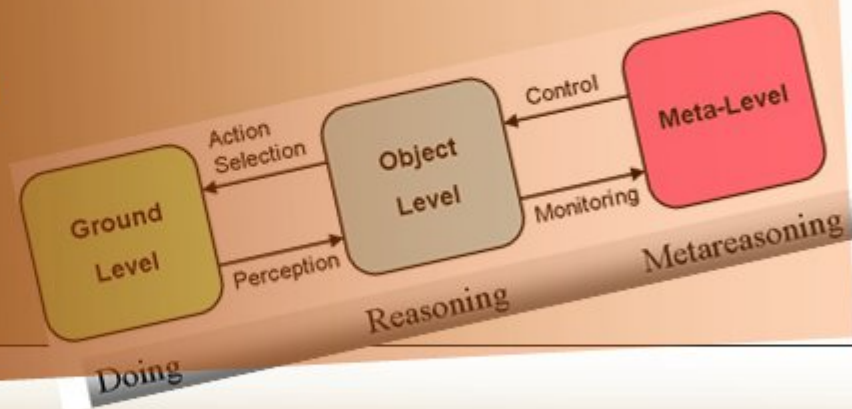


A Multi-layer Framework for Evolving and Learning Agents

Stefania Costantini

Pierangelo Dell'Acqua

Luis M. Pereira

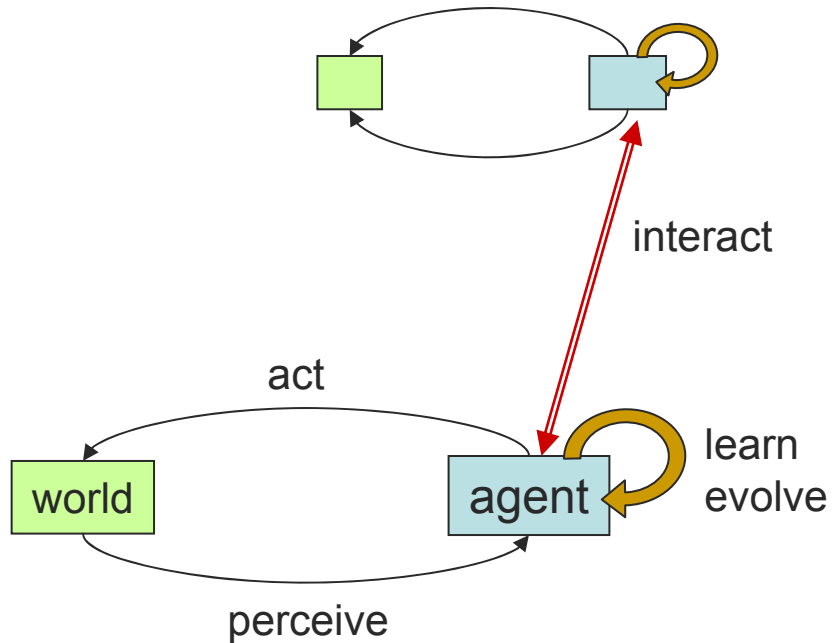


Metareasoning: Thinking about thinking Chicago 13-14 July 2008

Aim

- To illustrate a general model for agents that can learn and evolve
- The model consists of three layers:
 - an object level BA
 - a meta-level MA
 - a meta-meta level IEA
- MA performs various forms of meta-reasoning including learning and meta-control over BA
- IEA handles the agent's social interactions

Agent features



interact

- by knowledge exchange

learn

- by experience
- by imitation (by being told)

world

- dynamic
- unpredictable

evolve

- by acquiring new knowledge

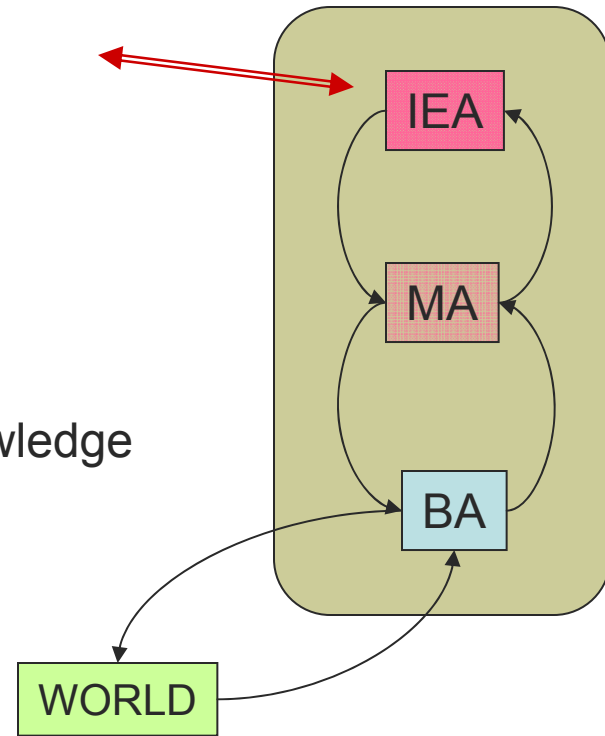
Agent layers

Meta-agent MA (meta-level)

- Supervises BA activities
- Evaluates BA
- Learns by experience
- Modifies BA behavior by changing its knowledge

Base agent BA (object level)

- Acts/perceives the world (ground level)

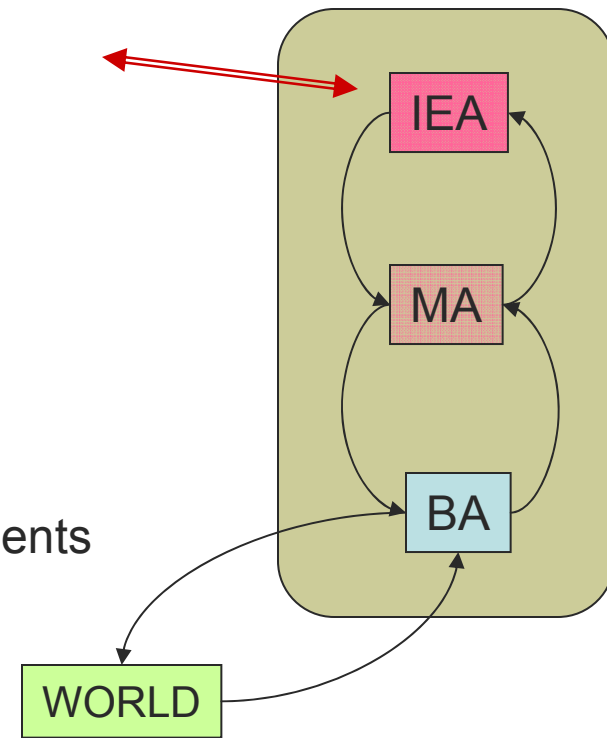


Agent layers

Information exchange agent IEA

(meta-meta level)

- Responsible for social interaction
- Knowledge exchange
- Learn:
 - by imitation
 - by eliciting behavioral patterns of other agents



Agent operational behavior

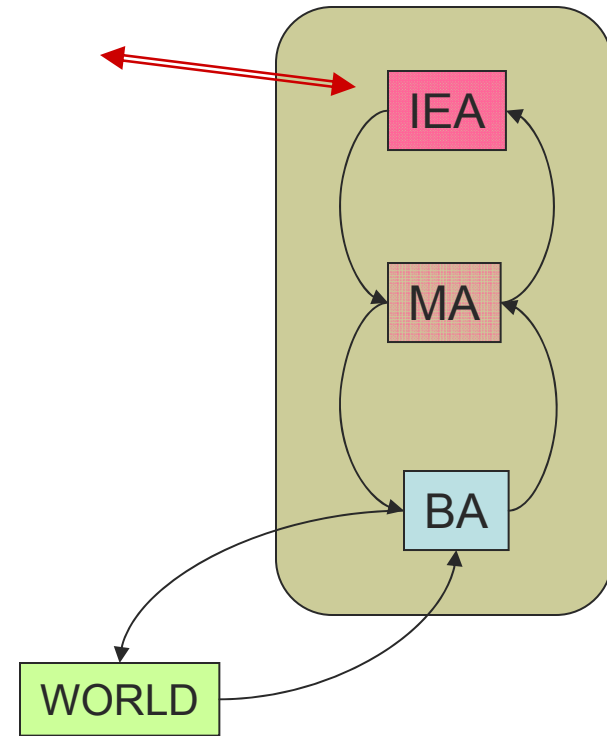
- Agent model: $\Gamma = \langle BA, MA, IEA, C, CI \rangle$
 - BA, MA, IEA: logic programs
 - C: control component
 - CI: control information

- Underlying control mechanism U

$$\Gamma_i \xrightarrow{U(C_i, CI_i)} \Gamma_{i+1}$$

Implements operational behavior

- Upward/downward reflection
 - Updates agent model
- EVOLP



Meta agent MA

- MA supervises BA
 - Verifies properties of BA at run-time
(not in advance in the model checking style)
 - MA formalized via linear time logic extended with time intervals, A-ILTL
 - A-ILTL allows to express constraints whose violation may require the undertaking of suitable actions (repairs)

A-ILTL meta-rules

S: set of literals

conj(S): set of conjunctions of literals in S

OP(m,n;k): temporal operator

Define Q as:

- $S \subset Q$
- if $\varphi \in \text{conj}(Q)$, then $\text{OP}(m,n;k) \varphi \in Q$

A-ILTL meta-rule

$\text{OP}(m,n;k) \varphi$ for any $\varphi \in \text{conj}(Q)$

A-ILTL meta-rules

Contextual A-ILTL rule

$OP(M,N;K) \varphi :: X$

X evaluation context

Contextual A-ILTL rule with repairs

$OP(M,N;K) \varphi :: X \dot{\div} a$

a is a repair action

A-ILTL operators

NOW(t)

holds if t is the current state

NEXT(j) φ

φ should be true at state j

FINALLY(j;k) φ

φ eventually has to hold somewhere on the path from the current state to j

ALWAYS(m,n;k) φ

φ should become true in m and hold until n (false in n+1)

NEVER(m,n;k) φ

φ should be false in any state between m and n

Example: meta-control

Monitoring goals' achievement

NEVER (not achieved(G), dropped(G)) ::

(goal(G), deadline(G,T), NOW(T1), $T1 \leq T$) \div inc_comt(T1)

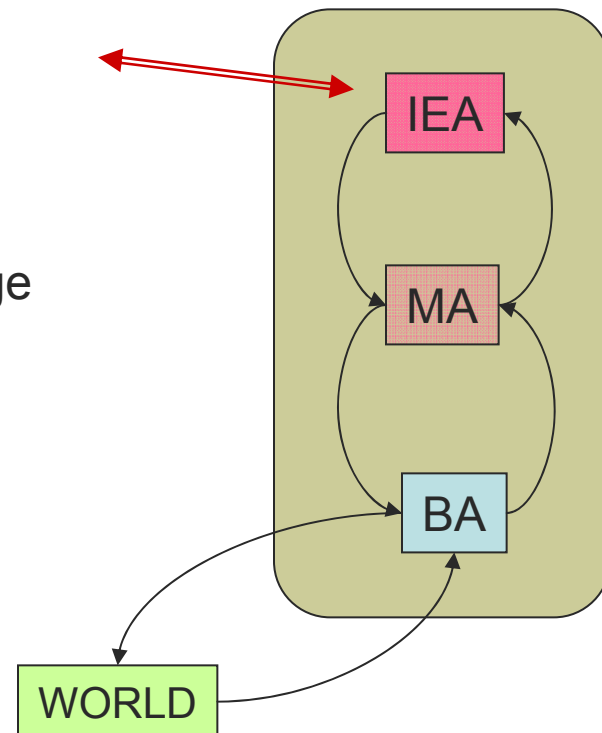
Meta-level decision: increase level of commitment

incr_comt(T) \leftarrow level(commitment, L),
 increase_level (L, L1),
 assert(neg(commitment_mod(L)),
 assert(commitment_mod(L1)),
 assert(incr_comt_at(T))

Semantically, the execution of the repair will determine the update of the current agent theory

Learning and evolution

- When performing monitoring/training of a user, the agent performs three learning activities:
 - Initialization
 - Observation
 - Interaction
- These activities require information exchange



Learning and evolution

Initialization phase

- The agent receives from other agents/society rules defining:
 - Basic agent behavior
 - The role that the agent will impersonate in the society
- This stage will provide general meta-rules to be included in MA

Learning by being told

Personal assistant

FINALLY(T) A :: (action(A), mandatory(user,A), timeout(A,T))

NEVER A :: (action(A), forbidden(user,A))

Learning and evolution

Observation phase

- The agent observes the environment
- Tries to classify its observations to elicit general rules

Learns by experience

- Introspect on possible outcomes
 - Prediction (looking ahead)
 - Trigger meta-control alerts

Learning and evolution

Interaction phase

- In case of need, the agent tries to obtain the necessary knowledge from the society
 - Verifies the adequacy of the learned rules
 - Revise/retract them in face of new evidence (evolving capabilities)
- The agent contributes to the society by:
 - Evaluating learned knowledge
 - Providing feedback on the usefulness/effectiveness of knowledge
 - Providing its own knowledge

Modeling imitation learning

- IEA responsible for information exchange
- Activated when an agent A asks B:
 - how to solve a certain problem
 - how to improve its performance
 - and when A proactively recommends B relevant information
- Exploits techniques for:
 - Reputation/trust measures
 - Social evaluation and consensus
 - Compatibility and performance of new knowledge
- Uses evaluation of resources of MA

Modeling imitation learning

Information exchange

- IEA maintains experience repository
 - Trust/reputation ratings of agents
 - History of exchanged information (Log)
- Information exchange protocol consists of 4 phases

Modeling imitation learning

Information exchange protocol

■ Phase1: Resource searching

- Information request
- Agent reply $x_i=(\underline{x}, E_i)$

\underline{x} : answer to the query

E_i : evaluation (meta-knowledge) of \underline{x} performed by agent A_i :

- prerequisites and objectives of \underline{x}
- conditions for which \underline{x} is not suitable
- risk, gain, and probability of success of \underline{x}
- performance results

Modeling imitation learning

Information exchange protocol

- Phase2: Resource selection and vote polling

- Selection

Evaluating received answers:

$$w_i^x = f(x_i, A_i)$$

f is an evaluation function specific to agents

- depends on trust and reputation of proposing agent A_i

- Vote polling

Before acceptance the agent may ask other agents' opinion about the answer as well as the proposing agent

Modeling imitation learning

Information exchange protocol

- Phase3: Vote evaluation

- Weight of the vote:

$$w_x^S = H_{i \in S}(w_i^x)$$

S is the set of responding agents to the voting

H can be an average function

Modeling imitation learning

Information exchange protocol

- Phase4: Resource acceptance
 - If resource quality not sufficient
repeat voting process on another available resource
 - Otherwise, the agent accepts the resource:
 - Register it into its Log
 - Send it to the MA
 - Set a posteriori resource evaluation