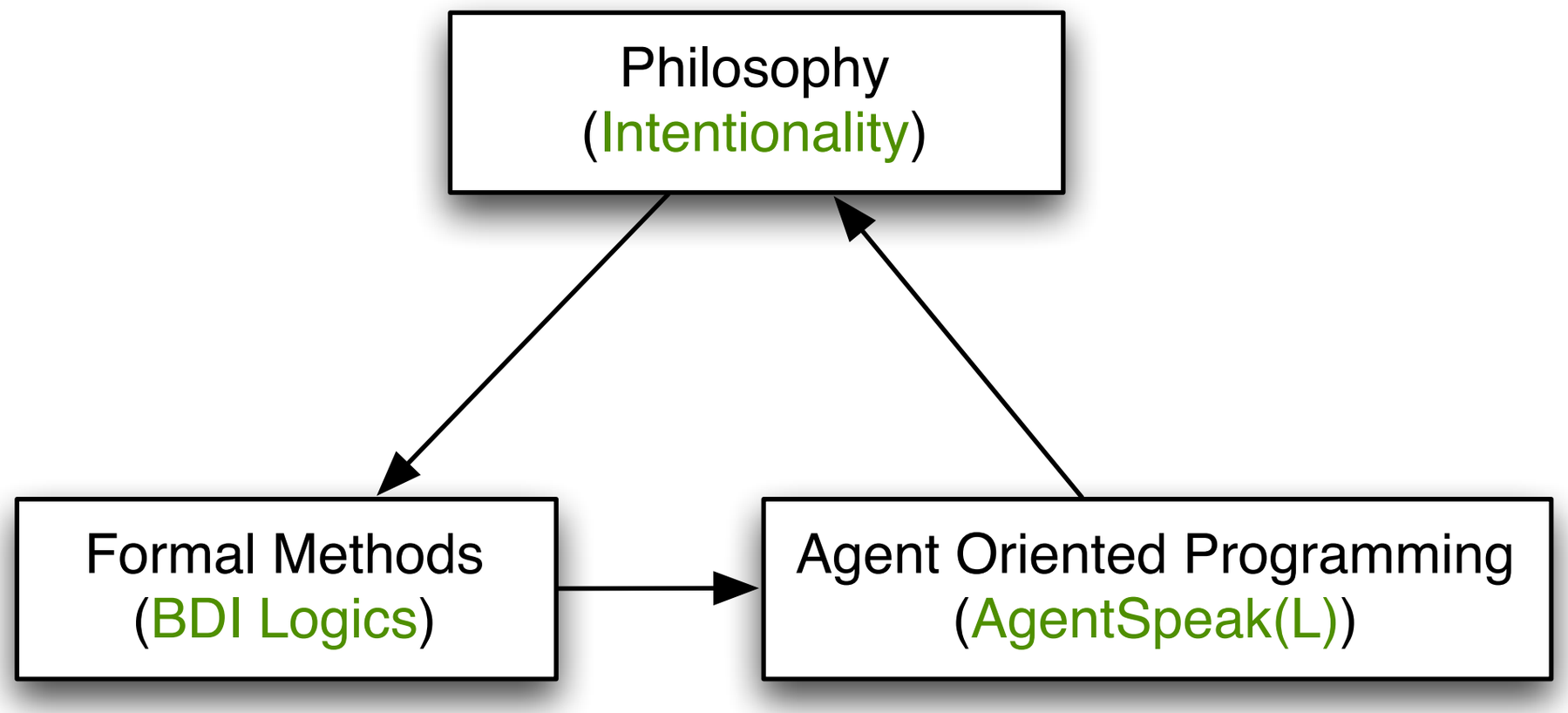
Intentional Learning and Reconsideration in Rational Agents

Dr. Alejandro Guerra Hernández
Universidad Veracruzana (México)
Departamento de Inteligencia Artificial

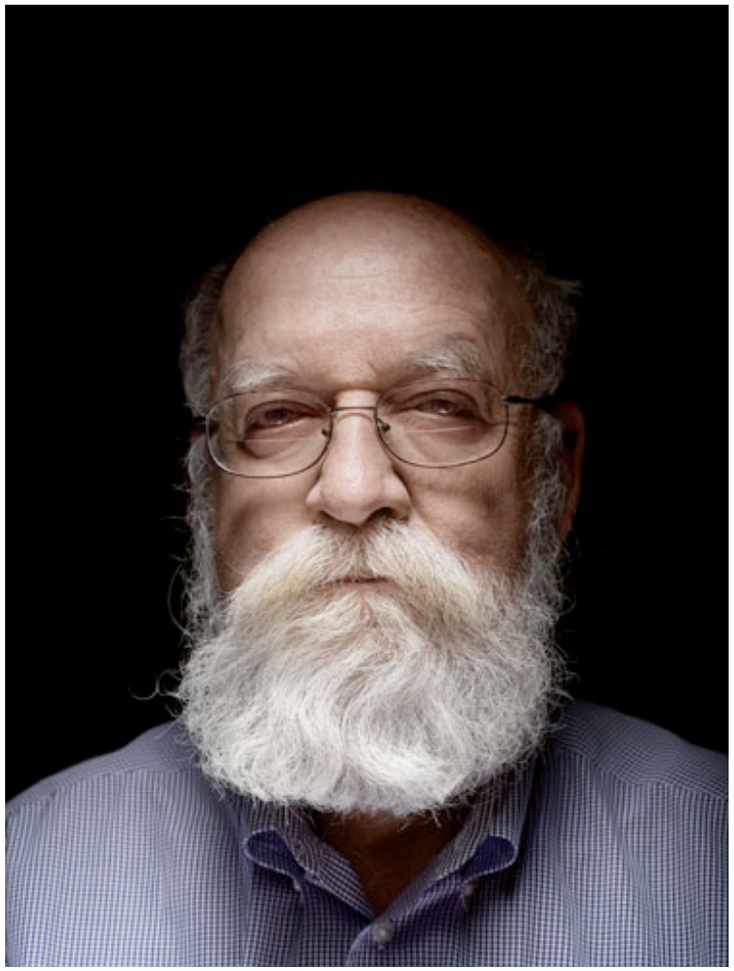
Modus Operandi (for Artificial Intelligence)



Philosophical Background

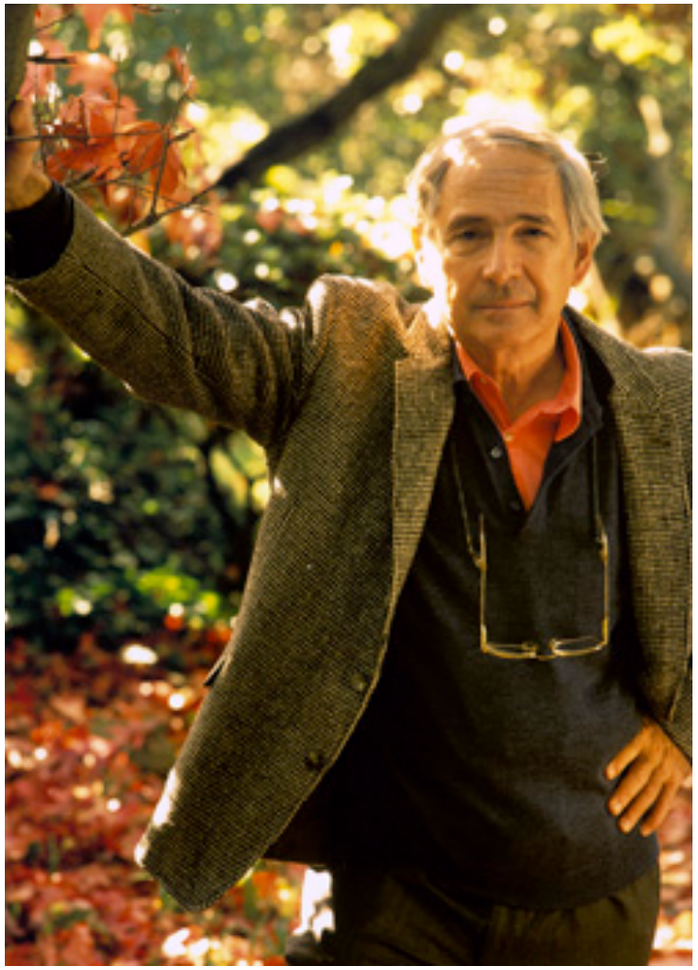
Bratman

Practical Reasoning



Dennett

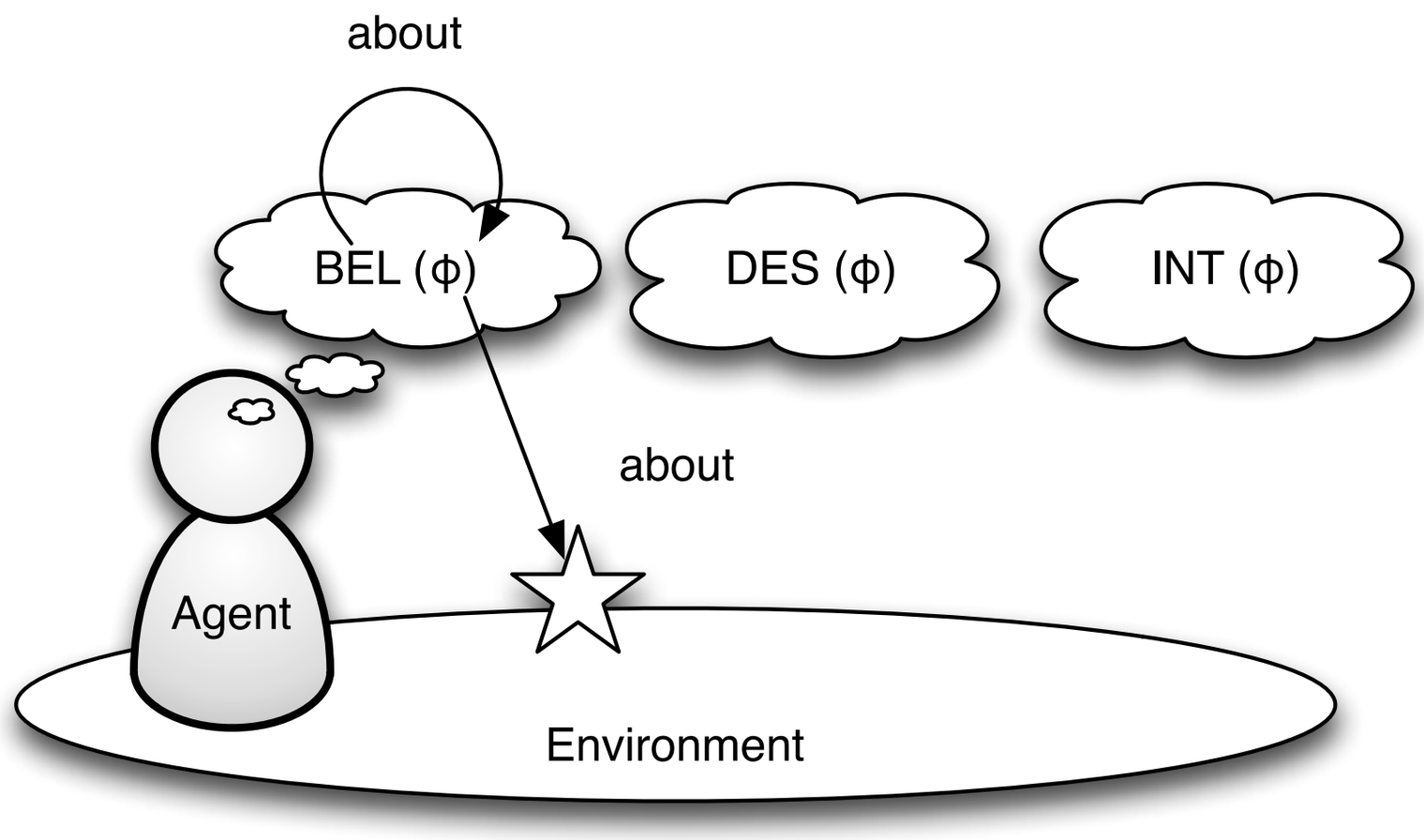
Intentional Stance



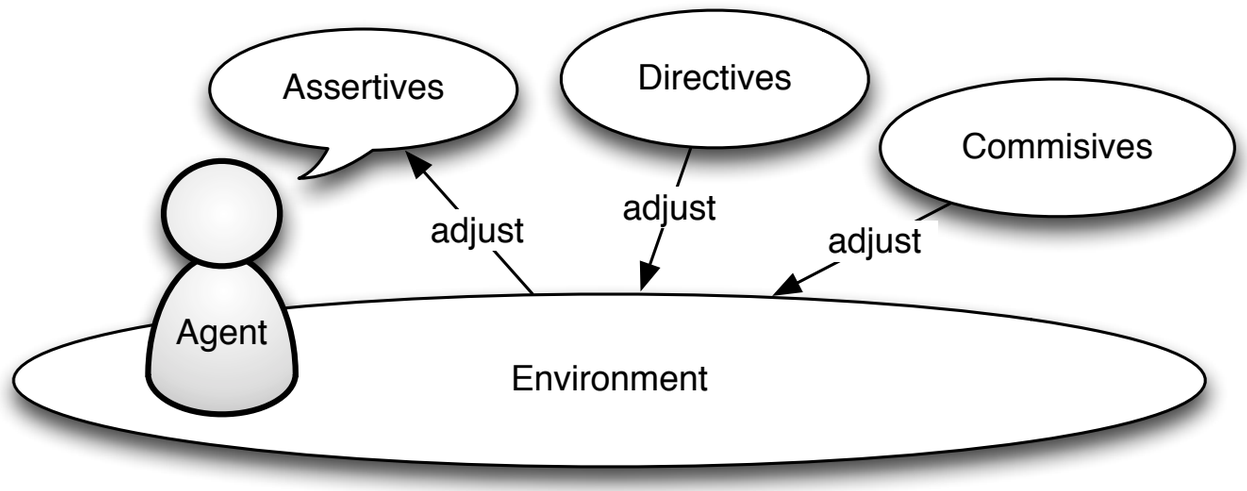
Searle

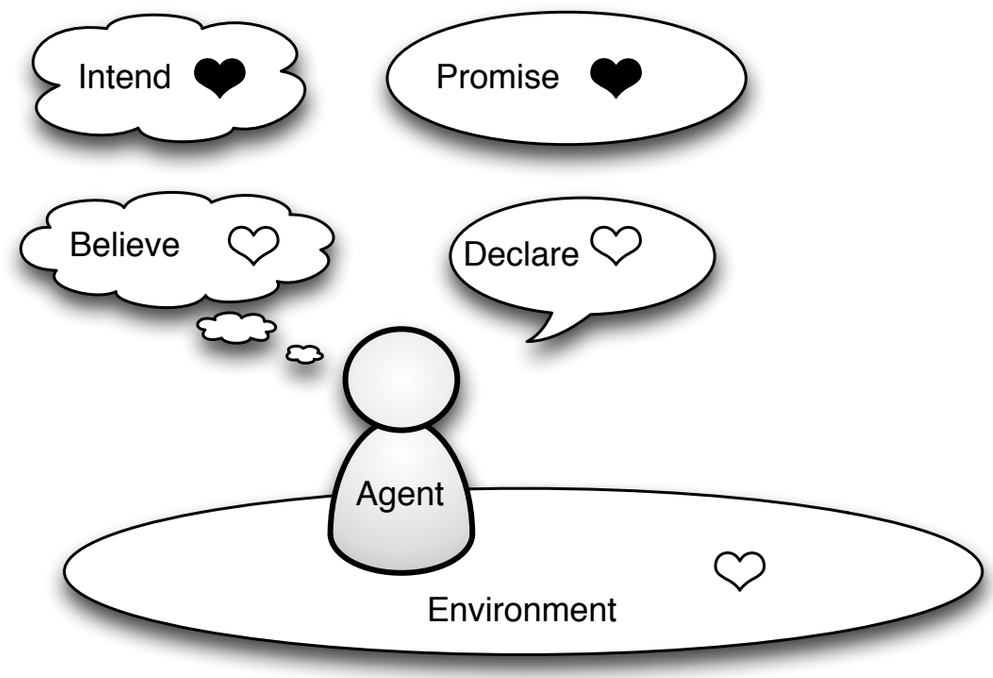
Speech Acts

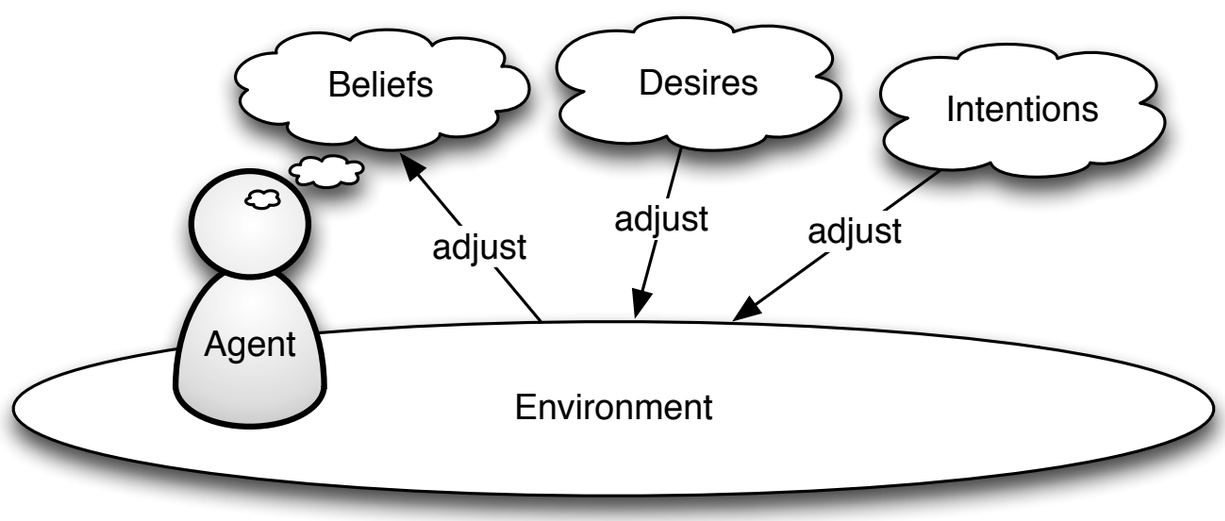
Intentionality 1: Intentional Stance / Representation



Intentionality 2: Speech Acts / Communication







Intentionality 3: Practical Reasoning

- How to do the right thing? vs Epistemic Reasoning.
- The input to such process are Beliefs, Desires and Intentions.
- Intentions are piles of partial, hierarchical, consistent and coherent plans.
- We will be able to deal with temporal aspects of practical reasoning as: Future Intentions, Persistence, Reconsideration and Commitment.
- Policy based Reconsideration is composed by rules determining when to reconsider, if you are a single-minded agent.

Commitment ($B^{KD45}D^{KD}I^{KD}{}_{CTL}$) [Rao, 1998]

- Blind Commitment

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ (\mathsf{BEL}(\phi))$$

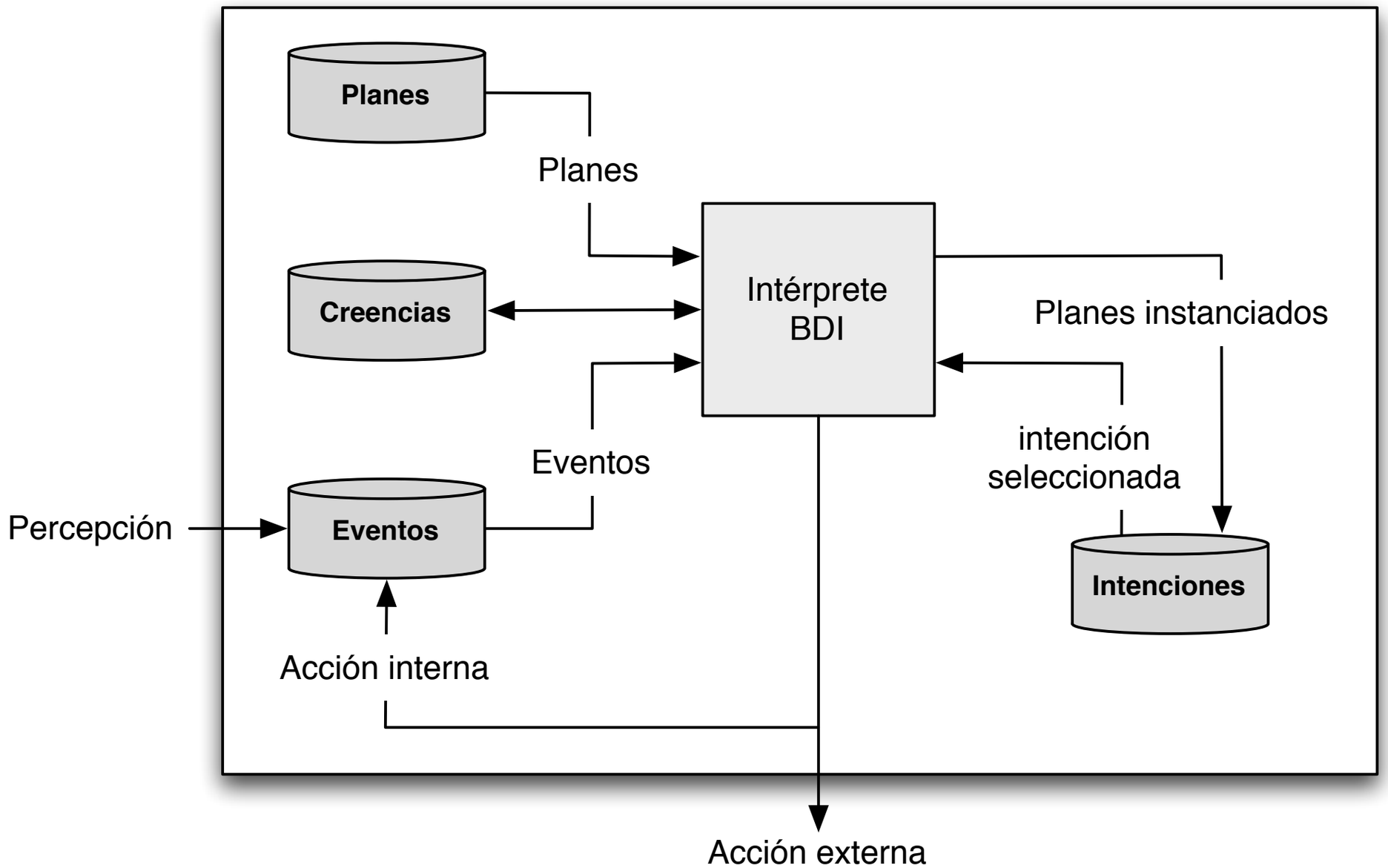
- Rational Commitment (Single-Minded)

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ (\mathsf{BEL}(\phi) \vee \neg\mathsf{BEL}(\mathsf{E}\Diamond\phi))$$

- Emotional Commitment (Open-Minded)

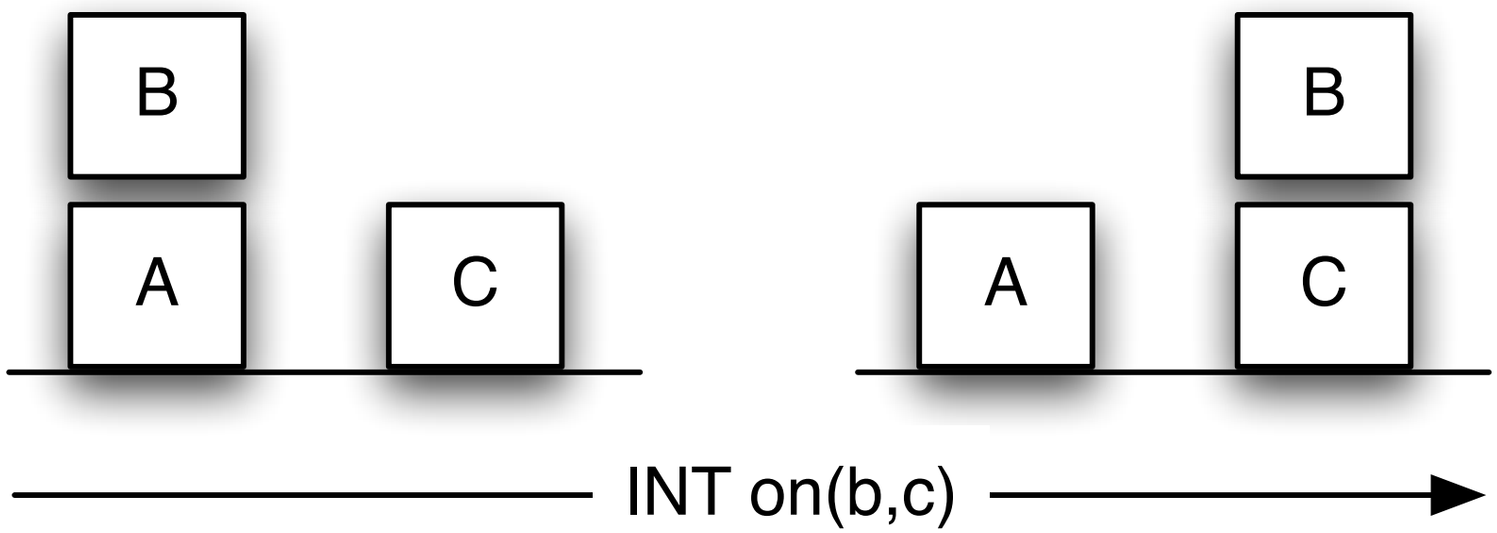
$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ (\mathsf{BEL}(\phi) \vee \neg\mathsf{DES}(\mathsf{E}\Diamond\phi))$$

Agent Architectures -> Agent Oriented Programming



dMARS

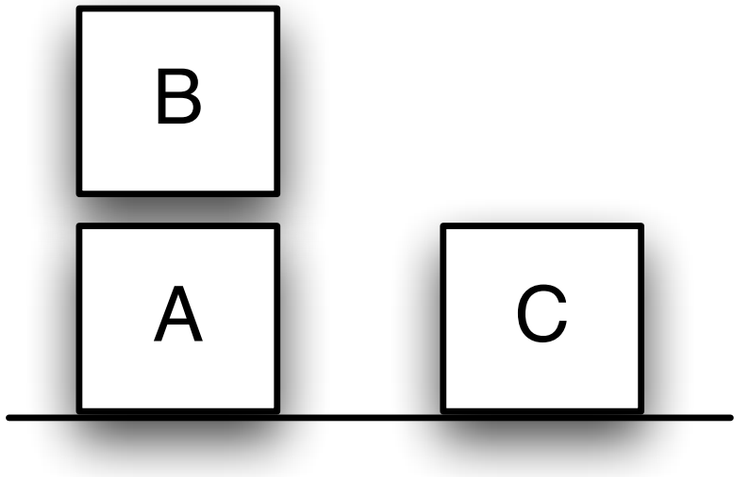
A simple scenario



Blocks World

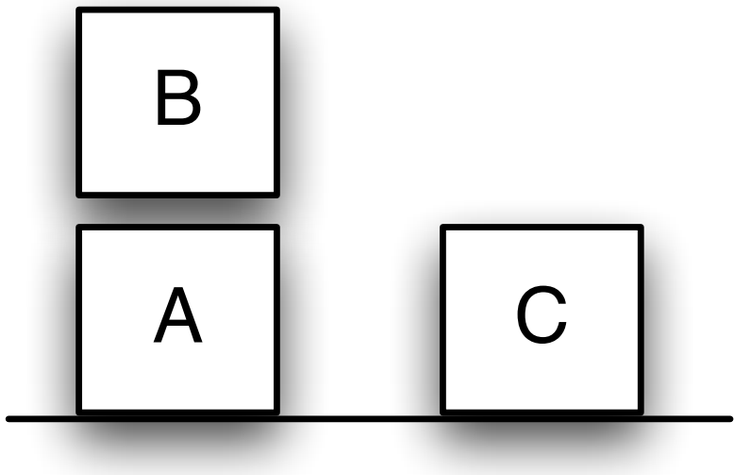
Beliefs

```
6 on(b,a).
7 on(a,table).
8 on(c,table).
```



Beliefs and Rules

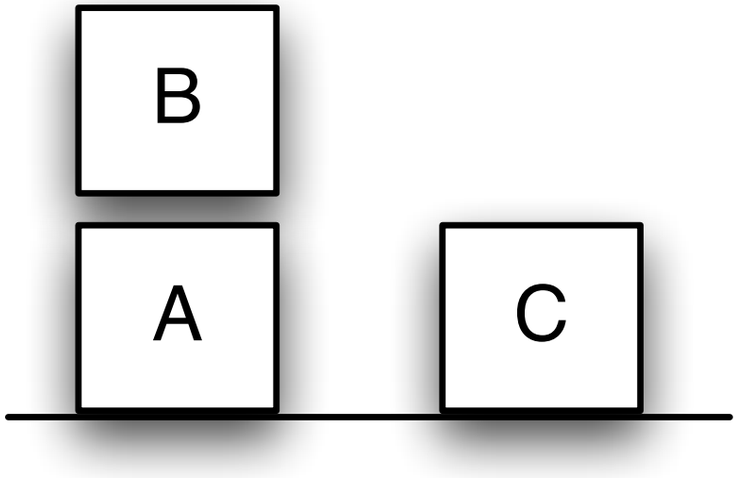
```
6 on(b,a).
7 on(a,table).
8 on(c,table).
```




```
4 clear(table).
5 clear(X) :- not(on(_,X)).
```

Beliefs, Rules, and Test Goals (?)

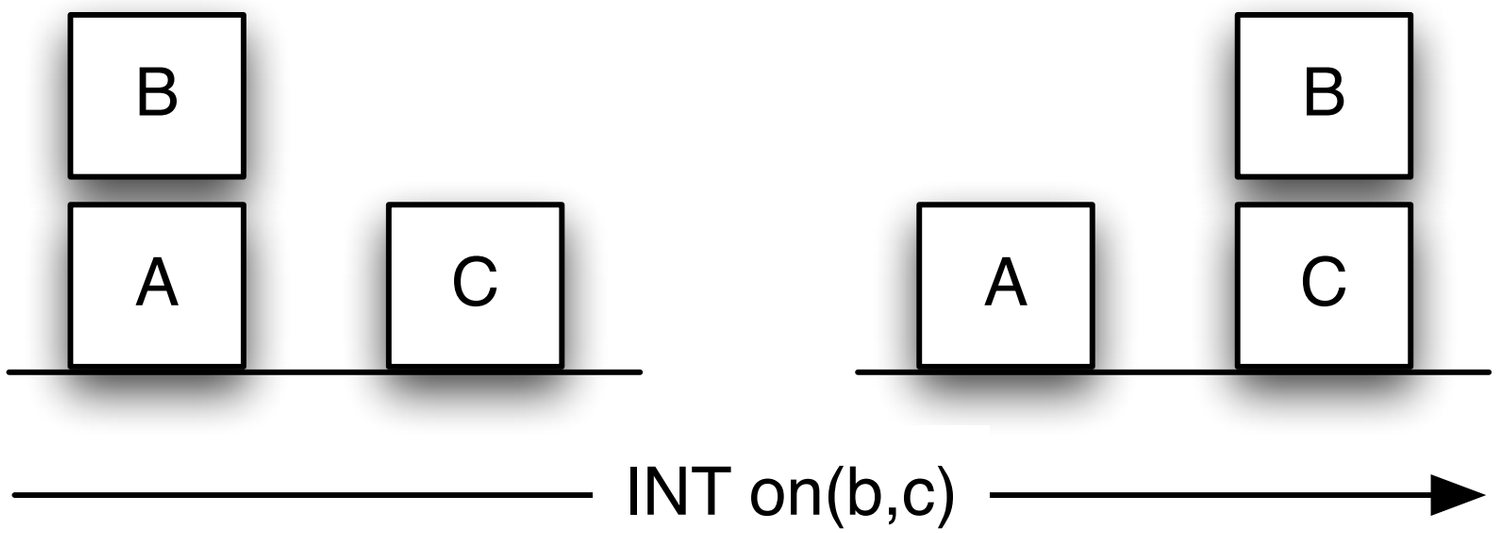
```
6 on(b,a).
7 on(a,table).
8 on(c,table).
```




```
4 clear(table).
5 clear(X) :- not(on(_,X)).
```

?clear(b) => true ?on(a,X) => {X\b}

Achieve Goals as Desires (!)



`!on(b,c)`

Plans (Trigger Event : Context <- Body)

Trigger Event

Add or Delete a Goal or Belief

```
26 +!on(X,Y) : on(X,Y). // already achieved
27 +!on(X,Y) <- !clear(X); !clear(Y); move(X,Y).
```

Plans

Context

Conjunction of First Order Literals

```
26 +!on(X,Y) : on(X,Y). // already achieved
27 +!on(X,Y) <- !clear(X); !clear(Y); move(X,Y).
```

Plans

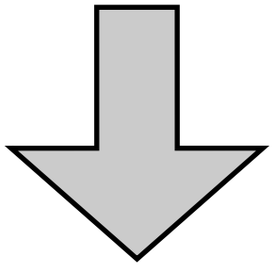
```
26 +!on(X,Y) : on(X,Y). // already achieved
27 +!on(X,Y) <- !clear(X); !clear(Y); move(X,Y).
```

Sequence of Goals, Actions and/or Beliefs updates.

Body

Intentions

```
+!on(a,c) <- !clear(a); !clear(c); move(a,c).
```



```
+!clear(a) : ?on(b,a) <- ?clear(W); move(b,W).
+!on(a,c) <- !clear(c); move(a,c).
```

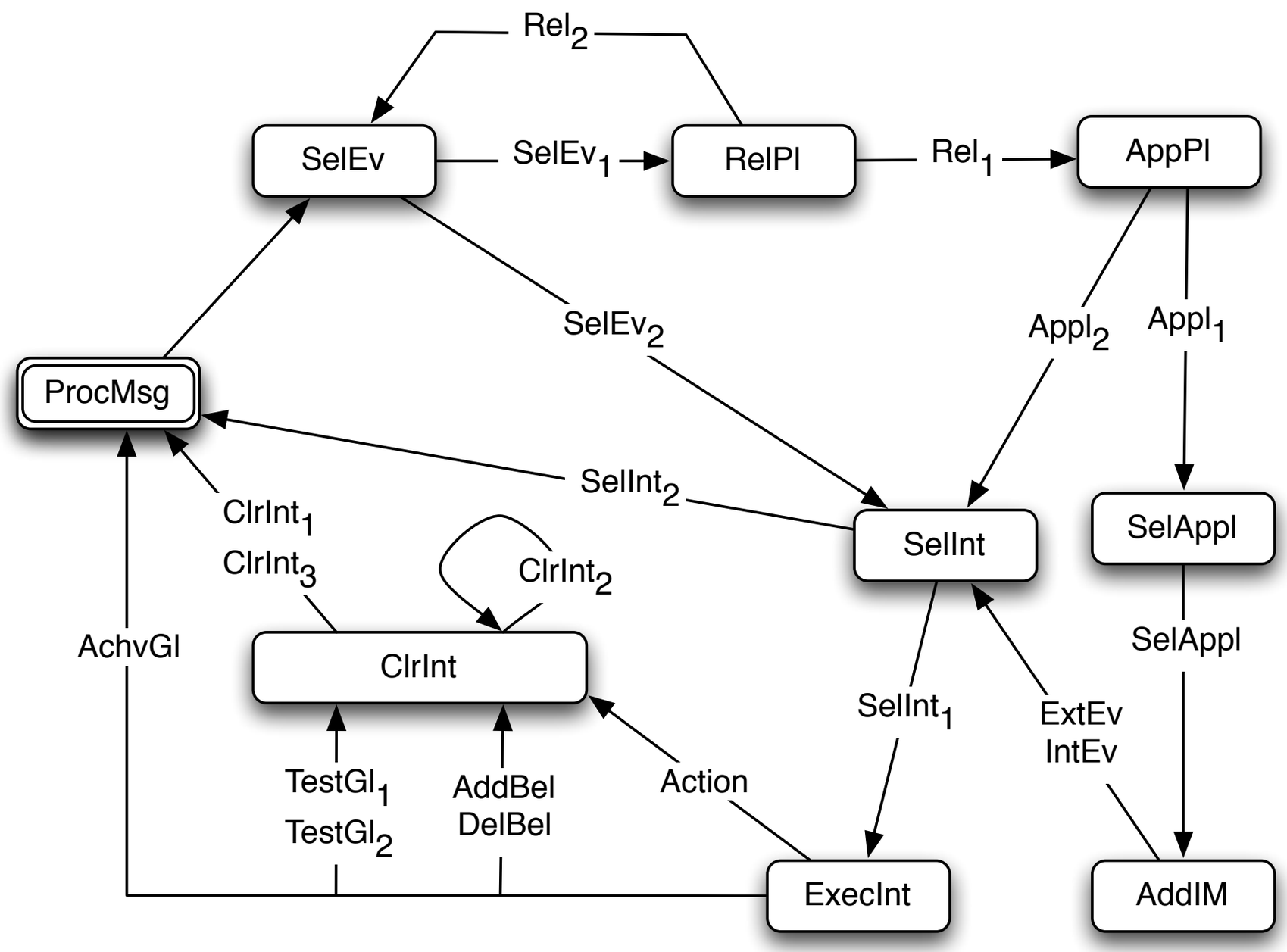
AgentSpeak(L): Configurations [Bordini, 2007]

$$c = \langle ag, C, M, T, s \rangle$$

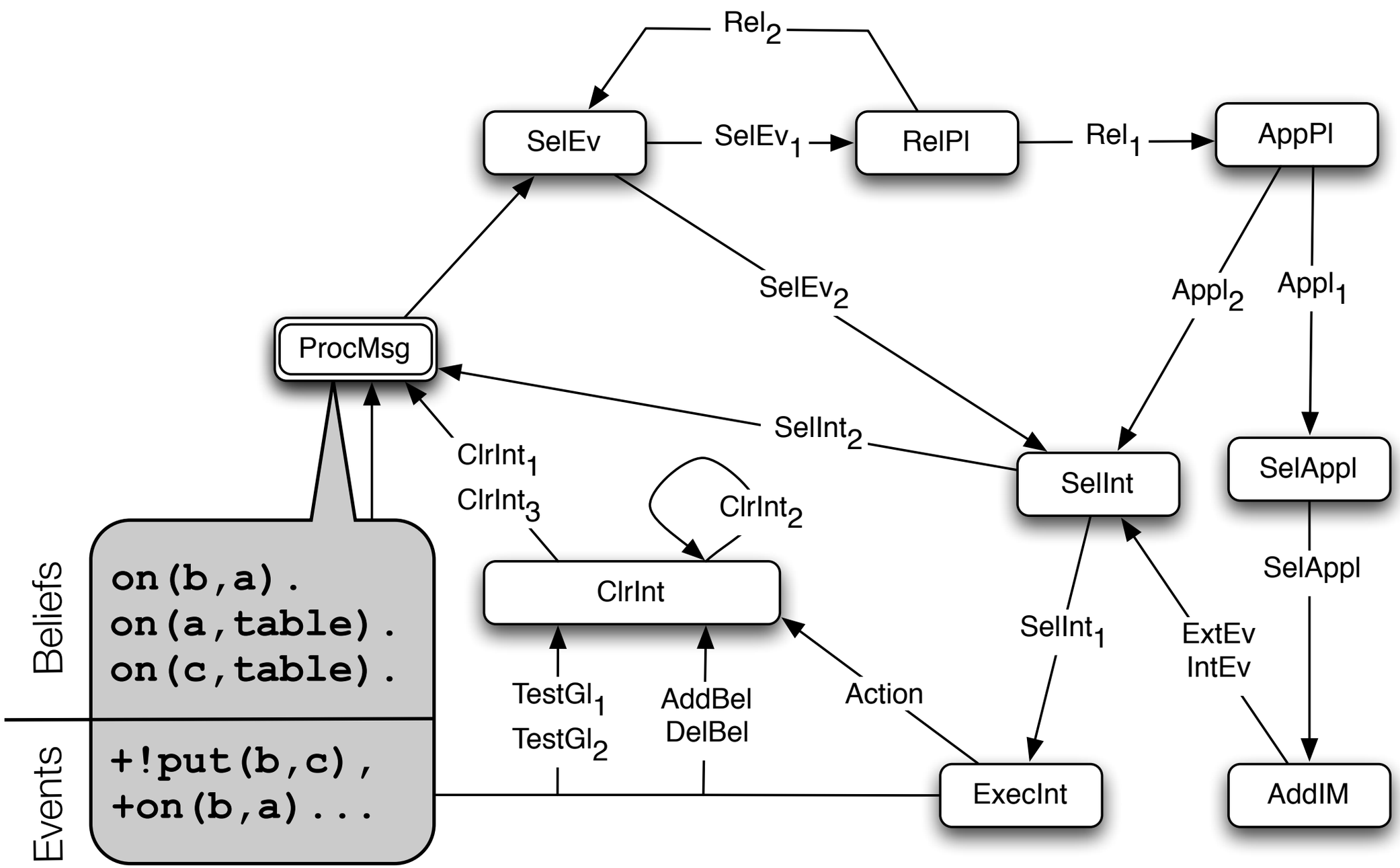
Where:

$$\begin{aligned} ag &= \langle bs, ps \rangle \\ C &= \langle I, E, A \rangle \\ M &= \langle In, Out, SI \rangle \\ s &\in Labels \end{aligned}$$

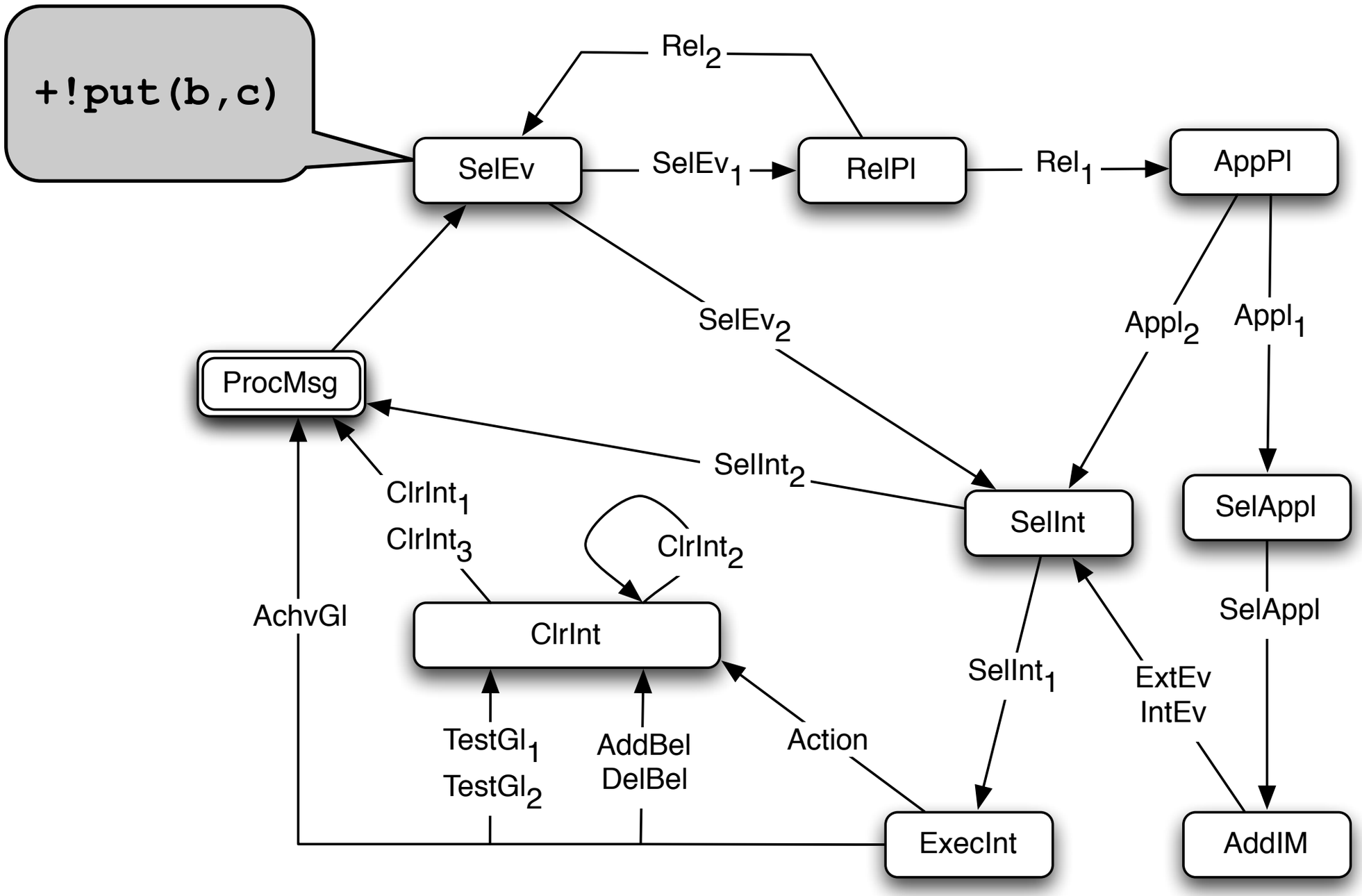
AgentSpeak(L): Semantics



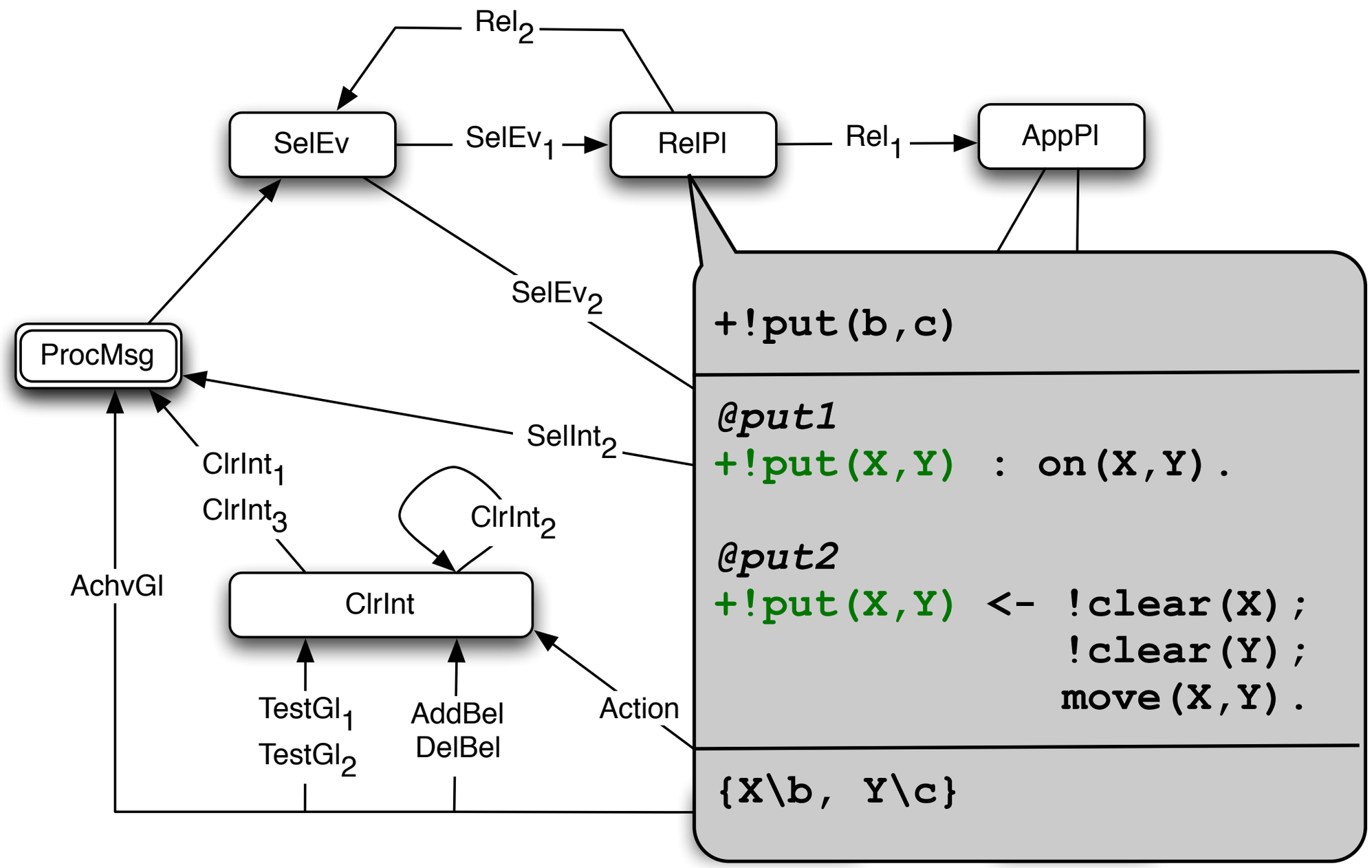
AgentSpeak(L): Process Messages



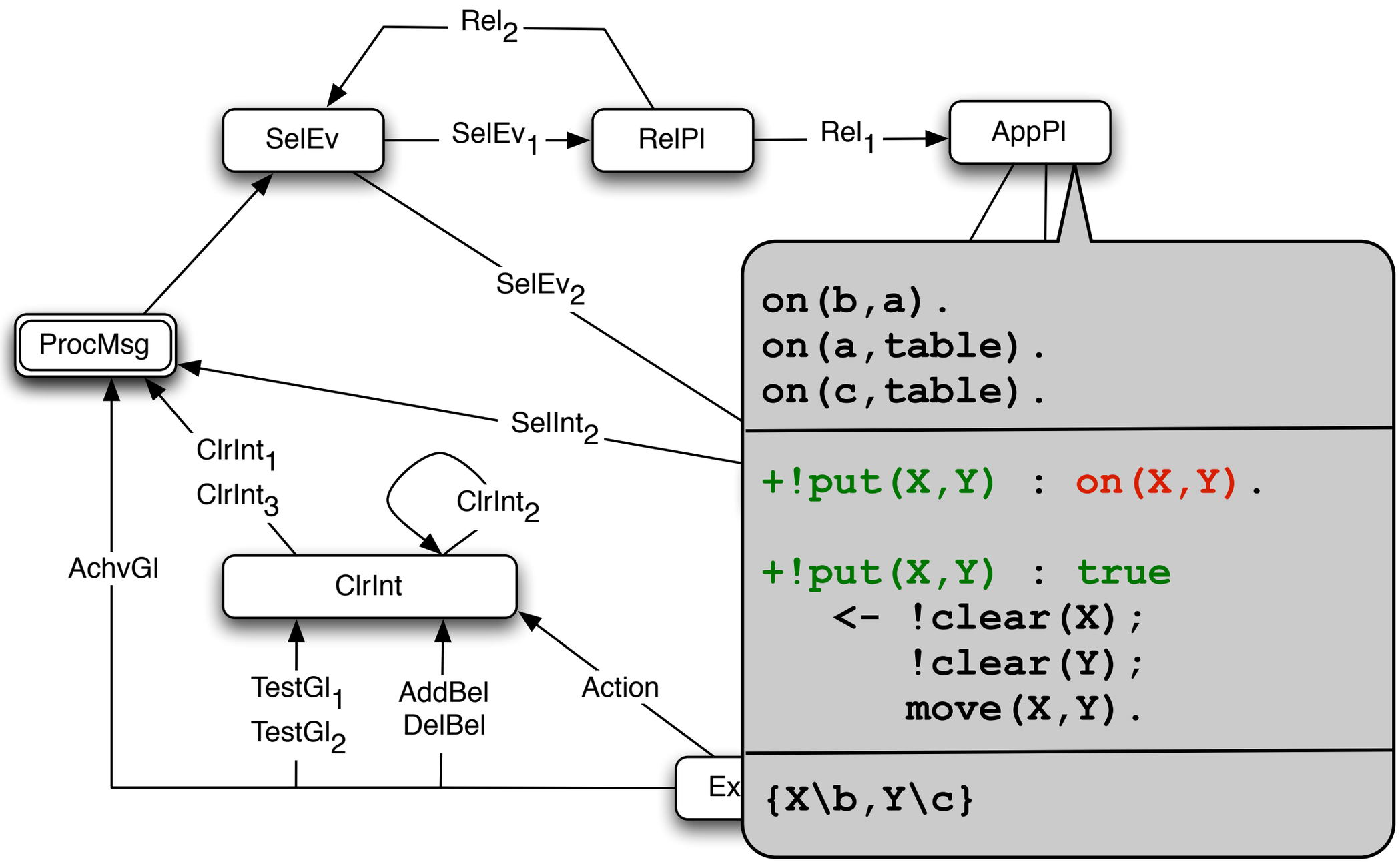
AgentSpeak(L): Select an Event



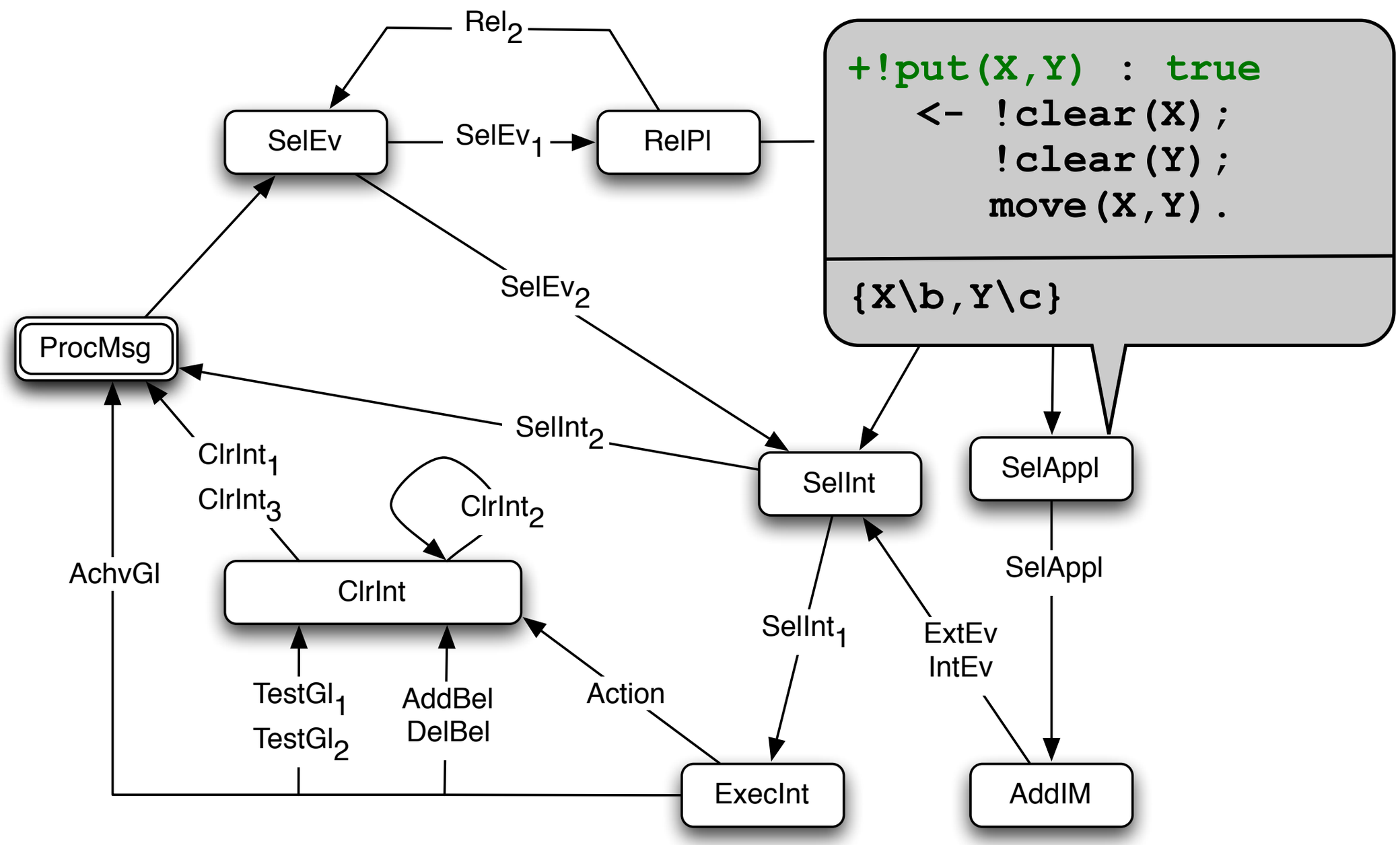
AgentSpeak(L): Select Relevant Plans



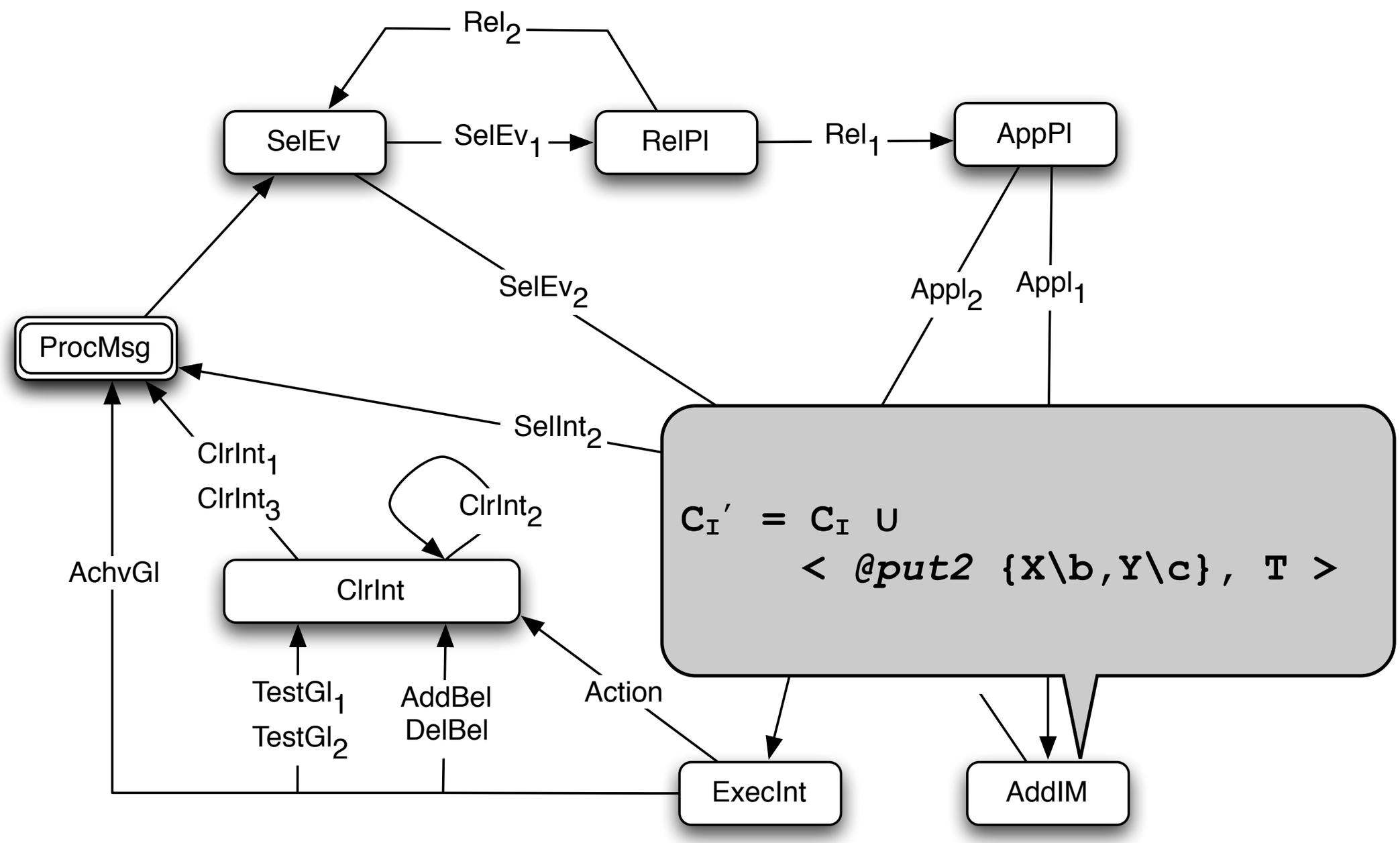
AgentSpeak(L): Select Applicable Plans



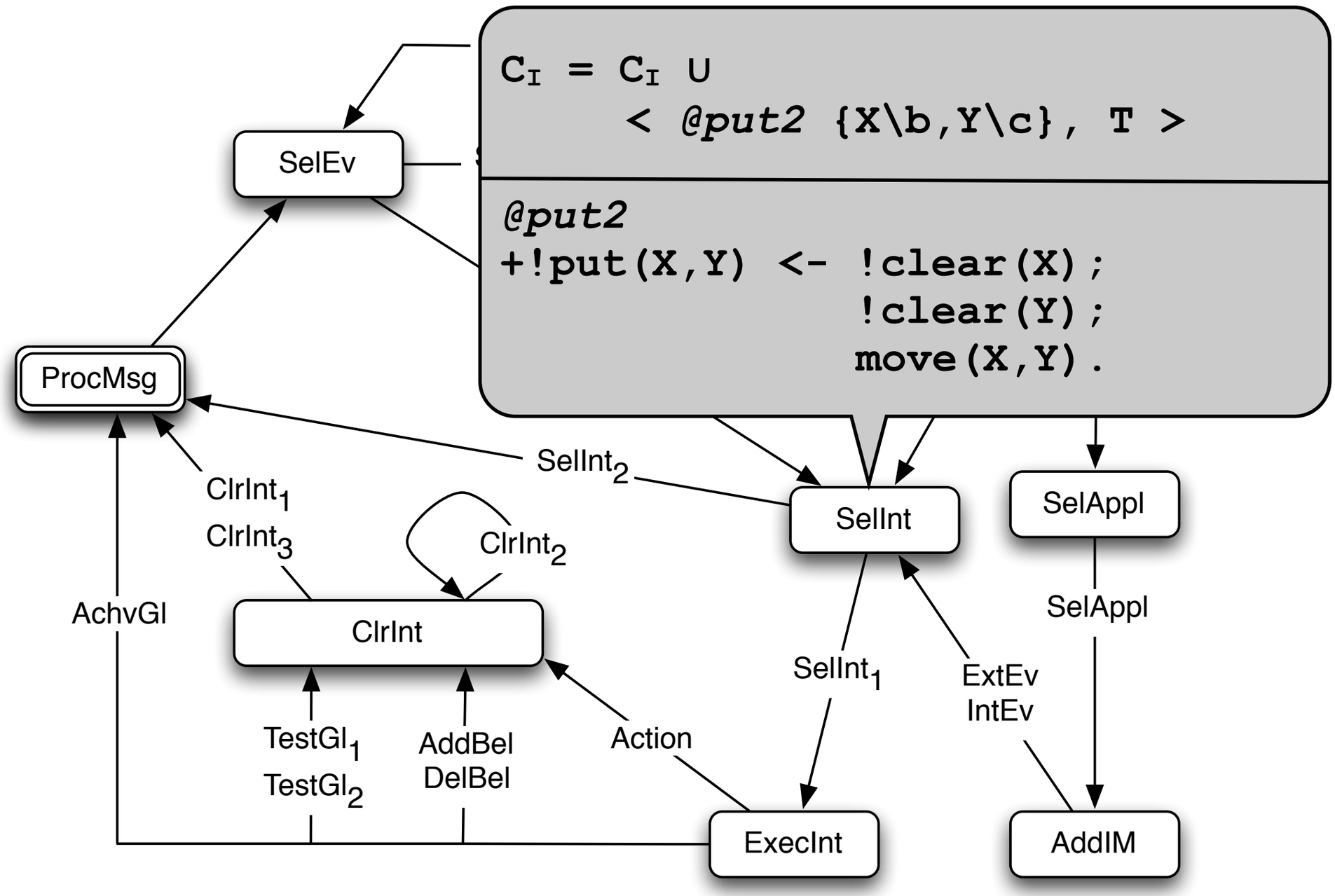
AgentSpeak(L): Select one Applicable Plan



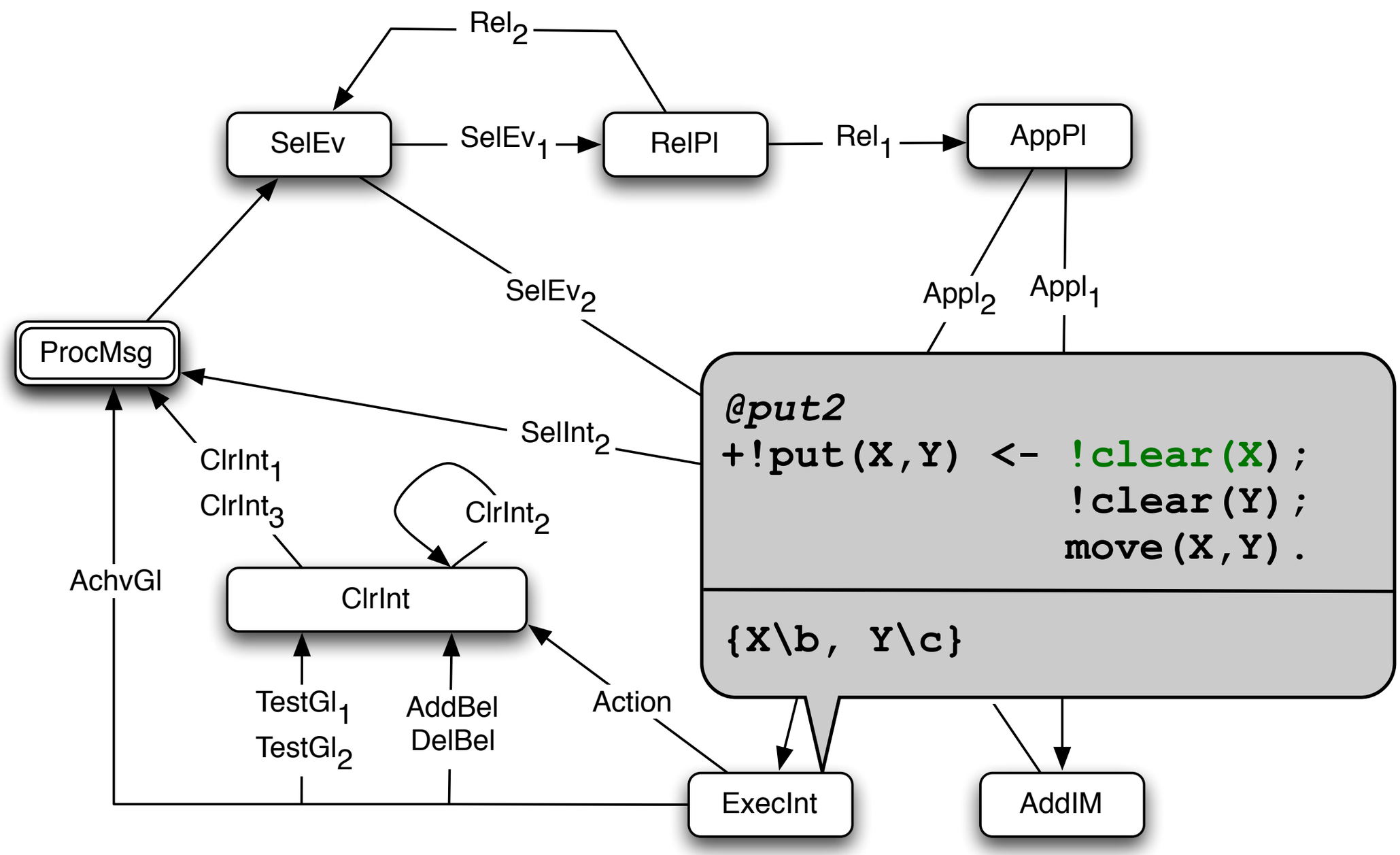
AgentSpeak(L): Add an Intention



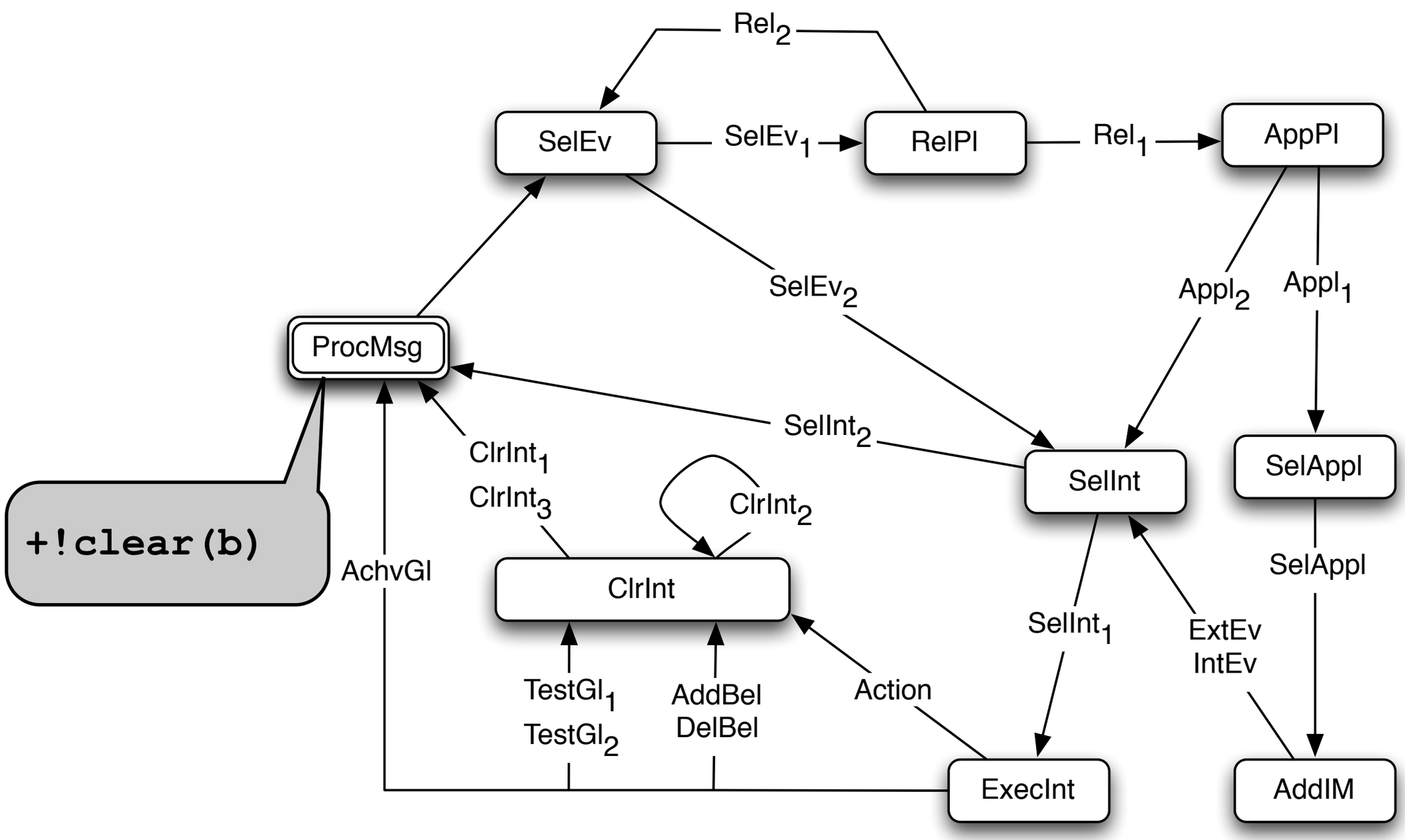
AgentSpeak(L): Select an Intention



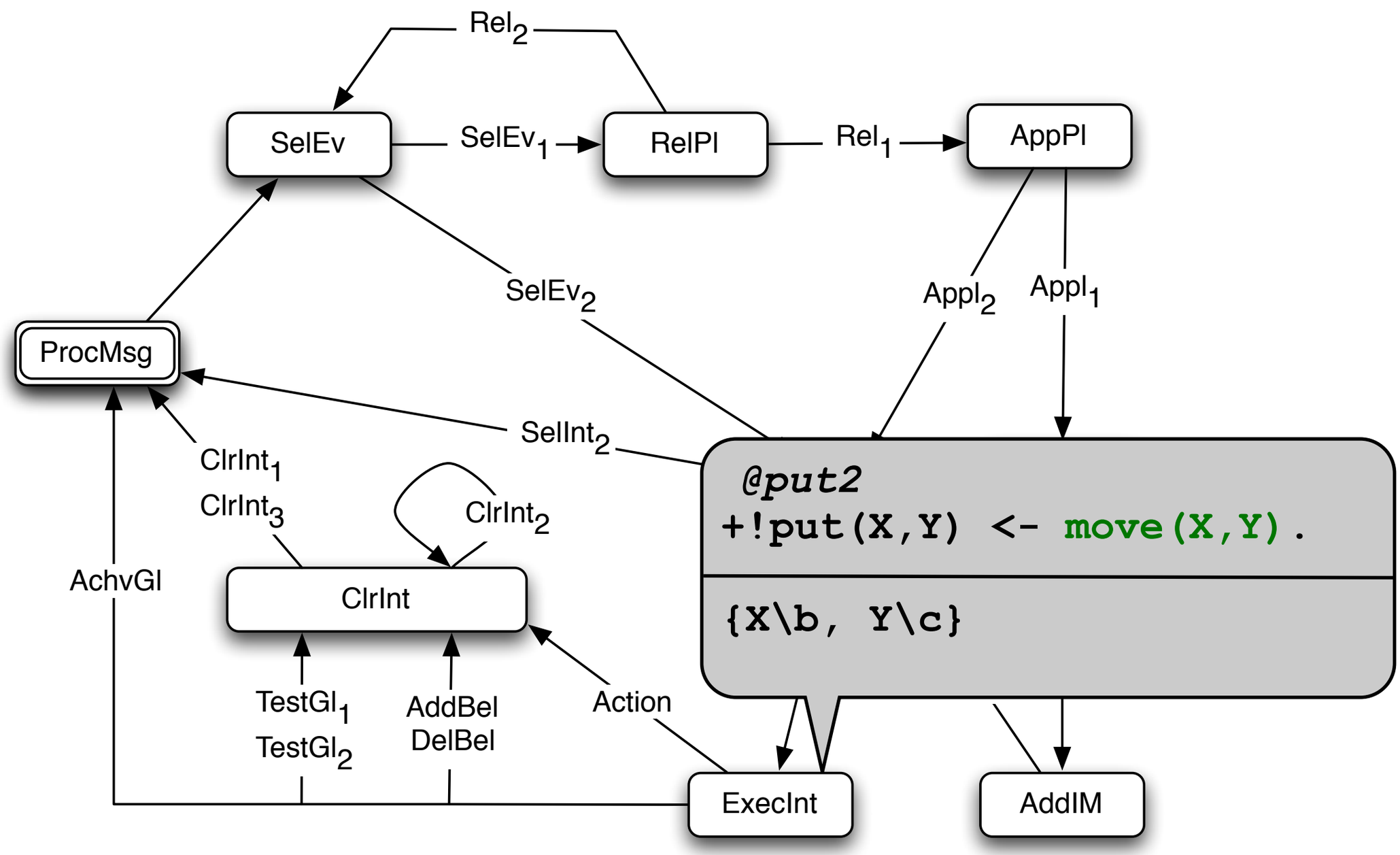
AgentSpeak(L): Execute the Intention



AgentSpeak(L): Events are added



AgentSpeak(L): If everything is ok...



AgentSpeak(L): ... But what if moves fails?

- The full intention fails!

- It is desirable to prevent such failures by learning a new context for the given plan, so that it won't be selected again when it is not applicable (Intentional Learning):

```
+!put(X,Y) : clear(X) & clear(Y) <- move(X,Y).
```

- It is desirable to learn abandon policies for Reconsidering intentions before they actually fail:

```
drop(put(X,Y)) :- .intend(put(X,Y)) & not clear(X).
drop(put(X,Y)) :- .intend(put(X,Y)) & not clear(Y).
```

AgentSpeak(L): Reconsideration

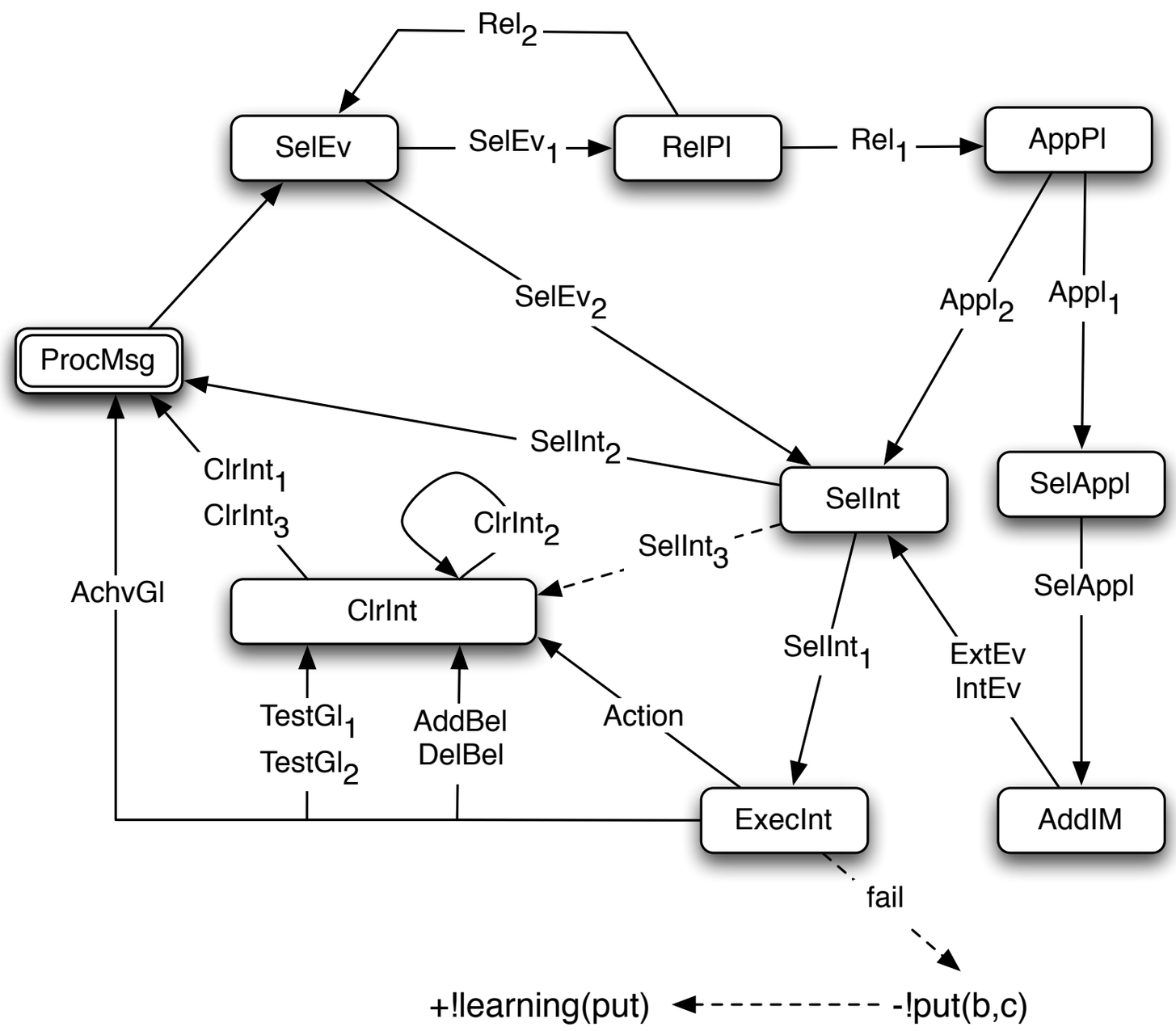
- Formally: Agents must be single minded committed:

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ (\mathsf{BEL}(\phi) \vee \neg\mathsf{BEL}(\mathsf{E}\Diamond\phi))$$

- Where after $\mathrm{BDI}_{\mathrm{CTL}}$, A = for all possible future, $\Diamond$ = eventually, and U = until.

- The reasons for not believing that in some future (E) eventually Φ are to be learned, and they cause that, when selected, the intention towards Φ is being dropped instead of executed.

AgentSpeak(L): Intentional Learning & abandon



Problems 1: The Learning Method

- What kind of learning method can these agents use, given that:
  - They use First-Order representations for Beliefs, Desires and Intentions.
  - The feedback for learning is the success or failure of executed Intentions.
  - The learning target (plan contexts) is possible conjunctive (it was in dMARS).
  - They have background knowledge to be exploited while learning.
- Top-Down Induction of Logical Decision Trees [Blockeel et al., 1999] in a dMARS like architecture [Guerra et al., 2005] including social learning.

Problems 2: Commitment

- But, what kind of commitment follows an AgentSpeak(L) agent in Jason? Ups, where are my Intentional and Temporal Operators?

- $CTL_{AgentSpeak(L)}$ [Guerra et al., 2008, 2009] A BDI-temporal Logic grounded on the Operational Semantics of Jason/AgentSpeak(L).

  - Agents satisfy the non-infinite deferral axiom: They eventually drop their intentions.

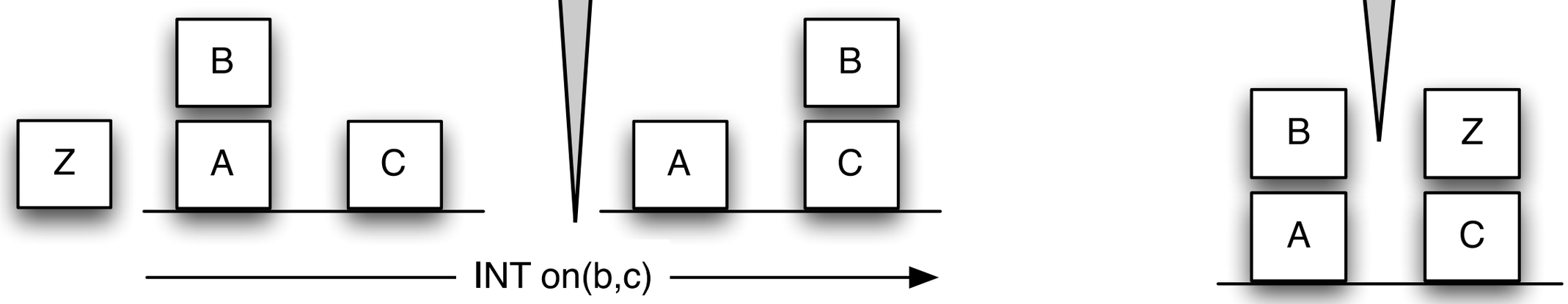
  - Agents do not satisfy the blind-commitment axiom: It is possible for them to abandon their intention for Φ without believing Φ.

  - Agents satisfy a limited version of the single-minded commitment: Explicit reasons to abandon an Intention are not represented anywhere, but they can be learned as rules.

Learning: Training Examples

```
begin(model(1))
  succ.
  intend(put,b,c).
  on(b,a).
  on(a,table).
  on(c,table).
  on(z,table).
end(model(1))
```

```
begin(model(2))
  fail.
  intend(put,b,c).
  on(b,a).
  on(a,table).
  on(c,table).
  on(z,c).
end(model(2))
```



Learning: Language Bias

```
rmode(clear(+V1)).
rmode(on(+V1,+V2)).
rmode(on(+V2,+V1)).
```

```
<-Q = intend(put,A,B).
```

```
clear(A).
clear(B).
on(A,B).
on(B,A).
```

Learning: Background Knowledge

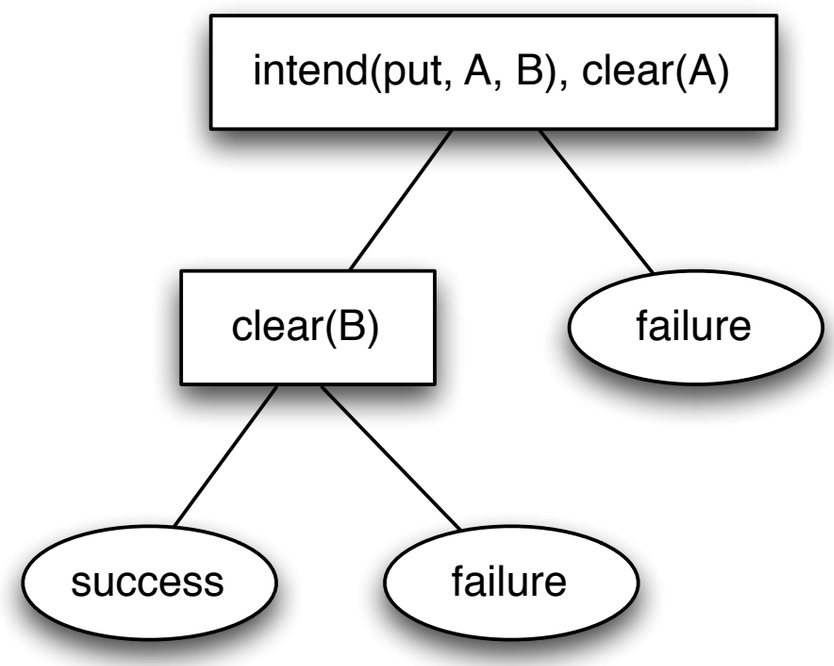
```
10 clear(X) :- not(on(_,X)).
11 clear(table).
```

Learning: Tilde (First-Order C4.5)

Algorithm 1 Top-down Induction of Logical Decision Trees.

procedure BUILDTREE(E,Q) ▷ E is a set of examples, Q a query
  $\leftarrow Q_b := best(\rho(\leftarrow Q))$ ▷ $best$ max information gain
  if $stopCriteria(\leftarrow Q_b)$ **then** ▷ E.g., No information gain obtained
    $T := leaf(majority\_class(E))$
  else
    $Conj \leftarrow Q_b \backslash Q$
    $E_1 \leftarrow \{e \in E \mid e \wedge B \models Q_b\}$
    $E_2 \leftarrow \{e \in E \mid e \wedge B \not\models Q_b\}$
    $buildTree(Left, E_1, Q_b)$;
    $buildTree(Right, E_2, Q)$
    $T \leftarrow nodei(Conj, Left, Right)$
  end if
  return T ▷ The built tree
end procedure

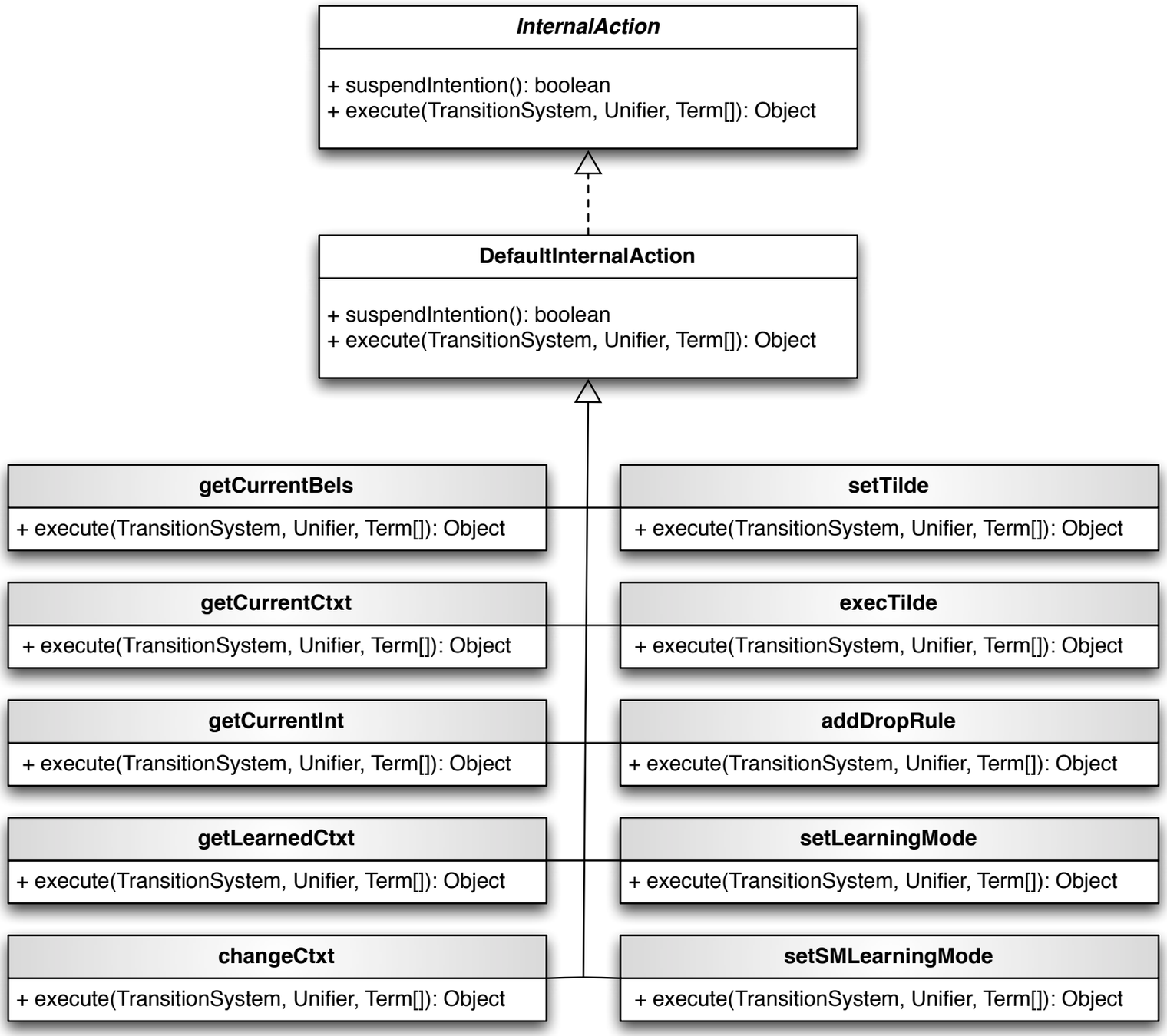
Learning: Logical Decision Trees




```
+!put(X,Y) : clear(X) & clear(Y)
    <- move(X,Y).
```

```
drop(put(X,Y)) :- .intend(put(X,Y)) & not clear(X).
drop(put(X,Y)) :- .intend(put(X,Y)) & not clear(Y).
```

Problems 3: Implementation [Guerra et al., 2010]



Implementation (JILDT): Extending Plans

```
@put
+!put(X,Y) : true <-
  jildt.getCurrentInt(I);
  jildt.getCurrentBels(Bs);
  +intending(I,Bs);
  move(X,Y);
  -intending(I,Bs);
  +example(I,Bs,succ).
```

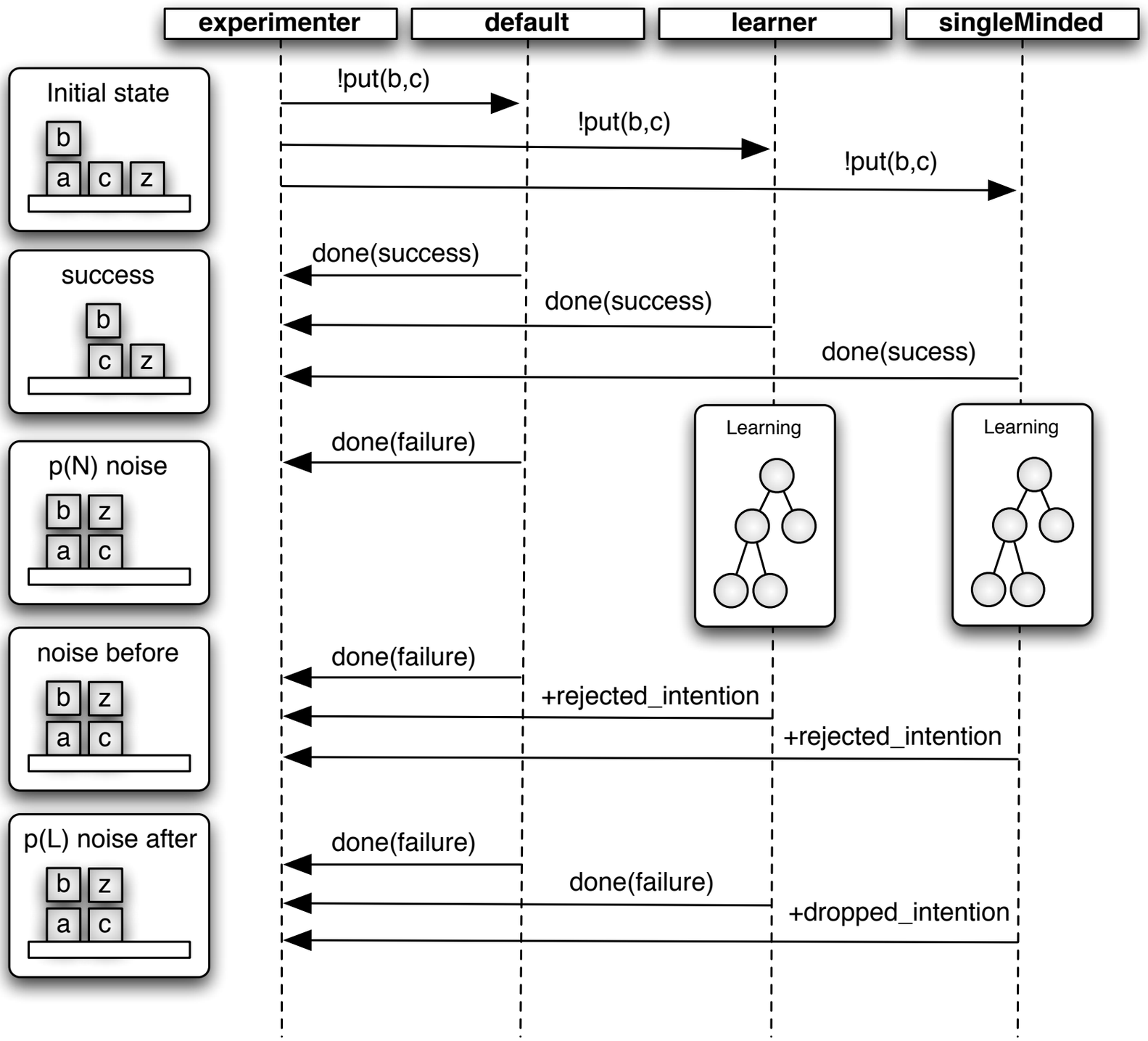
Implementation (JILDT): When to learn?

```
@put_failCase
-!put(X,Y) : intending(put(X,Y), Bs) <-
  -intending(I,Bs);
  +example(I,Bs,fail);
  !learning(put);
  +example_processed.
```

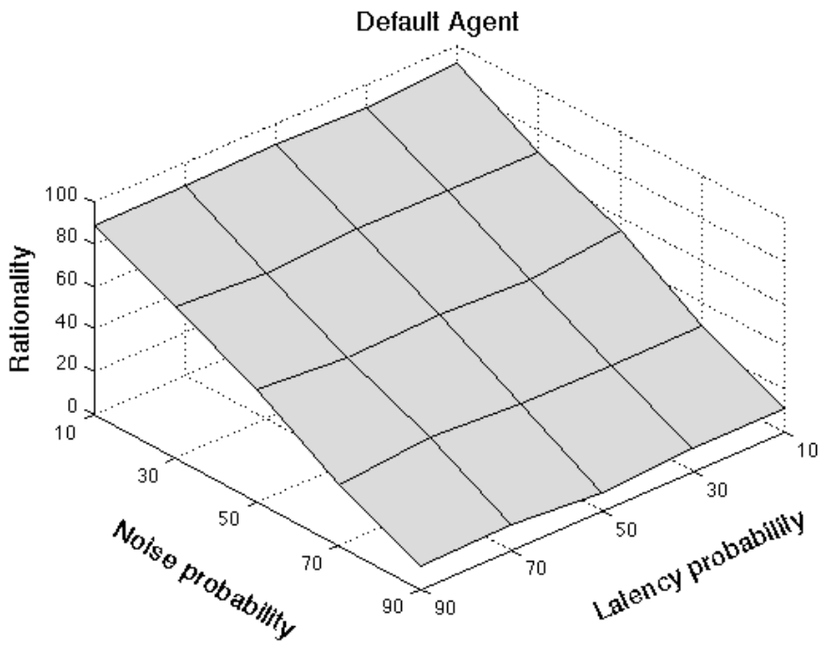
Implementation (JILDT): Learning

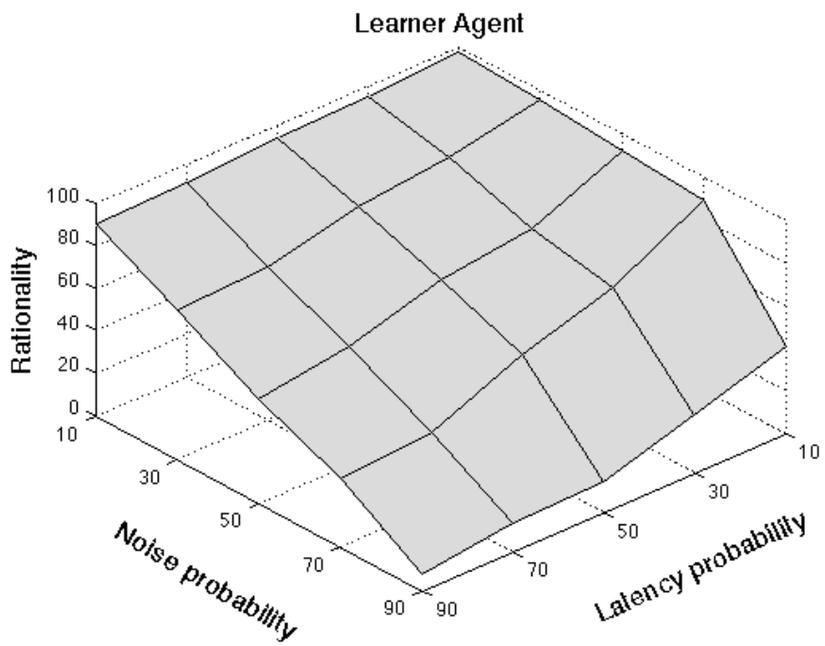
```
@learning
+!learning(P): true <-
    .print(\"Trying to learn a better context...\");
    jildt.setTilde(P);
    jildt.execTilde(false,false);
    jildt.getLearnedCtxt(P,LC,F);
    !learningTest(P,LC,F).
```

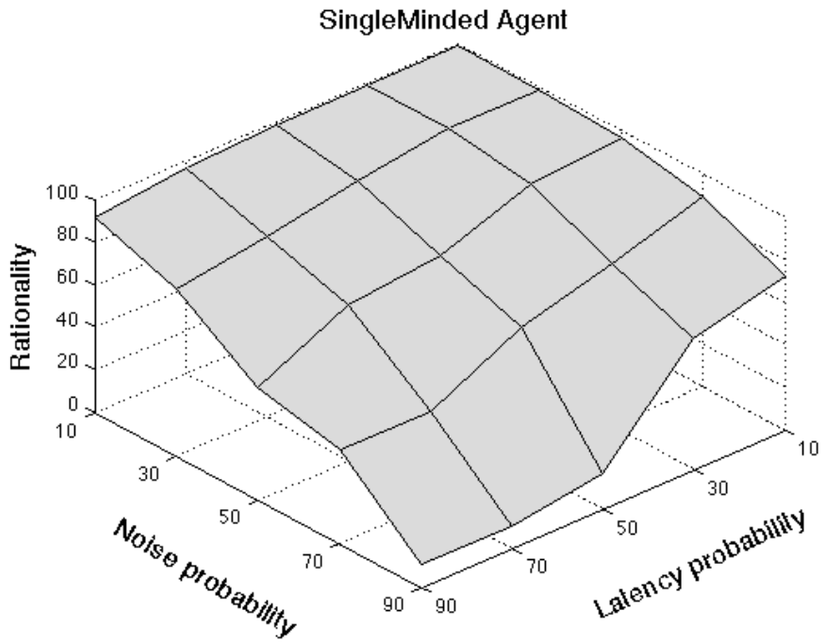
A simple experiment: `+!put(X,Y) <- move(X,Y).`



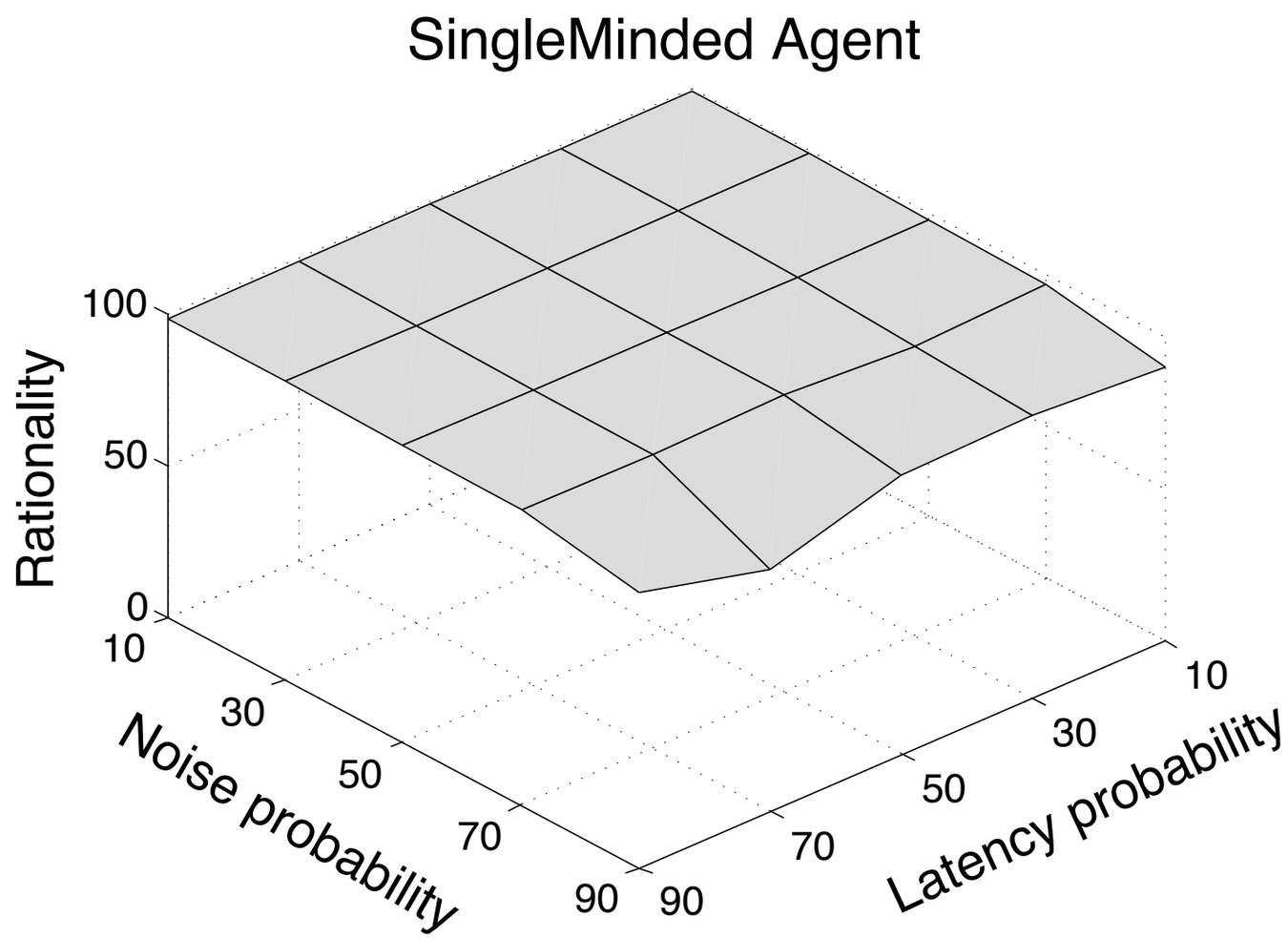
Results: Rationality, Noise, and Latency.







Results: Modified failure examples.



Current and Future Work

- JILDT is going to be released as a public library under GNU license, after some debugging and redesign (execTilde becomes a set of plans, and the action is broken in more atomic internal actions: Different strategies for learning, i.e., selecting best candidates, etc.) [González-Alarcón, et al. 2011]

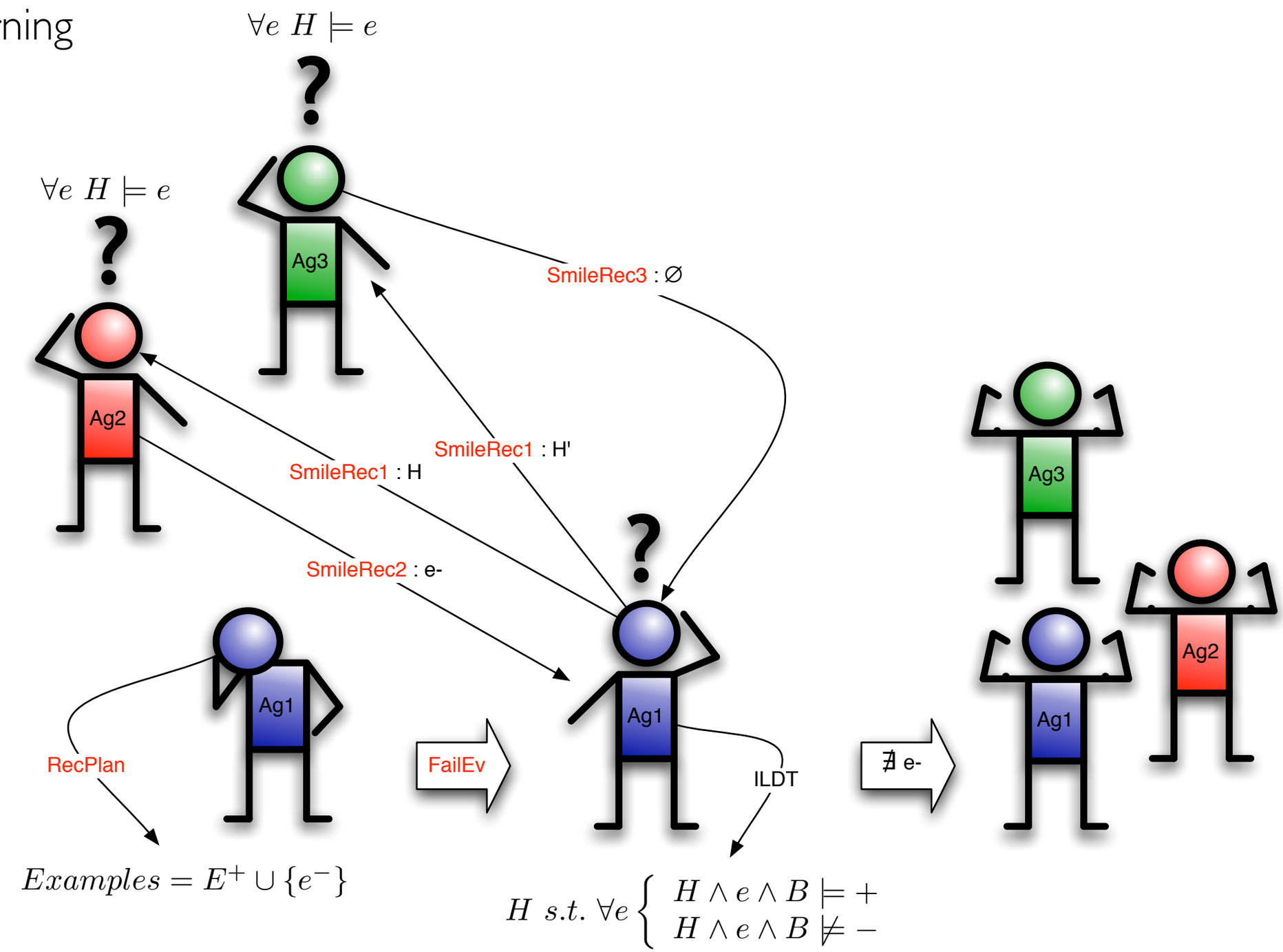
- JILDT is going to be tested within a more realistic scenario: Sharing vehicles to reduce CO2 emissions [J-MADeM Grimaldo, et al. 2011, 2012]

- A Theory about Intention maintenance is being studied: True reconsideration, instead of abandon [Castro-Manzano, et al. 2009, 2010, 2011]

- A discretization method is possibly required for the urban mobility scenario: an open problem given that training examples arrives incrementally and they are scarce. [Guerra, et al. 2012] : Learning from Data Streams, Randomized Binary Search Trees.

Current and Future Work

- Social Learning



References

- H. Blockeel, L. D. Raedt, N. Jacobs, and B. Demoen. Scaling up inductive logic programming by learning from interpretations. *Data Mining and Knowledge Discovery*, 3(1):59–93, 1999.

- R. H. Bordini, J. F. Hübner, and M. Wooldridge. *Programming Multi-Agent Systems in Agent-Speak using Jason.* John Wiley & Sons Ltd, 2007.

- A. S. Rao. Modeling Rational Agents within a BDI-Architecture. KR'91. Morgan Kaufmann Publishers Inc., p.p. 473-484, San Mateo CA., USA 1991.

- M. E. Bratman. *Intention, Plans, and Practical Reason.* Harvard University Press, Cambridge, MA., USA, and London, England, 1987.

References

- A. Guerra-Hernández, A. El-Fallah-Seghrouchni, and H. Soldano. Learning in BDI Multi-agent Systems. CLIMA IV, Revised and Selected Papers, volume 3259 of LNCS, 218–233, Berlin Heidelberg, 2005. Springer-Verlag.

- A. Guerra-Hernández and G. Ortíz-Hernández. Toward BDI sapient agents: Learning intentionally. In *Toward Artificial Sapience: Principles and Methods for Wise Systems*, pages 77–91. Springer, London, 2008.

- A. Guerra-Hernández, J. M. Castro-Manzano, and A. El-Fallah-Seghrouchni. CTL AgentSpeak(L): a Specification Language for Agent Programs. *Journal of Algorithms*, (64): 31–40, 2009.

- A. Guerra-Hernández, C. A. González-Alarcón, and A. E. F. Seghrouchni. Jason induction of logical decision trees. In MICAI 2010, Part I, volume 6437 of LNAI, 374–385, Berlin Heidelberg, 2010. Springer-Verlag.

Dedicated to

Amal El Fallah & Henry Soldano

& My students at the Maestría en IA

Contact

Dr. Alejandro Guerra Hernández

Profesor - SNI Nivel I - Perfil Deseable

Universidad Veracruzana
Facultad de Física e Inteligencia Artificial
Departamento de Inteligencia Artificial
Sebastián Camacho No. 5
Xalapa, Ver., México 91000
T (+52) 228 817-29-57
F (+52) 228 817-28-55

www.uv.mx/aguerra
aguerra@uv.mx • aguerrahdz@gmail.com