.48)\}, \neg\mathit{ws}(P_1, P_2), 0.48\rangle$$

$$\mathcal{A}_9 = \langle\{(\mathit{ws}(P_2, P_3) \leftarrow \mathit{lef}(P_2, P_3), 0.49), (\mathit{lef}(P_2, P_3), 0.27)\}, \mathit{ws}(P_2, P_3), 0.27\rangle$$

$$\mathcal{A}_{10} = \langle\{(\neg\mathit{ws}(P_3, P_2) \leftarrow \mathit{lef}(P_3, P_2), 0.49), (\mathit{lef}(P_3, P_2), 0.27)\}, \neg\mathit{ws}(P_3, P_2), 0.27\rangle$$

$$\mathcal{A}_{11} = \langle\{(\mathit{ws}(P_3, P_1) \leftarrow \mathit{lef}(P_3, P_1), 0.49), (\mathit{lef}(P_3, P_1), 0.47)\}, \mathit{ws}(P_3, P_1), 0.47\rangle$$

$$\mathcal{A}_{12} = \langle\{(\neg\mathit{ws}(P_1, P_3) \leftarrow \mathit{lef}(P_1, P_3), 0.49), (\mathit{lef}(P_1, P_3), 0.47)\}, \neg\mathit{ws}(P_1, P_3), 0.47\rangle$$

$$\mathcal{A}_{13} = \langle\{(\mathit{ws}(P_2, P_1) \leftarrow \mathit{osr}(P_2, P_1), 0.74), (\mathit{osr}(P_2, P_1), 0.56)\}, \mathit{ws}(P_2, P_1), 0.56\rangle$$

$$\mathcal{A}_{14} = \langle\{(\neg\mathit{ws}(P_1, P_2) \leftarrow \mathit{osr}(P_1, P_2), 0.74), (\mathit{osr}(P_1, P_2), 0.56)\}, \neg\mathit{ws}(P_1, P_2), 0.56\rangle$$

$$\mathcal{A}_{15} = \langle\{(\mathit{ws}(P_3, P_1) \leftarrow \mathit{osr}(P_3, P_1), 0.74), (\mathit{osr}(P_3, P_1), 0.53)\}, \mathit{ws}(P_3, P_1), 0.53\rangle$$

$$\mathcal{A}_{16} = \langle\{(\neg\mathit{ws}(P_1, P_3) \leftarrow \mathit{osr}(P_1, P_3), 0.74), (\mathit{osr}(P_1, P_3), 0.53)\}, \neg\mathit{ws}(P_1, P_3), 0.53\rangle$$

$$\mathcal{A}_{17} = \langle\{(\mathit{ws}(P_2, P_3) \leftarrow \mathit{osr}(P_2, P_3), 0.74), (\mathit{osr}(P_2, P_3), 0.53)\}, \mathit{ws}(P_2, P_3), 0.53\rangle$$

$$\mathcal{A}_{18} = \langle\{(\neg\mathit{ws}(P_3, P_2) \leftarrow \mathit{osr}(P_3, P_2), 0.74), (\mathit{osr}(P_3, P_2), 0.53)\}, \neg\mathit{ws}(P_3, P_2), 0.53\rangle$$

$$\mathcal{A}_{19} = \langle\{(\mathit{ws}(P_1, P_2) \leftarrow \mathit{lsr}(P_1, P_2), 0.99), (\mathit{lsr}(P_1, P_2), 0.84)\}, \mathit{ws}(P_1, P_2), 0.84\rangle$$

$$\mathcal{A}_{20} = \langle\{(\neg\mathit{ws}(P_2, P_1) \leftarrow \mathit{lsr}(P_1, P_2), 0.99), (\mathit{osr}(P_1, P_2), 0.84)\}, \neg\mathit{ws}(P_1, P_2), 0.84\rangle$$

$$\mathcal{A}_{21} = \langle\{(\mathit{ws}(P_1, P_3) \leftarrow \mathit{lsr}(P_1, P_3), 0.99), (\mathit{lsr}(P_1, P_3), 0.83)\}, \mathit{ws}(P_1, P_3), 0.83\rangle$$

$$\mathcal{A}_{22} = \langle\{(\neg\mathit{ws}(P_3, P_1) \leftarrow \mathit{lsr}(P_3, P_1), 0.99), (\mathit{lsr}(P_3, P_1), 0.83)\}, \neg\mathit{ws}(P_3, P_1), 0.83\rangle$$

$$\mathcal{A}_{23} = \langle\{(\mathit{ws}(P_3, P_2) \leftarrow \mathit{lsr}(P_3, P_2), 0.99), (\mathit{lsr}(P_3, P_2), 0.77)\}, \mathit{ws}(P_3, P_2), 0.77\rangle$$

$$\mathcal{A}_{24} = \langle\{(\neg\mathit{ws}(P_2, P_3) \leftarrow \mathit{lsr}(P_2, P_3), 0.99), (\mathit{lsr}(P_2, P_3), 0.77)\}, \neg\mathit{ws}(P_2, P_3), 0.77\rangle$$

$$[\Psi_1, \mathit{ws}(P_2, P_1), 0.82], \boldsymbol{[\Psi'_1, \neg \mathit{ws}(P_2, P_1), 0.85]}, \Psi_1 = \mathcal{A}_1 \cup \mathcal{A}_7 \cup \mathcal{A}_{13}, \Psi'_1 = \mathcal{A}_{20}$$

$$[\Psi_2, \neg\mathit{ws}(P_1, P_2), 0.82], [\boldsymbol{\Psi'_2, \mathit{ws}(P_1, P_2), 0.85}], \Psi_2 = \mathcal{A}_2 \cup \mathcal{A}_8 \cup \mathcal{A}_{14}, \Psi'_2 = \mathcal{A}_{19}$$

$$[\Psi_3, \mathit{ws}(P_2, P_3), 0.73], [\boldsymbol{\Psi'_3, \neg \mathit{ws}(P_2, P_3), 0.79}], \Psi_3 = \mathcal{A}_3 \cup \mathcal{A}_9 \cup \mathcal{A}_{17}, \Psi'_3 = \mathcal{A}_{24}$$

$$[\Psi_4, \neg\mathit{ws}(P_3, P_2), 0.73], [\boldsymbol{\Psi'_4, \mathit{ws}(P_3, P_2), 0.79}], \Psi_4 = \mathcal{A}_4 \cup \mathcal{A}_{10} \cup \mathcal{A}_{18}, \Psi'_4 = \mathcal{A}_{23}$$

$$[\Psi_5, \mathit{ws}(P_3, P_1), 0.77], [\boldsymbol{\Psi'_5, \neg \mathit{ws}(P_3, P_1), 0.84}], \Psi_5 = \mathcal{A}_5 \cup \mathcal{A}_{11} \cup \mathcal{A}_{15}, \Psi'_5 = \mathcal{A}_{22}$$

$$[\Psi_6, \neg\mathit{ws}(P_1, P_3), 0, 77], [\boldsymbol{\Psi'_6, \mathit{ws}(P_1, P_3), 0.84}], \Psi_6 = \mathcal{A}_6 \cup \mathcal{A}_{12} \cup \mathcal{A}_{16}, \Psi'_6 = \mathcal{A}_{21}$$

Fig. 1: Arguments and accrued structures

D. Theorem

Theorem 2. *$\nabla^{abm} = \Omega(\langle X, \mathcal{K}, \mathcal{R}\rangle)$ is indeed a contraction operator ∇ satisfying **CO1-CO5***

Sketch. Postulates **CO1** (i.e. $\nabla^{abm}(\Pi)$ is a plan library) and **CO2** (i.e. $\nabla^{abm}(\Pi) \subseteq \Pi$) hold as plans are simply selected and removed from the original plan library Π.

Postulate **CO3** (i.e. relativity of contraction) holds for ∇^{abm} due to two computation steps of uncertain clauses λ. The normalisation of criteria values (in **Step 1**) and the absolute value of the remainder of the normalised criteria values (in **Step 2**) imply that a plan is deemed worse than the others is in a relative sense in a given set of plans.

Postulate **CO4** says that the contraction should not witness the decrease of both *activeness* and *success rate* of the plan library. It holds for ∇^{abm} because the selected plans to be removed are those which are deemed worse either in *success rate* criterion or both *success rate* and *execution frequency* criteria than other plans. Therefore, at least *success rate* of all plans will be increased after contraction. Hence **CO4** holds.

Postulate **CO5** (i.e. the protection of fragility of the plan library) holds for ∇^{abm} because we exclude plans which do not have any replaceable plans beforehand show in step 2 in Algorithm 1. Therefore, there are still plans which can replace deleted plans after the contraction. $\square$

VI. Related Work

Many works have tried to overcome the drawbacks associated with pre-defined plan libraries in BDI implementations. The first branch of BDI research tackles the inadequacy of a fixed plan library. A large number of works integrated automated planning techniques into BDI agents to generate plans at runtime, as surveyed by Meneguzzi and De Silva [9]. Only a few works start challenging the static nature of plan library and pursue towards augmenting the range of behaviours by expanding the set of BDI plans such as [15]. We are not aware of any work which attempts to contract the set of plans as we do here in this work.

Another branch of BDI research accepts the fixedness of plan library and works on the refinements of the BDI agent reasoning. For example, the work of [4] accommodates the hierarchical planning as an advanced plan selection engine to avoid potential troublesome execution sequences by looking ahead rather than simply selecting one applicable pre-defined plan. Another noticeable work [25] proposed a plan selection strategy that chooses a set of plans that fulfills the maximum number of goals while maintaining context consistency and resource-tolerance among the chosen plans. The argumentation based decision making framework also seems suitable to refine the deliberation reasoning of beliefs and desires of BDI agents in works of [26] and [27].

The summary information based agent reasoning in BDI has also yielded many promising results for the agent reasoning. The work of [14], [28], [29] presented algorithms for deriving summary information from the pre-defined plan library in a multi-agent system. That derived knowledge is then used to synthesize "abstract plans" or coordinate the activities of the agents at run time. Another similar summary work [30] focuses on the single agent to exploit positive interactions between goals and to avoid negative interactions. Furthermore, the work of [18] extends both works on summarisation discussed above with an account of non-deterministic actions, and with support for specifying actions and goals within a single plan that can execute concurrently.

The measure techniques in BDI agent systems have also been intensively studied. The work of [31] measures the cost of a plan while [17] measures the overage and overlap of a plan. Both of them seek to use these measures to find either cheapest or most likely-to-be successful plans and propose them to the agent. The measure approach is also well appreciated in the knowledgebase community. For instance, [32] provides a comprehensive review of the measures of information in developing intelligent systems that can tolerate inconsistencies when reasoning with real-world knowledge.

VII. CONCLUSION

In this paper, we described measures that characterise the performance and structure of plans, and provided rationales to guide the process of new plan adoption (i.e. plan expansion) and unsuitable plan deletion (i.e. plan contraction). These measures are based on useful concepts in BDI, such as how plans relate to each other and how the agent performs when dealing with specific event-goals. The merit of this paper is that we are one of the first works which formally define an evolving capacity of the BDI agents through changes to the set of existing BDI plans. Furthermore, the measure strategies and rationales we provide in this work are generic and do not require additional information from the BDI agent developer or the domain, beyond what is required in the typical BDI agent development. However, we recognise that since the plan library is assumed to evolve at real-time, it poses the challenge of an online calculation mechanism which is not yet given in this work. We also acknowledge that we have not yet implemented the reasoning described, based on these measures and postulates, and so do not yet have experimental results of their optimal values, which we believe to be application-specific. For future work, we plan to develop complete algorithms to use in BDI implementations that allow to compute the relevant measures in an online fashion. More concretely, instead of redoing the whole calculation every time, we want the agent to calculate measures based on previous results, and newly adopted and deleted plans.

REFERENCES

[1] A. S. Rao and M. P. Georgeff, "An abstract architecture for rational agents." in *Proceedings of Knowledge Representation and Reasoning*, 1991, pp. 439–449.

[2] A. S. Rao, "AgentSpeak (L): BDI agents speak out in a logical computable language," in *European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, 1996, pp. 42–55.

[3] M. Winikoff, L. Padgham, J. Harland, and J. Thangarajah, "Declarative and procedural goals in intelligent agent systems," in *International Conference on Principles of Knowledge Representation and Reasoning*, 2002, pp. 470 – 481.

[4] S. Sardina, L. De Silva, and L. Padgham, "Hierarchical planning in BDI agent programming languages: A formal approach," in *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, 2006, pp. 1001–1008.

[5] S. S. Benfield, J. Hendrickson, and D. Galanti, "Making a strong business case for multiagent technology," in *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent systems*. ACM, 2006, pp. 10–15.

[6] L. Braubach, A. Pokahr, and W. Lamersdorf, "Negotiation-based patient scheduling in hospitals," in *Advanced Intelligent Computational Technologies and Decision Support Systems*, 2014, pp. 107–121.

[7] S. McArthur, E. Davidson, V. Catterson, A. Dimeas, N. Hatziargyriou, F. Ponci, and T. Funabashi, "Multi-agent systems for power engineering applications – part i: Concepts, approaches, and technical challenges," vol. 22, no. 4. IEEE, 2007, pp. 1743–1752.

[8] M. Bajracharya, M. W. Maimone, and D. Helmick, "Autonomy for mars rovers: Past, present, and future," vol. 41, no. 12. IEEE, 2008.

[9] F. Meneguzzi and L. De Silva, "Planning in BDI agents: a survey of the integration of planning algorithms and agent reasoning," in *The Knowledge Engineering Review*, vol. 30, no. 1, 2015, pp. 1–44.

[10] K. Bauters, K. McAreavey, J. Hong, Y. Chen, W. Liu, L. Godoís, and C. Sierra, "Probabilistic planning in AgentSpeak using the POMDP framework," in *Combinations of Intelligent Methods and Applications*, 2016, pp. 19–37.

[11] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," in *Artificial intelligence*, 1998, pp. 99–134.

[12] O. Despouys and F. F. Ingrand, "Propice-plan: Toward a unified framework for planning and execution," in *5th European Conference on Planning*, 1999, pp. 278–293.

[13] F. Meneguzzi and M. Luck, "Composing high-level plans for declarative agent programming," in *Declarative Agent Languages and Technologies*, 2007, pp. 69–85.

[14] L. De Silva, S. Sardina, and L. Padgham, "First principles planning in BDI systems," in *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, 2009, pp. 1105–1112.

[15] F. Meneguzzi and M. Luck, "Leveraging new plans in agentspeak (PL)," in *International Workshop on Declarative Agent Languages and Technologies*, 2008, pp. 111–127.

[16] K. Bauters, W. Liu, J. Hong, C. Sierra, and L. Godo, "CAN(PLAN)+: Extending the operational semantics of the BDI architecture to deal with uncertain information," in *the 30th Conference on Uncertainty in Artificial Intelligence*, 2014, pp. 52–61.

[17] J. Thangarajah, S. Sardina, and L. Padgham, "Measuring plan coverage and overlap for agent reasoning," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, 2012, pp. 1049–1056.

[18] Y. Yao, L. De Silva, and B. Logan, "Reasoning about the executability of goal-plan trees," in *International Workshop on Engineering Multi-Agent Systems*, 2016, pp. 176–191.

[19] C. E. Alchourrón, P. Gärdenfors, and D. Makinson, "On the logic of theory change: Partial meet contraction and revision functions," in *The journal of symbolic logic*, 1985, pp. 510–530.

[20] A. Darwiche and J. Pearl, "On the logic of iterated belief revision," in *Artificial intelligence*, vol. 89, 1997, pp. 1–29.

[21] S. Konieczny, J. Lang, and P. Marquis, "DA2 merging operators," in *Artificial Intelligence*, vol. 157, no. 1-2, 2004, pp. 49–79.

[22] W. Ouerdane, N. Maudet, and A. Tsoukias, "Arguing over actions that involve multiple criteria: A critical review," in *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, 2007, pp. 308–319.

[23] L. Amgoud and H. Prade, "Explaining qualitative decision under uncertainty by argumentation." in *Proceedings of the National Conference on Artificial Intelligence*, 2006, pp. 21–9.

[24] E. Ferretti, M. L. Errecalde, A. J. García, and G. R. Simari, "A possibilistic defeasible logic programming approach to argumentation-based decision-making," in *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 26, no. 4, 2014, pp. 519–550.

[25] J. Ma, W. Liu, J. Hong, L. Godo, and C. Sierra, "Plan selection for probabilistic BDI agents," in *Proceedings of 26th International Conference of Tools with Artificial Intelligence*. IEEE, 2014, pp. 83–90.

[26] N. D. Rotstein, A. García, and G. R. Simari, "Reasoning from desires to intentions: A dialectical framework," in *Proceedings of Association for the Advancement of Artificial Intelligence*, vol. 22, 2007, pp. 136–141.

[27] C. S. Toranzo, M. L. Errecalde, and E. Ferretti, "A framework for multi-criteria argumentation-based decision making within a BDI agent," in *Journal of Computer Science & Technology*, vol. 14, 2014.

[28] B. J. Clement and E. H. Durfee, "Theory for coordinating concurrent hierarchical planning agents using summary information," in *Proceedings of Association for the Advancement of Artificial Intelligence*, 1999, pp. 495–502.

[29] B. J. Clement, E. H. Durfee, and A. C. Barrett, "Abstract reasoning for planning and coordination," in *Journal of Artificial Intelligence Research*, vol. 28, 2007, pp. 453–515.

[30] J. Thangarajah, L. Padgham, and M. Winikoff, "Detecting and avoiding interference between goals in intelligent agents," in *International Joint Conference on Artificial Intelligence*, 2003.

[31] K. H. Dam and M. Winikoff, "Cost-based BDI plan selection for change propagation," in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, 2008, pp. 217–224.

[32] A. Hunter and S. Konieczny, "Approaches to measuring inconsistent information," in *Inconsistency tolerance*, 2005, pp. 191–236.