Integrating Preferences in Reactive BDI Agents

Mostafa Mohajeri Parizi, Giovanni Sileno and Tom van Engers. UvA, Complex Cyber Infrastructures (CCI) group

Introduction

This work is about internalizing explicit preferences into BDI agents whilst preserving their reactivity. as an example consider a person wants to buy some bread, Is it safe to say that everyone has a set of preferences over bread? But to buybread at the store:

- Do people get out all of their preferences about nutrition, taste, etc. and *reflect* on them to make their choice?
- Or do they already have an *internalized* idea of what type they prefer?

Summary

- Implementations of belief-desire-intention (BDI) model of agency are generally reactivite
- BDI agents typically do not support explicit preferences
- There are works that add preferences to BDI agents as a reflective step by modifying the BDI reasoning cycle
- We use CP-nets to represent the partial preferential ordering between states of the world
- This partial ordering is used to infer and internalize a sequential ordering into agent's procedural knowledge

Reaction vs. Reflection vs. Internalized Reaction

	Compile time	Plan Selection	Outcome
Normal Reactive Agent	N/A	Select first option	Sub-optimal Reaction
Reflective Agent	N/A	Introspect to find the best reaction	Most preferred outcome
Reactive w. Internalized Preferences	Prioritize reactions based on preferences	Select first option	Most preferred outcome

Background.

BDI Agents, Plans and Primitive actions

BDI agents react to events based on their plans,

- A plan `+!g : c => p.`:
  - is a means to achieve goal `g`
  - is achieved by performing the steps in `p`
  - `p` may contain both other goals or primitive actions

```
>> +!buy_bread => #buy_brown_bread.
>> +!buy_bread => #buy_white_bread.
```

- Primitive actions are described with their expected effects

```
>> #buy_brown_bread { => +heath, -taste }
>> #buy_white_bread { => +taste }
```

Preference Language

- Conditional *ceteris paribus* preferences networks (CP-nets)
- CP-nets have the assumption that there is an implicit "all things equal" in human preferences expressions

"I prefer to be more healthy" + all else being equal

"I prefer tasty food if I'm healthy" + all else being equal

```
>> health > ~health.
>> taste > ~taste : health.
```

Method and Example.

Step 1: Translation

Translate the agent script to a Discrete Event Calculus ASP program

Step 2: Plan outcomes

Use a solver (e.g. clingo) to solve the program to get the outcome of each plan

- 1st plan's outcome: health, ~taste
- 2nd plan's outcome: health, taste

Step 3: Prioritization

Reorder the plans based on their outcomes according to CP-net preferences

Result

```
>> +!buy_bread : health, ~taste
       => #buy_white_bread.
       => #buy_brown_bread.
```

Conclusion.

- This work builds upon our previous work on procedural preferences.
- This approach is done fully offline at compile time.
- The resulting script can be run with most off-the-shelf BDI frameworks (JASON,2APL,etc.).
- Explicit verbalized preferences improve re-usability and expressiveness of agent scripts.
- Prioritized procedural knowledge improves both readability, efficiency.
- The primitive action description and preferences are not part of the final script.

Acknowledgments

This work results from work done within Data Logistics for Logistics Data project (DL4LD, www.dl4ld.net). The DL4LD is funded by the Dutch Science Foundation in the Commit2Data program (grant no: 628.001.001).

Contact info:
1: m.mohajeriparizi@uva.nl
2: g.sileno@uva.nl
3: tom.vanengers@tno.com