Interaction among agents that plan

Felipe Meneguzzi
felipe.meneguzzi@kcl.ac.uk
Michael Luck
michael.luck@kcl.ac.uk

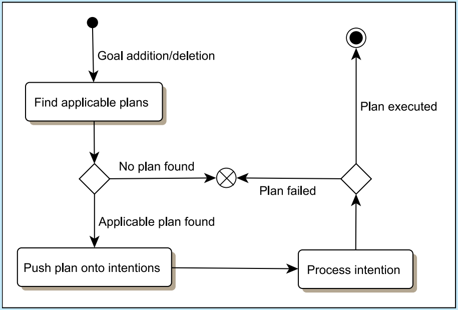
[1]Department of Computer Science
King's College London

From Agent Theory to Agent Implementation
AAMAS 2008 – Estoril, Portugal

1. Background
2. AgentSpeak(PL)
3. Communication and Cooperation
4. Conclusions and Questions

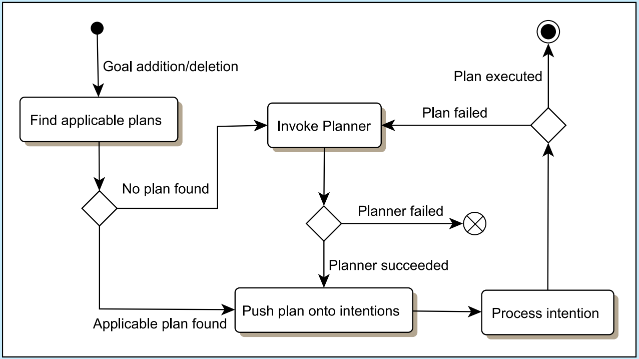
AgentSpeak(L)

- Procedural agent language
- Based on the BDI model
- Designer specifies plans in a library
  - Plans encode procedures
  - Plans are characterised by trigger and context condition
  - Goals are implicit in the plans



Planning in AgentSpeak(PL)

- AgentSpeak(L) + Planning
  - Standard AgentSpeak(L) language
  - Planner invoked through an atomic action
- In principle, any state-space planner can be used

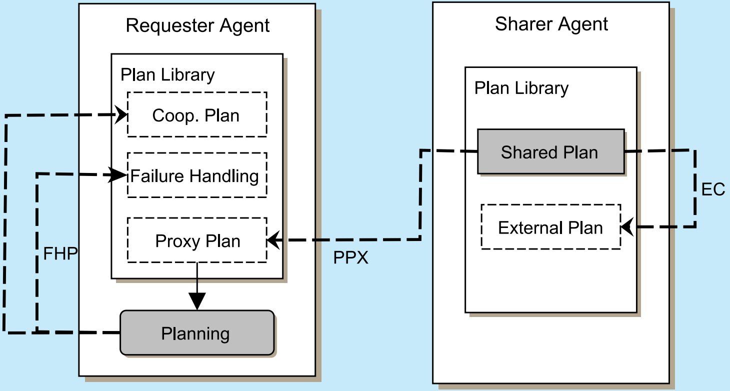


Speech-act based communication

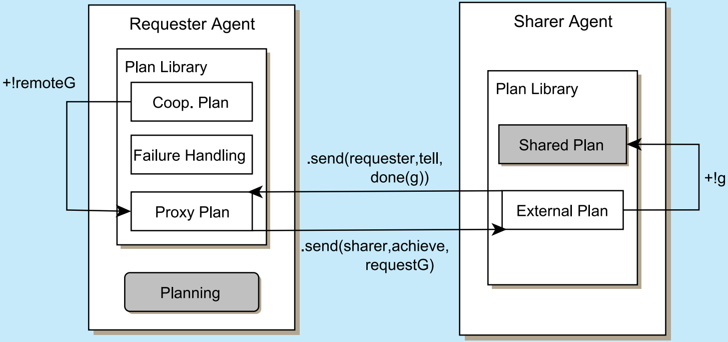
- Popular model of communication in agent languages
- Used in languages like KQML and FIPA
- In our paper, we need the following speech acts:
  - *ask* – to request information from others
  - *tell* – to supply information to others
  - *achieve* – to request another agent to achieve a procedural goal

Plan Patterns

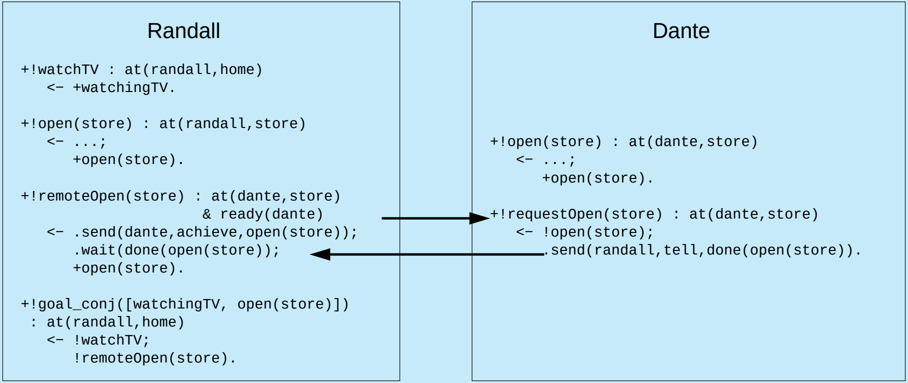
- Patterns here are *plan rewrite rules*
- Take as input one or more plans
- Create new plans based on the original plans



Communication



Simplified Example



Randall

```
+!watchTV : at(randall,home)
   <- +watchingTV.

+!open(store) : at(randall,store)
   <- ...;
      +open(store).

+!remoteOpen(store) : at(dante,store)
                    & ready(dante)
   <- .send(dante,achieve,open(store));
      .wait(done(open(store));
      +open(store).

+!goal_conj([watchingTV, open(store)])
 : at(randall,home)
   <- !watchTV;
      !remoteOpen(store).
```

Dante

```
+!open(store) : at(dante,store)
   <- ...;
      +open(store).

+!requestOpen(store) : at(dante,store)
   <- !open(store);
      .send(randall,tell,done(open(store)).
```

Conclusions

- We show a simple, yet effective cooperation mechanism
- System generates speech-act dialogs (in a way)
- Allows a minimum specification for agents to communicate and distribute control
- Two outstanding issues (foreseen as modules):
  - Distribution
  - Reliability

Session Questions

- What is important about the task/problem addressed?
  - Gap in agent languages regarding support for the **multi** part of MAS
- In what way(s) did the the adopted theory/the implementation framework help/hinder you with solving your problem?
  - Jason has a plan library with KQML communication
  - Lacks proper *reflection*
  - Plan patterns not very flexible

More Session Questions

- In what interesting ways is the realised system (over-/under-)determined? I.e., ”how satisfied” are you with the degree and quality of control achieved?
  - Planning changes AgentSpeak(L) reasoning cycle