Generalised BDI Planning

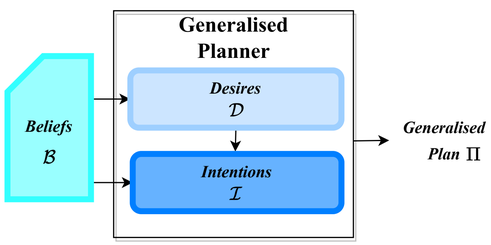
Felipe Meneguzzi, Ramon Fraga Pereira†, and Nir Oren

University of Aberdeen (Scotland, UK) – University of Manchester (England, UK)

✉{felipe.meneguzzi, n.oren}@abdn.ac.uk ✉ramon.fragapereira@manchester.ac.uk

Motivation and Goals

- Means-end reasoning underpins BDI agents. Automated planning is a natural candidate for such reasoning
- Historic disconnect between planning research and agents research
- GEPETTO: BDI architecture driven by Generalised Planning:
  - Agents can reason about committing to multiple sets of desires through generalised planning
  - Allows reasoning about multiple intentions and plan sketch caching



Generalised Planning

- A variation of the planning problem wwith multiple initial states and goals $\mathcal{GP} = \langle \mathcal{P}_0, \mathcal{P}_1, ..., \mathcal{P}_N \rangle$
- Solution to a generalised planning problem is analogous to an *agent program*
- Provides a natural way to implement means-ends reasoning

BDI Gen. Planning Architecture

- Based on the mental attitudes of Beliefs, Desires and Intentions ($\langle \mathcal{B}, \mathcal{D}, \mathcal{I} \rangle$):
  - Declarative desires $\langle \varphi, D, \sigma \rangle$, with precondition φ, formula D, and preference σ
  - Means-ends reasoning driven by a generalised planner

Input: Filters DESIREFILTER, INTENTIONFILTER

Input: Selectors INTENTIONSELECTION,

Input: Interfaces SENSE, ACT, BELIEFUPDATE, NEXT

procedure REASONINGCYCLE($\mathcal{B}, \mathcal{D}, \mathcal{I}, \Xi$)
  loop
    $\mathcal{B} \leftarrow$ BELIEFUPDATE($\mathcal{B}$, SENSE( ))
    if $\mathcal{I}$ is not empty **then**
      $\langle \langle \varphi, D \rangle, \Pi_i \rangle \leftarrow$ INTENTIONSELECTION($\mathcal{B}, \mathcal{I}$)
      $result \leftarrow$ ACT(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** $\neg$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
      $\mathcal{D}_e \leftarrow$ DESIREFILTER($\mathcal{B}, \mathcal{D}, \mathcal{I}, \Xi$)
      $\mathcal{I} \leftarrow$ INTENTIONFILTER($\mathcal{B}, \mathcal{D}, \mathcal{I}$)

BDI Theoretical Properties

1. Intentions pose problems for the agent . . . ;
2. Intentions provide a "screen of admissibility";
3. Agents "track" . . . (progress to) intentions;
4. The agent believes p_i is possible.
5. The agent does not believe it will not bring about p_i.
6. . . . the agent believes it will bring about p_i.
7. Agents need not intend all the expected side effects of their intentions.

Experiments and Evaluation

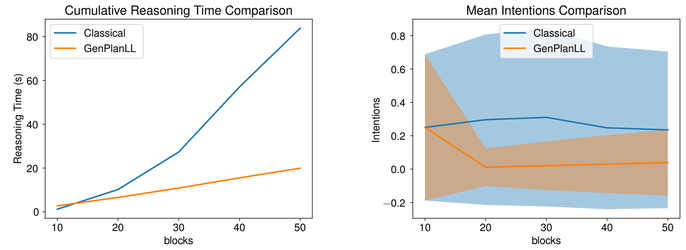


Figure 1: Production Cell results.

- We implemented GEPETTO in Python using BFGP++ as its generalised planner, and SimPlaFy as a classical planner
- Deployed the planner in two scenarios: Production Cell, and Packaging
- Generalised planning combined with plan sketches shows substantial improvements in total reasoning time and intention success under intermittent action failure

Conclusions and Future Work

- GEPETTO is the first fully-fledged practical BDI architecture driven by generalised planning
- Effective but relatively simple reasoning cycle
- Future work:
  - Desire and intention filters
  - Social aspects of declarative agents