

Formal ReSpecT



Andrea Omicini,
Enrico Denti

DEIS – Università di Bologna, Italy

Outline of this Talk



- Goal
- Agent infrastructures
- Objective coordination as a service
- ReSpecT logic-based coordination
- Formal characterisation – *sketched*
- Open issues & conclusions

Goals of this Work



- Focus on agent infrastructures
- Which role for logic-based languages?
- What formalisation is meant to?
- Formalising ReSpecT
 - logic tuple centres
 - logic-based specification language

Issues from Agent-Oriented SE

- Ontology
 - individual agents
 - agent societies
 - agent environment
 - Internet-based MAS
- Agent-Oriented Software Engineering
 - infrastructures for agent-based systems

Agent Infrastructures



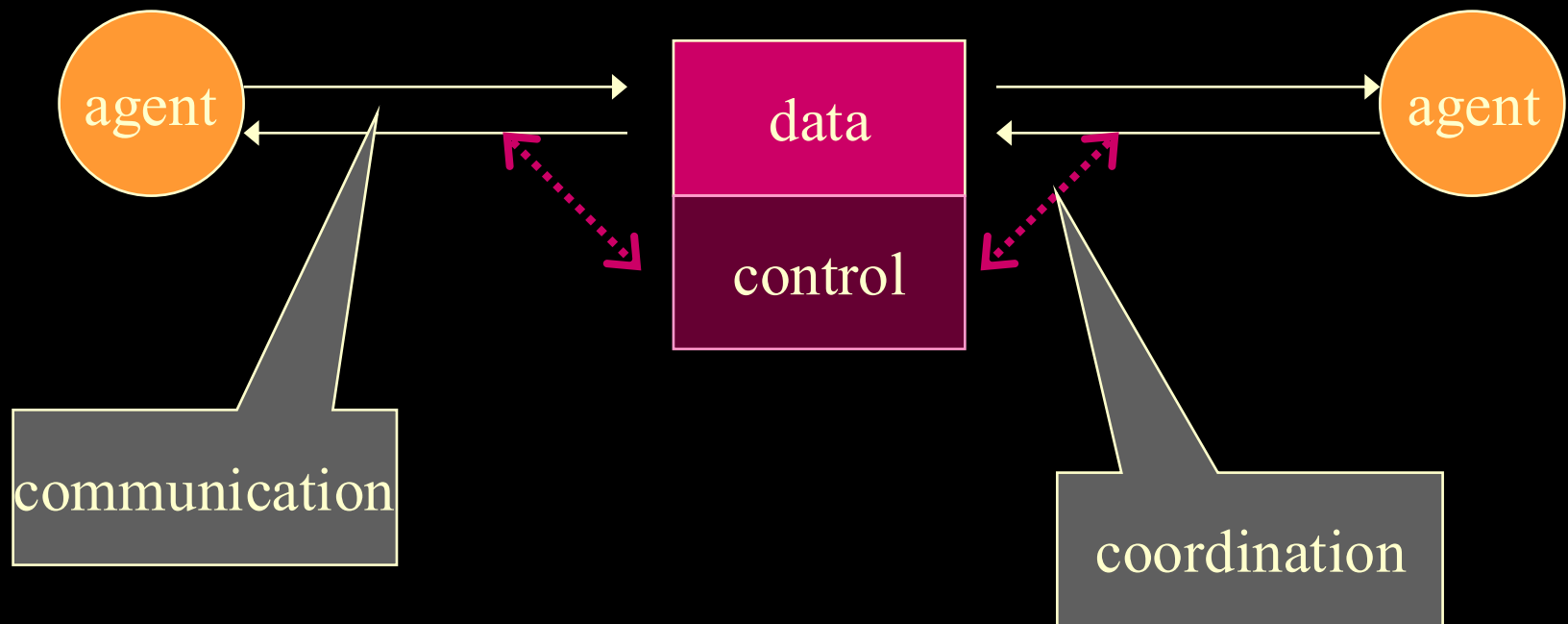
- Factorising MAS common needs
 - application independent *services*
- Focus on interaction
 - among agents
 - between agents and environment
- Multi-service, multi-purpose
 - trade-off btw. generality and specialisation
- Configurability of services

Coordination Service



- Objective vs. subjective coordination
 - coordination model
- Coordination as a service
 - coordination medium
 - MAS-independent coordination run-times
- Example: Linda
 - model, medium, run-times

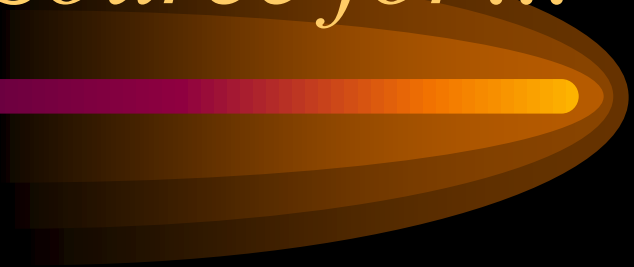
Objective Coordination



A Place for Logic Languages

- OO-based SE
 - weakness in programming in the large
 - integration of OO and LP
- Component-based SE
 - small, heterogeneous components
 - LP for small specialised components
- AOSE
 - logic languages → declarative KR
 - logic inference → deliberation/intelligence

LP as a Natural Source for...



- Computation languages
 - building logic-based agents
- Communication languages
 - representing information
- Coordination languages?
 - representing social laws

Logic Tuple Centres

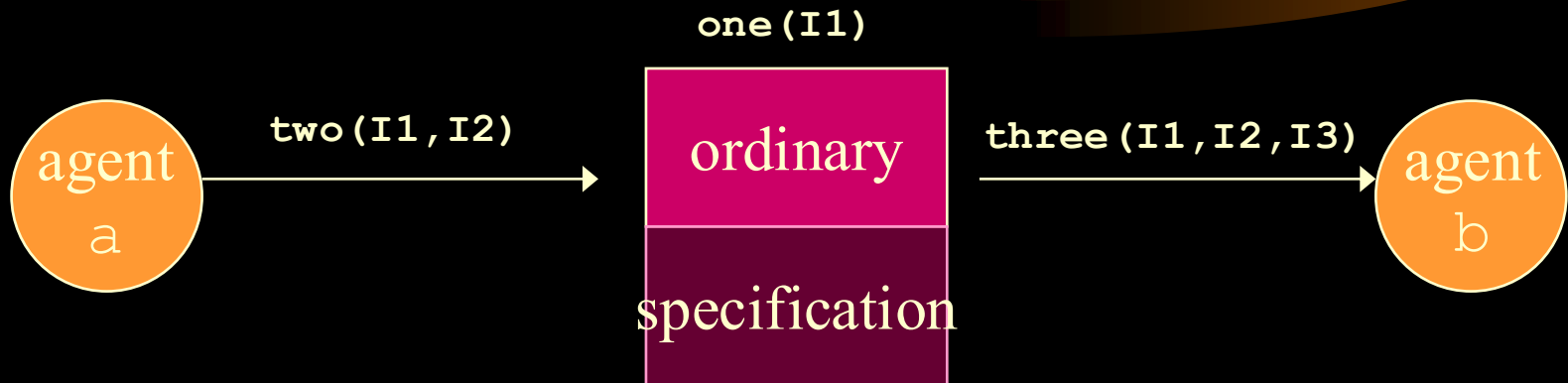


- Logic tuple space
 - logic tuples as the communication language
 - coordination through logic tuples
- Programmable behaviour specification
 - through *reactions* to communication events
 - ReSpecT
 - a logic-based language
 - Reaction Specification Tuples

ReSpecT Tuple Centres

- Logic tuples
 - for both communication and coordination
 - `map (OpName, Event), react (Event, ReactBody)`
 - `reaction (Operation, ReactBody)`
- Logic tuple centre composed by
 - logic *tuple space*
 - logic *specification space*
- Execution model
 - reactions as *transactions*
 - communication events *plus* triggered reactions perceived atomically at the agent level

Showing some ReSpecT



```
reaction(out(two(I1,I2)), (  
    current_agent(a), in_r(two(I1,I2)),  
    out_r(one(I1)), out_r(one(I2))  
))
```

ReSpecT: Main Predicates



Formalisation & Infrastructures



- Formal models in agent systems
- Openness, autonomy, unpredictability
 - black boxes and white boxes
- Infrastructures as invariants
- Services
 - standardisation
 - formalisation

Formalising Interaction: Principles



- Output events
 - speaking
- Input events
 - listening
- Interaction events
 - evolving
- Agents as black boxes
 - observable behaviour
- Coordination media as input/output automata
 - full behaviour specification

Formalising Tuple Centres

- Admissible communication event

$$o^a?_n t$$

- Agents as event emitters/receivers
- Logic tuple centres as transition systems

$$\langle T, W, Z \rangle_{\sigma}$$

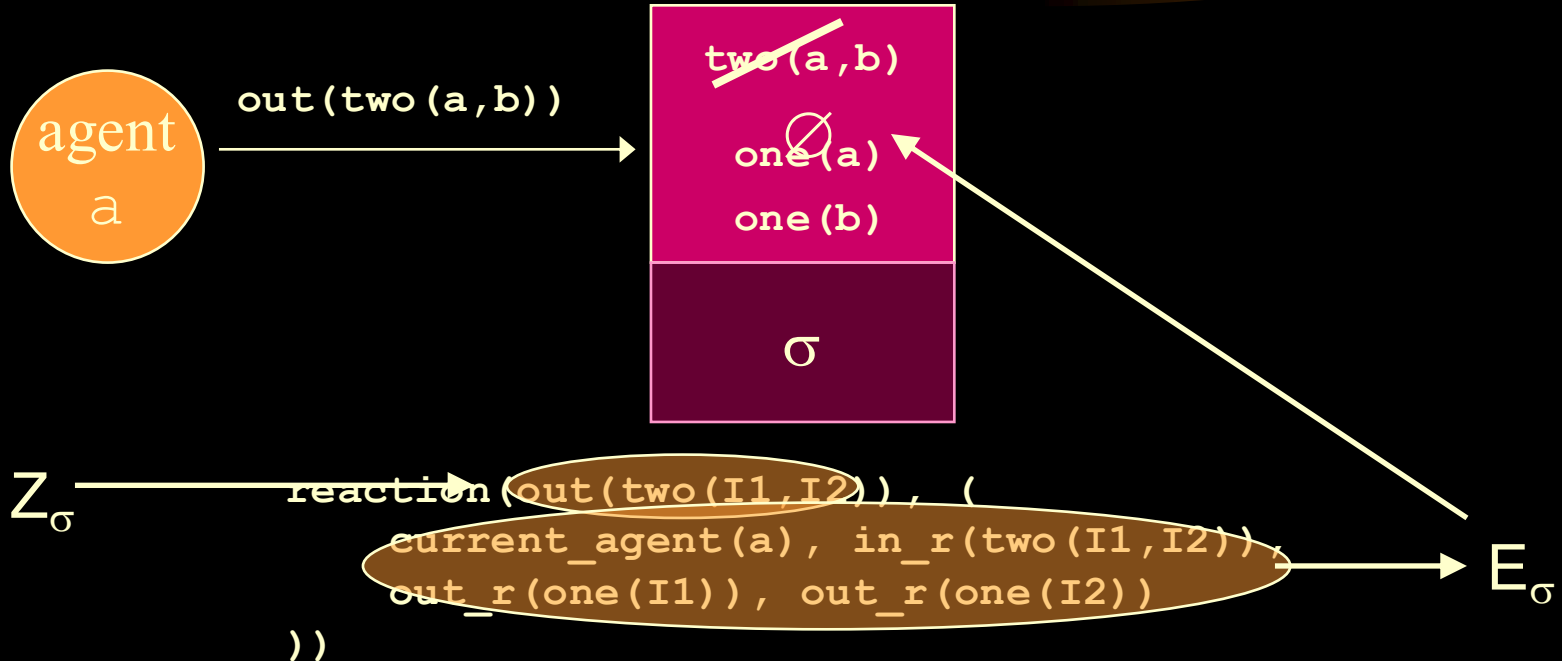
+

listening, speaking, reacting transitions

Formalising ReSpecT

- Transition system parametrised through
 - reaction specification function Z
 - reaction evaluation function E
- $Z_{\sigma}(e)$
 - reactions triggered by event e according to σ
- $E_{\sigma}(z, T) \rightarrow (T', Z')$
 - new tuple space state
 - newly triggered reactions

Showing further ReSpecT



Lessons Learned

- Formalisation of agent infrastructures
 - open, autonomous, unpredictable systems
 - models & languages for configurable services
 - interaction-oriented formal frameworks
- Logic-based languages are effective tools for agent infrastructures
 - declarative configuration
 - good trade-off
 - high-level procedural languages
 - unification
 - easy formal specification

Open Issues (1)

- Trade-off btw. general purpose abstractions and specialised *ad hoc* mechanisms
 - expressiveness & generality vs. efficiency?
 - knowledge mediation, security, ..., whatever?
- Topological abstractions
 - supporting intelligent agent exploration
 - enabling & constraining exploration
 - e.g., explicit representation of coordination laws

Open Issues (2)



- Security / Identity
 - granularity
 - agent / societies / MAS / environments
 - declarative specification of security policies
- Mixed feelings
 - Java-based vs. logic-based?
 - hybrid languages & models

ReSpecT Home Page



ReSpecT home page

<http://lia.deis.unibo.it/research/ReSpecT/>

Forthcoming: Open Source

- tuProlog
- ReSpecT
- LuCe
- TuCSoN