Generalised BDI Planning

Felipe Meneguzzi‡ Ramon Fraga Pereira† Nir Oren‡

†The University of Manchester, England, UK
ramon.fragapereira@manchester.ac.uk
‡University of Aberdeen, Scotland, UK
{felipe.meneguzzi,n.oren}@abdn.ac.uk

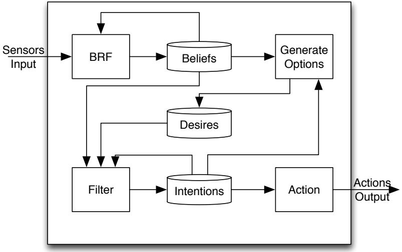
Detroit, May 2025

Table of Contents

BDI Agents and Planning

What?

- Agent architecture based on three “mental” structures:
  - Beliefs, Desires, and Intentions
- Based on a philosophical model for *practical reasoning*
- Provides a blueprint for agent reasoning, suitable for:
  - Agent implementations
  - Reasoning about other agents
- Key process is means ends reasoning:
  - Typically using a *plan library*
  - More recent work focuses on *automated planning*



Planning in BDI

Why?

- Focus of much research in AAMAS for the past three decades, primarily, on:
  - Agent Oriented Software Engineering
  - Agent reasoning cycle
  - Multiagent systems (populated by BDI agents)
- Relatively fewer efforts on the interface of the agent model and **compute-intensive** means-ends reasoning

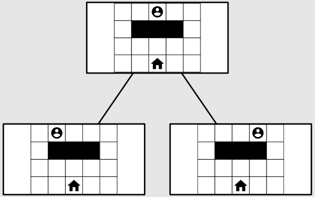
Table of Contents

Definition (Planning Task)

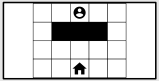
A planning task $\Pi = \langle \Xi, s_0, G \rangle$ is a tuple composed of a domain definition Ξ, an initial state s_0, and a goal state specification G. A solution to a planning task is a plan or policy π that reaches a goal state G starting from the initial state s_0 by following the transitions defined in the domain definition Ξ.

Planning problems have three key ingredients

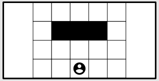
Domain Description



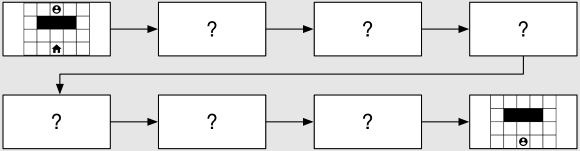
Initial State



Goal State

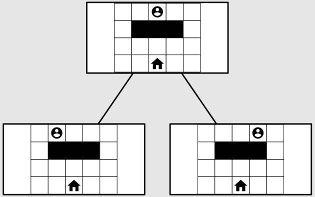


Solution

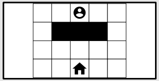


Planning problems have three key ingredients

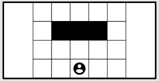
Domain Description



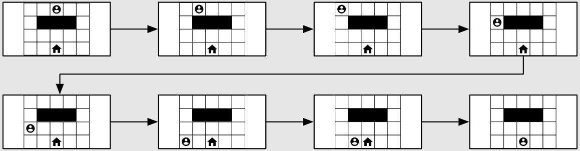
Initial State



Goal State



Solution

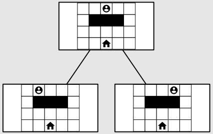


Definition (Generalised Planning Problem)

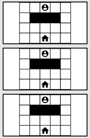
A *generalised planning problem* $\mathcal{GP} = \langle \mathcal{P}_0, \mathcal{P}_1, ..., \mathcal{P}_N \rangle$ is a set of planning problems ($N \geq 2$), where each problem $\mathcal{P}_i = \langle s_0, s_g \rangle$ that share some common structure (typically a planning domain Ξ). A solution to a generalised planning problem is a generalised plan Π, which when followed will solve any problem in $\mathcal{GP}$.

Generalised Planning problems have three key ingredients

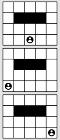
Domain Description



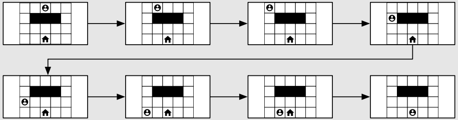
Initial States



Goal States

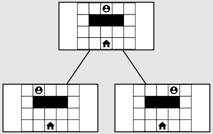


Solution

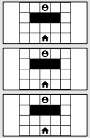


Generalised Planning problems have three key ingredients

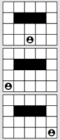
Domain Description



Initial States



Goal States



Solution

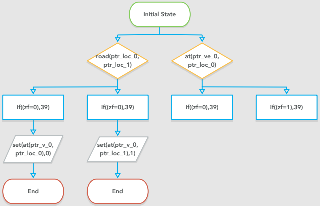
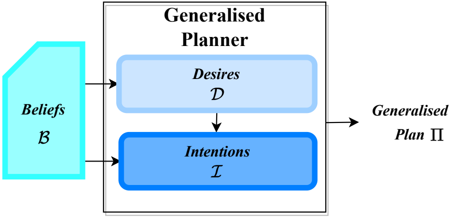


Table of Contents

- We develop a BDI reasoning cycle
  - Only declarative goals (no plan library)
  - Generalised planner as the means-ends reasoning process
- Key processes:
  - Intention selection
  - Desire filtering



[0] Felipe Meneguzzi and Lavindra de Silva. "Planning in BDI agents: a survey of the integration of planning algorithms and agent reasoning". In: *KER* 30.1 (2015), pp. 1–44.

Generalised Planning in BDI

Reasoning Cycle

Require: Filters $\textsc{DesireFilter}$, $\textsc{IntentionFilter}$; Selectors $\textsc{IntentionSelection}$; Interfaces $\textsc{Sense}$, $\textsc{Act}$, $\textsc{BeliefUpdate}$, $\textsc{Next}$

procedure $\textsc{ReasoningCycle}(\mathcal{B}, \mathcal{D}, \mathcal{I}, \Xi)$
    loop
        $\mathcal{B} \leftarrow \textsc{BeliefUpdate}(\mathcal{B}, \textsc{sense}(\ ))$
        if $\mathcal{I}$ is not empty **then** ▷ *Agent has unachieved intentions*
            $\langle\langle\varphi, D\rangle, \Pi_i\rangle \leftarrow \textsc{IntentionSelection}(\mathcal{B}, \mathcal{I})$
            $result \leftarrow \textsc{act}(\textsc{Next}(\mathcal{B}, \Pi_i))$
            if Π_i is empty **and** $\mathcal{B} \models D$ **and** $result \neq \bot$ **then** ▷ *Intention achieved*
                $\mathcal{I} \leftarrow \mathcal{I} - \langle\langle\varphi, D\rangle, \Pi_i\rangle$
            else if $result = \bot$ **and** $\neg\textsc{Retry}(\mathcal{B}, \langle\langle\varphi, D\rangle, \Pi_i\rangle)$ **then** ▷ *Intention Failed*
                $\mathcal{I} \leftarrow \mathcal{I} - \langle\langle\varphi, D\rangle, \Pi_i\rangle$
        else ▷ *Generate new intentions*
            $\mathcal{D}_e \leftarrow \textsc{DesireFilter}(\mathcal{B}, \mathcal{D}, \mathcal{I}, \Xi)$ ▷ *Filter eligible Desires*
            $\mathcal{I} \leftarrow \textsc{IntentionFilter}(\mathcal{B}, \mathcal{D}, \mathcal{I})$

Generalised Planning in BDI

Desire Filter

function $\textsc{DesireFilter}(\mathcal{B}, \mathcal{D}, \mathcal{I}, \Xi)$
    $\mathcal{D}_r \leftarrow \{\langle \varphi, D \rangle \mid \langle \varphi, D \rangle \in \mathcal{D}, \mathcal{B} \models \varphi \land \neg D\}$ ▷ *Relevant Desires*
    $\mathcal{D}_e \leftarrow \{\langle \varphi, D \rangle \mid \langle \varphi, D \rangle \in \mathcal{D}_r, \exists \pi_D \text{ s.t. } \gamma(\Xi, \mathcal{B}, \pi_D) \models D\}$ ▷ *Eligible desires*
    return $\mathcal{D}_e$

Enforces desires an agent can adopt:

- Desires are relevant if they are not yet true, and their pre-condition φ holds in $\mathcal{B}$
- Desires become eligible if there is a plan that will lead to them holding

function $\textsc{IntentionFilter}(\mathcal{B}, \mathcal{D}_e, \mathcal{I})$
    Find $\{\langle \varphi_1, D_1 \rangle \ldots \langle \varphi_n, D_n \rangle\} \in \mathbb{P}^{+}(\mathcal{D}_e)$
    s.t. $\exists \Pi,\ \Pi = \mathcal{G}\textsc{Planner}(\{\langle \Xi, \mathcal{B}, D_1 \rangle \ldots \langle \Xi, \mathcal{B}, D_n \rangle\})$
    $\mathcal{I} \leftarrow \{\langle \langle \varphi_1, D_1 \rangle, \Pi \rangle, \ldots \langle \langle \varphi_n, D_n \rangle, \Pi \rangle\}$
    return $\mathcal{I}$

Enforces **sets of intentions** an agent can adopt:

- In theory: any subset of eligible desire for which there exists a generalised plan
- In practice, further filtering of mutually consistent desires, then generalised planning

Generalised Planning in BDI

Executing Intentions

function $\textsc{IntentionSelection}(\mathcal{B}, \mathcal{D}, \mathcal{I})$

  ▷ *Filter out intentions whose desires are true* ◁

  $\mathcal{I} \leftarrow \{I \mid I = \langle\langle\varphi, D\rangle, \Pi_i\rangle \in \mathcal{I},\ \mathcal{B} \models \varphi \wedge \neg D\}$

  ▷ *Check if a linearisation of Π_i plan is executable* ◁

  $\mathcal{I}' \leftarrow \{I \mid I = \langle\langle\varphi, D\rangle, \Pi_i\rangle \in \mathcal{I},\ \mathcal{B} \models \mathit{pre}(\textsc{Next}(\mathcal{B}, \Pi_i))\}$

  return Pick any intention from $\mathcal{I}'$

Selects a single intention to advance (while dropping irrelevant intentions):

- If the next step is not executable, we defer that intention for a bit
- Otherwise, advance one step

Require: Filters $\textsc{DesireFilter}$, $\textsc{IntentionFilter}$;
Selectors $\textsc{IntentionSelection}$;
Interfaces $\textsc{Sense}$, $\textsc{Act}, \textsc{BeliefUpdate}, \textsc{Next}$

procedure $\textsc{ReasoningCycle}(\mathcal{B}, \mathcal{D}, \mathcal{I}, \Xi)$
  loop
    $\mathcal{B} \leftarrow \textsc{BeliefUpdate}(\mathcal{B}, \textsc{sense}(\,))$
    if $\mathcal{I}$ is not empty **then** ▷ *Agent has unachieved intentions*
      $\langle\langle\varphi, D\rangle, \Pi_i\rangle \leftarrow \textsc{IntentionSelection}(\mathcal{B}, \mathcal{I})$
      $result \leftarrow \textsc{act}(\textsc{Next}(\mathcal{B}, \Pi_i))$
      if Π_i is empty **and** $\mathcal{B} \models D$ **and** $result \neq \bot$ **then**
        $\mathcal{I} \leftarrow \mathcal{I} - \langle\langle\varphi, D\rangle, \Pi_i\rangle$
      else if $result = \bot$ **and** $\neg\textsc{Retry}(\mathcal{B}, \langle\langle\varphi, D\rangle, \Pi_i\rangle)$ **then**
        $\mathcal{I} \leftarrow \mathcal{I} - \langle\langle\varphi, D\rangle, \Pi_i\rangle$
    else ▷ *Generate new intentions*
      $\mathcal{D}_e \leftarrow \textsc{DesireFilter}(\mathcal{B}, \mathcal{D}, \mathcal{I}, \Xi)$
      $\mathcal{I} \leftarrow \textsc{IntentionFilter}(\mathcal{B}, \mathcal{D}, \mathcal{I})$

- Agent advances one intention per cycle
  - Tracks success of intention/desire
  - Failure can lead to retries
  - Both cases can lead to abandonment
- Desire selection when no intention commitments

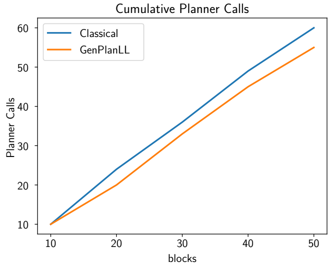
Table of Contents

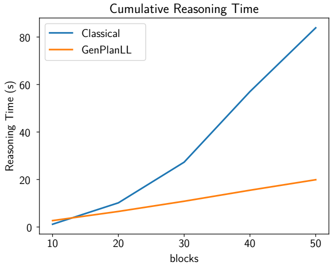
Theoretical Properties of Intentions

1. Intentions normally pose problems for the agent;
   the agent needs to determine a way to achieve them.
2. Intentions provide a "screen of admissibility" for adopting other intentions.
3. Agents "track" the success of their attempts to achieve their intentions.
4. The agent believes p_i is possible.
5. The agent does not believe it will not bring about p_i.
6. Under certain conditions, the agent believes it will bring about p_i.
7. Agents need not intend all the expected side-effects of their intentions.

We implemented the architecture in Python backed by the $\mathrm{BFGP++}$ planner

- Two scenarios: Production Cell and Packaging
- Both showcase advantages of a generalised planning approach to means-ends reasoning:
  - Fewer intention failures
  - Fewer calls to the planner (though each call is expensive)





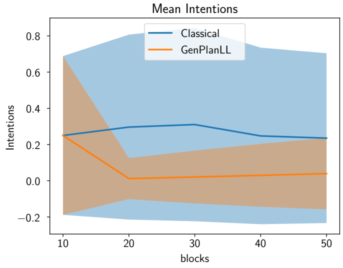


Table of Contents

Discussion, Challenges and Opportunities

Blue Sky, grounded

This paper instantiates our Blue Sky paper from AAMAS 2024:

- First concrete BDI architecture driven by generalised planning
- Automatically enforces BDI-theoretical properties

Moving forward:

- The first of an entirely new family of BDI architectures
  - Multiple potential refinements for the reasoning cycle
    - New strategies for failure recovery
    - Automatic caching of plan sketches from generalised plans
  - No explicit social abilities (Goal Recognition?)
- A research agenda for many years to come

In Memoriam

Ramon Fraga Pereira
15.04.1986 - 04.04.2025