KING'S
College
LONDON

University of London

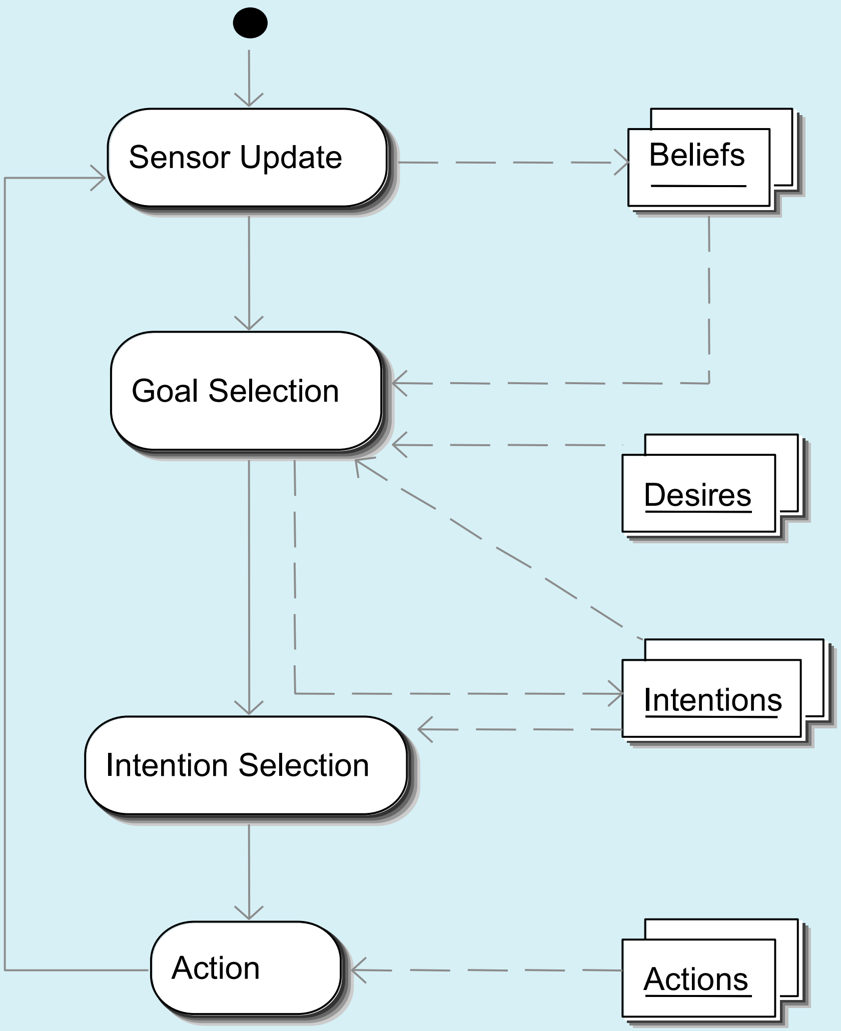
Extending agent languages for autonomy

Felipe Rech Meneguzzi
King's College London

Outline

- BDI Agents
- Agent Languages
- Goal Types
- AgentSpeak(PL)
- Motivated Reasoning
- Cooperation

BDI Agents



- Defined by three mental states:
  - Beliefs – World state
  - Desires – System goals
  - Intentions – Current commitments
- From a theory of practical reasoning

Agent Languages

- AGENT0
  - First agent language
- JADEX
- 3APL
  - Mix agent languages with Java
- AgentSpeak(L)
  - Classic agent language

AgentSpeak(L)

- Based on *Procedural Reasoning System*
- Agent is described in terms of a plan library
- Plans are defined by:
  - A *trigger* condition and a *context*
  - A *body* containing the plan itself
- *Events* drive the adoption of plans

Example AgentSpeak(L) Plan

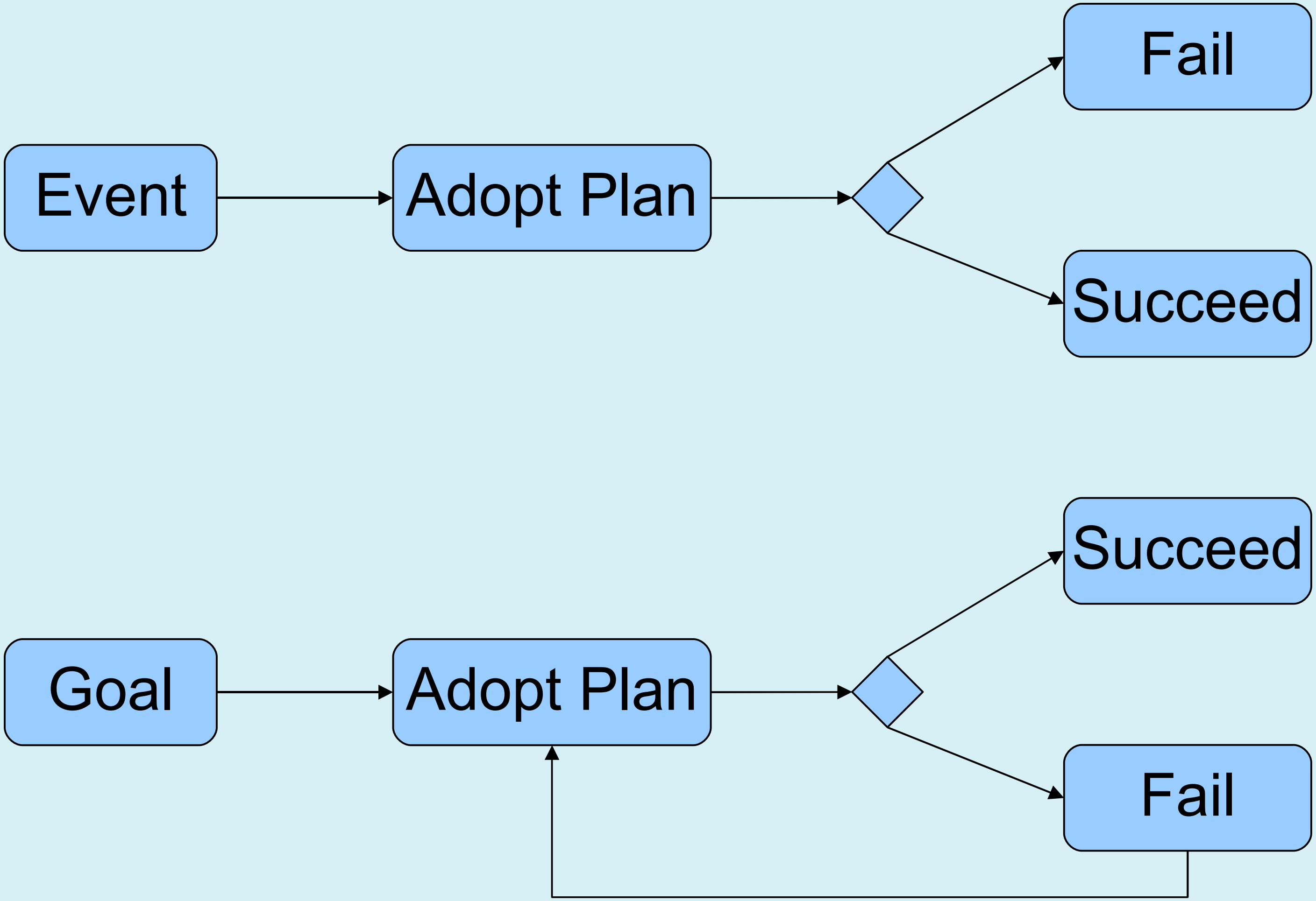
- Event is generated
- First plan with a matching trigger condition and a valid context is adopted
- If the plan fails to finish, the goal has failed

```
+!trigger : context
<- !subgoal1; //calls plan
   action1; //does smth
   !subgoal2;
   action2;
   +belief1; //updates bel.
   -belief2.
```

Goal Types

- Procedural versus Declarative
- Procedural – Efficient yet inflexible
  - Predefined encapsulated behaviours
  - Designer must foresee relevant plans
- Declarative – Expressive, not trivial
  - Desired world states
  - Requires a more complex reasoning mechanism

Procedural vs. Declarative



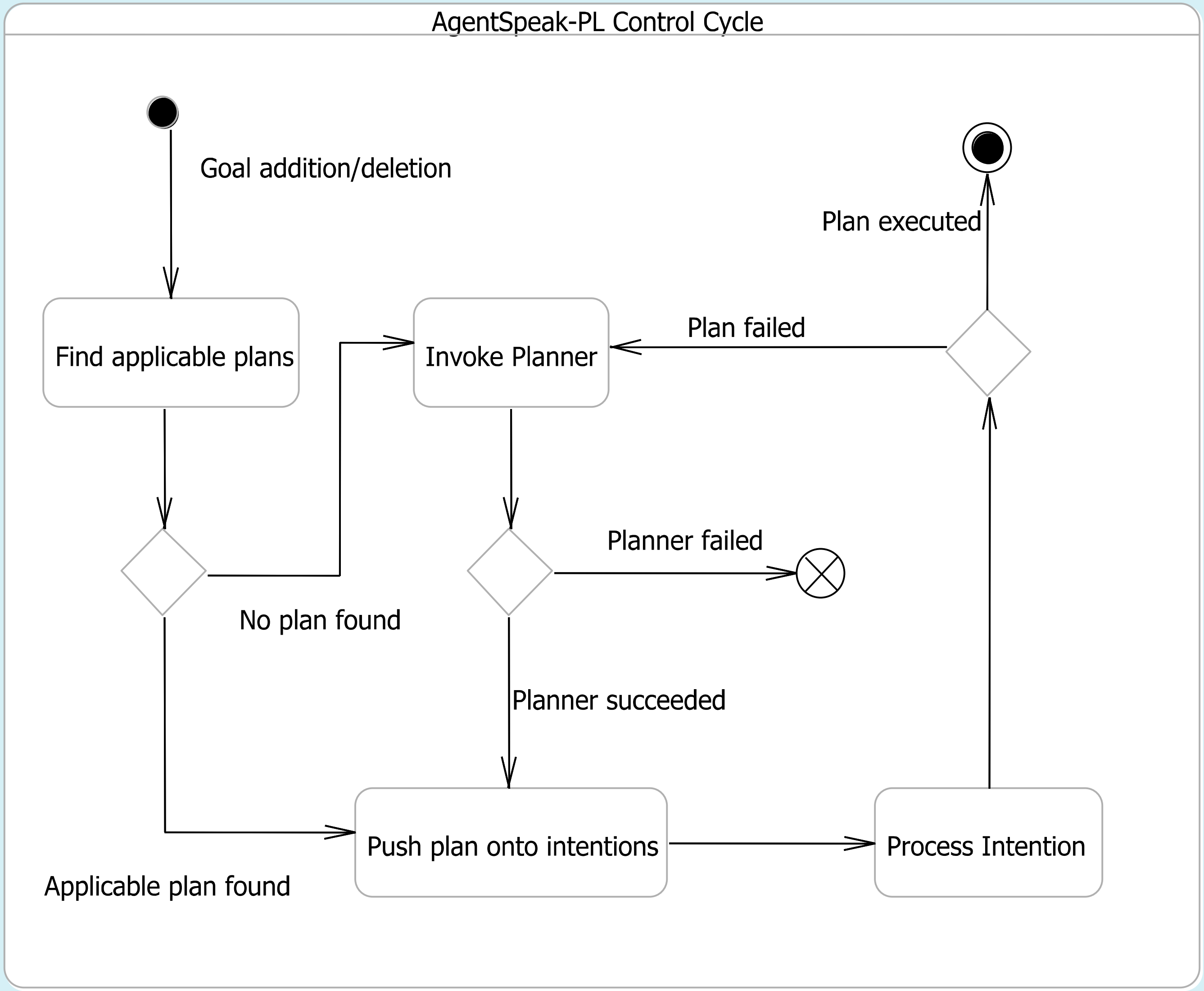
AgentSpeak(PL)

- AgentSpeak(L) interpreter extended with planning capabilities
- State-space planning is always towards declarative goals
- AgentSpeak(PL) supports declarative goals

AgentSpeak(PL) - detail

- Planner encapsulated within an action
- Planning allows the creation of high-level plans
- New plans are added to the plan library improving the agent over time
- Allows for shorter, declarative goal descriptions

AgentSpeak(PL) - Planning



What now?

- We can process declarative goals with AgentSpeak(PL)
- But agents not proactive
- AgentSpeak still based on reactive plans
  - Event → Plan adoption
  - Events in PL may be declarative goals
- We need a goal addition mechanism

Motivations

- Root cause of future-directed behaviour
- Studied by a number of other disciplines
  - Orientation towards particular goals
  - Associated with drives and incentives
  - Controls focus of attention
- In our work: abstraction of meta-reasoning
  - Goal generation
  - Representation of dynamic priorities

AgentSpeak(MPL)

- AgentSpeak(L) + Motivations
  - Standard AgentSpeak(L) language
  - External motivation specification
- Motivation model for
  - Goal generation
  - Plan selection
- Motivation model based on mBDI

Motivation Model

- Tuple that includes:
  - Motivation name
  - Intensity
  - Threshold value
- Motivation functions:
  - Intensity Update
  - Goal Generation
  - Mitigation

$<m, i, t, fi\ ,\ fg\ ,\ fm>$

Intensity Update Function

- Invoked when beliefs are updated
- Controls motivational intensity based on belief base
- Mapping of beliefs to intensity values

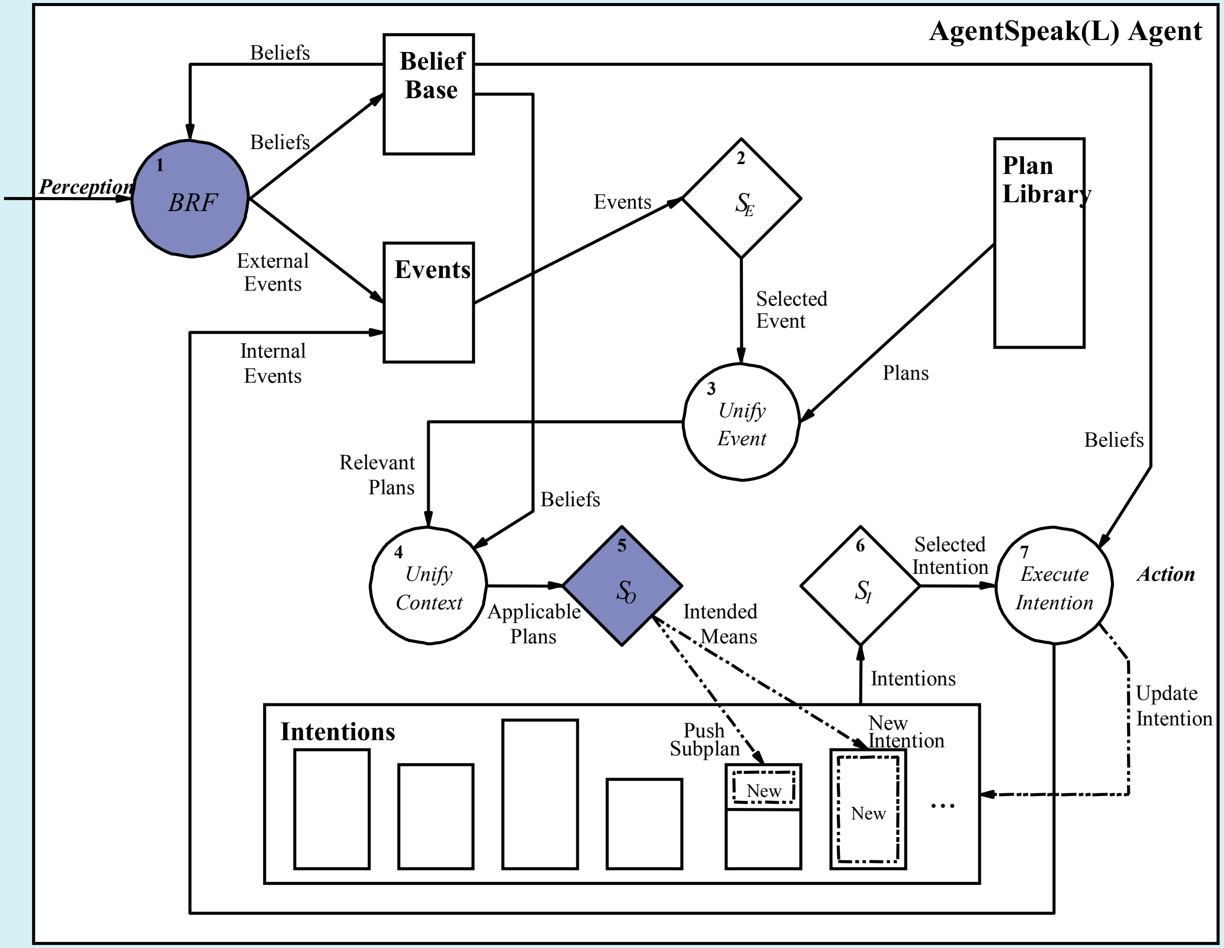
Goal Generation Function

- Invoked when threshold value is exceeded
- Posts new goal events to agent

Mitigation Function

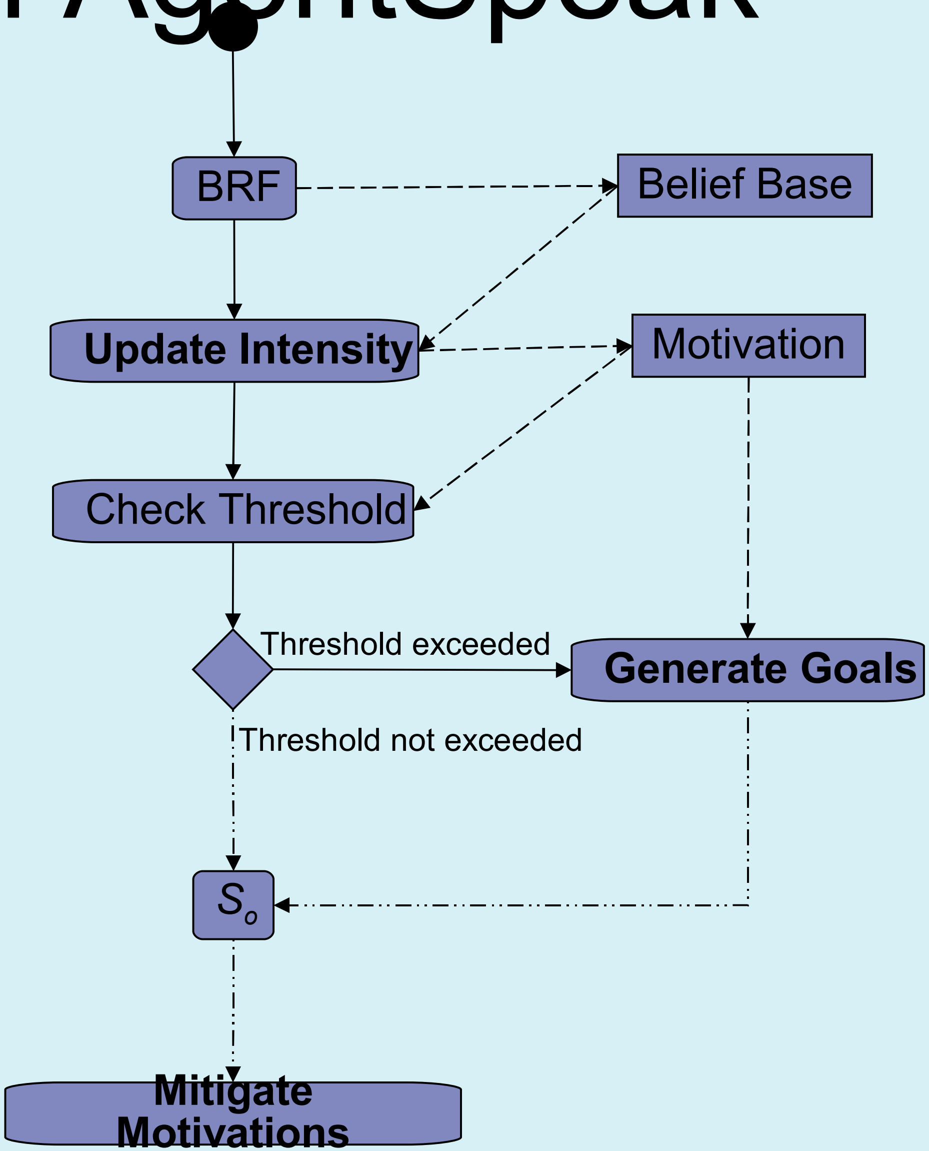
- Invoked after goals are generated
- Updates motivational intensity when a motivation is active
- Similar to Intensity Update Function
    - Also based on belief updates

AgentSpeak Control Cycle



Integration with AgentSpeak

- *Belief Revision Function* associated with motivation functions
- Motivated goals are posted as new achievement goals
- Motivation values are used in the Option Selection Function



Agent Cooperation

- Agent systems are an abstraction for a distributed system
- Multiple agents are expected to cooperate
- Traditional agent languages assume *scripted* cooperation

What is Cooperation?

- Several *modes* of cooperation possible:
  - multiple agents acting towards a common joint goal
  - one agent acting to achieve goals for another agent
  - agents synchronising their actions so as to avoid negative interference
- Here we consider the second alternative

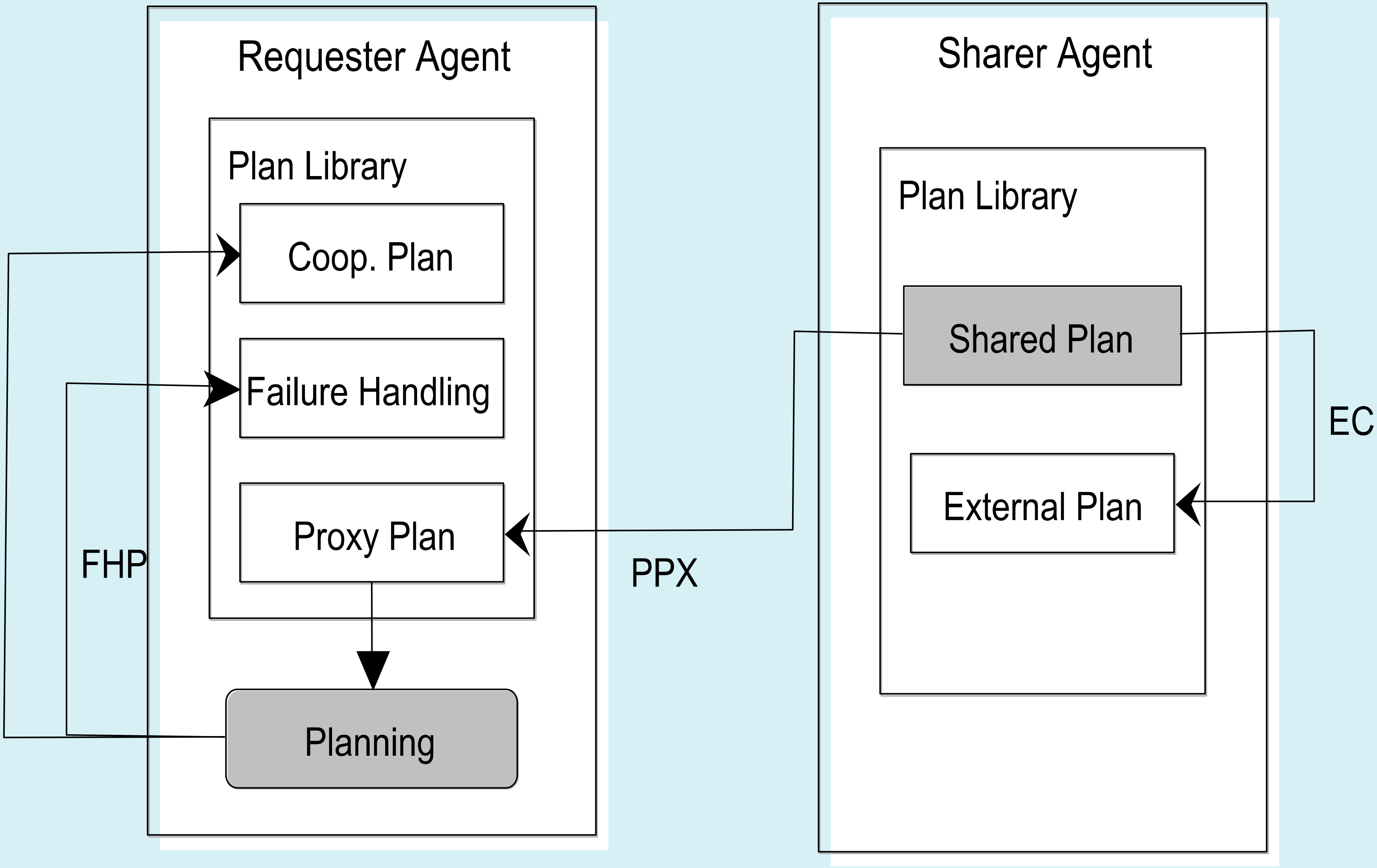
Recap

- AgentSpeak(L):
  - Plan library with all contingencies
  - Plans refer to an individual agent's capabilities
- AgentSpeak(PL):
  - Derives declarative information from AS plans
  - Generates new plans by recombining existing ones
- What can we do with this information?

Social AgentSpeak

- AgentSpeak with the ability to look for capabilities from other agents
- With AgentSpeak(PL) agent is aware of the consequences of its plans
- So he can tell others
- And let them use these capabilities

Social AgentSpeak



Conclusions

- Individually, these are not new techniques
- But nobody bothered to place this in a language
- Serious agent languages:
  - Must have built in features
  - Readily available

Questions?

felipe.meneguzzi@kcl.ac.uk

All software available at:

www.meneguzzi.eu/felipe/software.html