Comparing Soar and BDI architectures

Sayan Bhattacharyya

Comparison:

Aspects: As plan execution architectures

Testbed: Eaters

Architectures: Belief-Desire-Intention (BDI) architectures

- Explicit knowledge areas (plans), goals and intention stack
- Explicit and manipulable goal priorities

Soar

- Architecturally generated substates (problem spaces) in response to impasses.

Three issues

1. What are the observables and how to generate them.

2. What factors to consider in experiment design.

3. (How) are specific experimental results generalizable to a broader theory?

Observables and their generation

Integrative approach: Empirically investigate the impact of each of task, environment and features of the architecture, while holding the other two constant.

Differential approach: Anticipate and isolate architectural features that produce differentials in behavior and/or performance between the two architectures.

Considerations in experiment design

Two potentially conflicting constraints:

- "natural" way for an architecture to perform a task
- architectures should interface identically with task and environment.

Rim-following task for eaters: Soar ~7 sec
UM-PRS ~23 sec

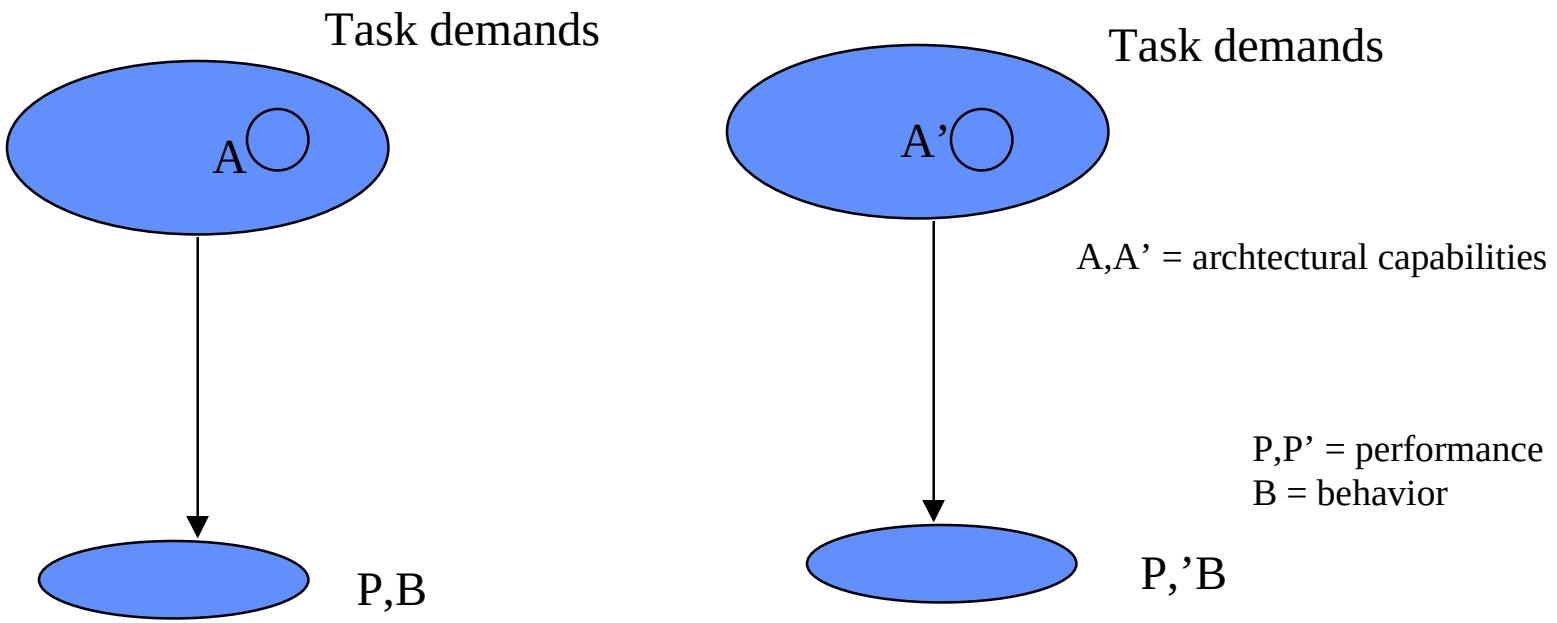
Caution: Normally in UM-PRS, read/writes are initiated as and when needed, unlike Soar.

Considerations in experiment design

Task /environment can interact with capabilities:

	Soar	UM-PRS
With regularities		
distractions ↓	5.12	29.82
	5.40	27.54
	5.41	26.40
	5.64	22.51
Without regularities	5.73	20.22
	6.93	14.92
distractions ↓	7.20	20.93
	8.12	21.98
	8.17	23.41
	8.42	24.81

Generalizability of results



Intuition:

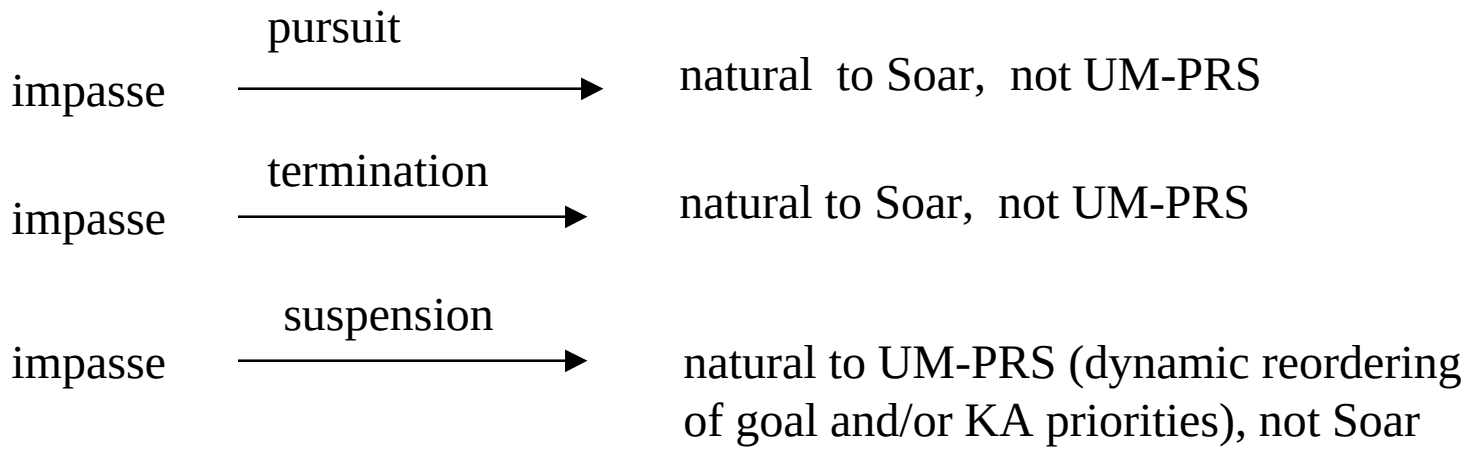
A < A' --> P < P' (Smaller coverage of task demands by architectural capabilities will lead to measurably worse performance).

Experimental study needed to validate this intuition and draw general conclusions about classes of A, T and B.

Generalizability

An example :

Responding to impasses in memory-driven behavior -->
Framework for planning



Mapping constraining focus

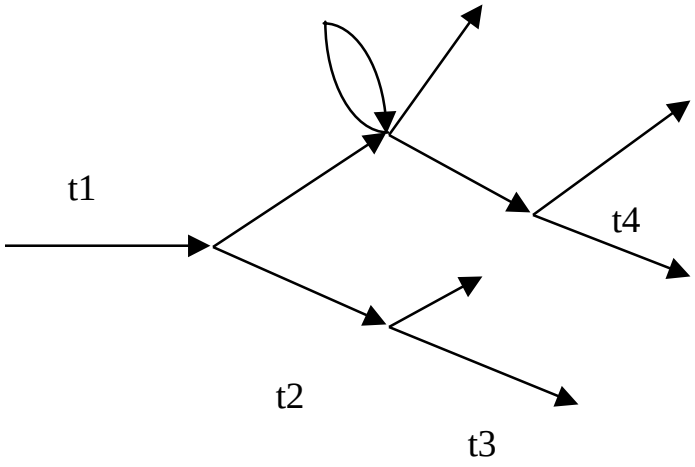
UMPRS: Basic BDI interpreter cycle:

```
do {
    options := option-generator (events, B,G,I)
    selected-options := deliberate (options, B,G,I)
    ...
  }
```

Commitment strategies achieved by constraining focus -- reducing the options generated by option generator and selected by option selector by manipulation of goal and KA priorities.

Soar: Constraining focus achieved by problem spaces.

Mapping impasses



CTL

tie impasses: nodes in CTL with equipriority successors

other impasses: pruning of choices at nodes in the CTL.

Identifying the “extra work” in transforming between these equivalencies.

Nuggets and Coal

Coal: 1) BDI and Soar too different to make comparisons well-structured?

2) Only exploratory forays so far, systematic experiments not yet done.

3) Knowing when not to make architectures "behave unnaturally".

Nuggets: 1) General predictions about classes of tasks and environments good for classes of architectures will be useful.

2) Good to know where Soar stands vis-a-vis other types of architectures.