



Spaces for Programming Languages in MASs

A Perspective on Coordination, Organisation & Security

Andrea Omicini

Alma Mater Studiorum–Università di Bologna a Cesena

andrea.omicini@unibo.it

Joint Research with Alessandro Ricci & Mirko Viroli

Outline

- [Spaces for Programming Languages in MASs
- [Coordination, Organisation & Security
- [Languages & Artefacts for Governing MAS Interaction
- [Our Traditional Example [Cut Down]
- [[Preliminary] Conclusion

Spaces for Programming Languages in MASs

SE & PL

[New Classes of Programming Languages Come from Paradigm Shifts in Software Engineering

- new meta-models / new ontologies for artificial systems build up new spaces
- new spaces have to be “filled” by some suitably-shaped new (class of) programming languages
- NOTE: SE here is taken in its broadest acceptation as the science of building software system, rather than the strange “theoretically practical” discipline you find at ICSE... Otherwise, one may easily see the thing the other way round

[The Typical Procedure

- first, existing languages are “stretched” far beyond their own limits, and become cluttered with incoherent abstractions and mechanisms
- then, academical languages covering only the some of the issues are proposed
- finally, new well-founded languages are defined, which cover new space well

Spaces from Agent-Based Systems

— [OK, What is an Agent?

— just joking :) But! Whatever an agent is, it definitely opens up new spaces

— [It is not just Languages for Programming Agents

— which more or less is the well-known story: first using old languages, then AOP and company, finally... 3APL? ASK MEHDI!

— and is deceptively near to “classical” programming languages

— [It was Obviously Languages for Agent Communication

— why no one talked about languages for object communication?

— [And Now it is About Languages of Many Differents Sorts

— coordination languages, WS languages, KR languages, ...

— [How to Put These Languages in Context?

A Simple Yet Powerful Metamodel for MASs

- [If an Agent Can Only Speak

- then MASs are to be made of agents only

- [But if they Act on Something

- then MASs should be made of something else, too

- [Agents & Artefacts

- you may already have heard about it somewhere else;)

- A MAS is a collection of *agents* and *artefacts*

- [What is an Artefact?

- well, what is an agent? :) no, really.

Artefacts in MASs

- [An Agent is a Computational Entity

- encapsulating control flow
- along with a criterion to drive control
 - say, a task, a goal, ...
 - so, autonomous by definition

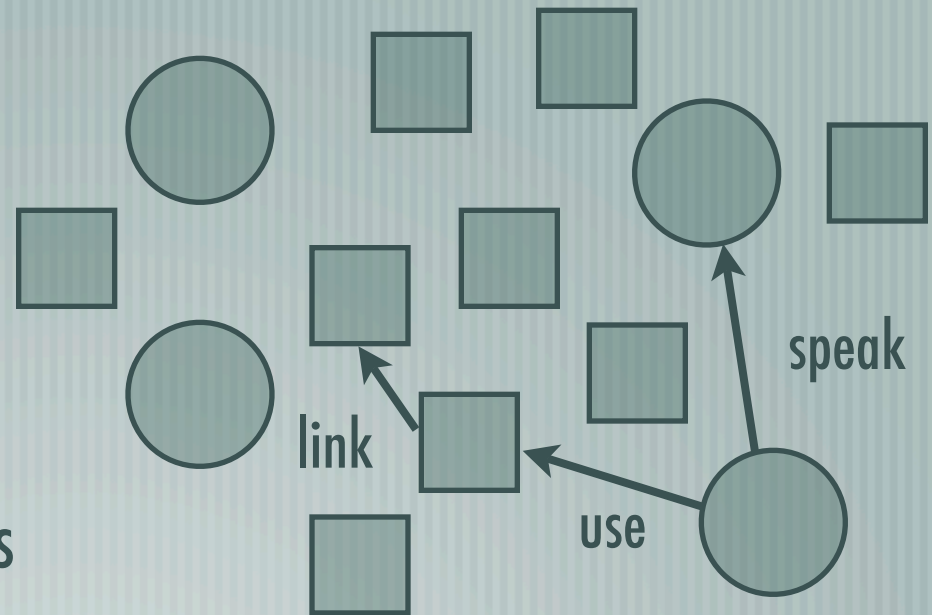
- [An Artefact is a Computational Entity

- without its own control flow
 - so, it is reactive by definition
- which can be *used* by agents for their own purposes

- [Artefacts Define the Agents' *Workspace*

- or Field of Work, or ... [Susi 2004, Schmidt 2005]
- they represent the articulation of the agent environment
 - encapsulating the responsibilities delegated to the agents' workspace

So, What Happens in a MAS?



Agents *Speak* with Agents
Agents *Use* Artefacts
Artefacts *Link* with Artefacts

This three Sentences Fully Describe *Interaction* within MASs

And So, What about Spaces for PL?

- Languages to *Be*, Languages to *Interact*

- Languages to Be

- languages for agents

- agent programming languages

- languages for artefacts

- *artefact programming languages*

- Languages to Interact

- languages to speak

- ACLs

- languages to operate (use, acting over artefacts)

- models for agent actions over artefacts (see operating instructions in Viroli)

- languages to link

- which should be somehow consistent with the action and the artefact models

Uh. Where is Coordination, then? And Organisation? I got Lost :(

— [They are Where Interaction is

— so, languages to interact

— [But!

— interaction in a MAS is first of all agent interaction

— interacting autonomous flow of control

— [And

— artefacts are mediators

— enablers / rulers

— for agent interaction

— Activity Theory, cognitive sciences, CSCW, HCI...

— [So

— languages for artefacts are at play, too

— they play a central role in managing interaction in MAS

Coordination, Organisation & Security

Coordination

- [Many Different Definitions

- need to simplify, here

- [In Short, Managing Interaction in any Possible Way

- from any viewpoint

- [It has a Typical “Dynamic” Acceptation

- enabling / governing interaction at execution time

- [Coordination in MAS

- chaos

- a useful definition to harness the many different acceptations in the field

- *subjective* vs. *objective* coordination

- agent’s vs. artefact’s viewpoint

Organisation

— [Again, A Not-so-clear and Shared Definition

- but mainly it concerns the *structure* of a system
- mostly design-driven
- which affects and determines admissible / required interactions
- permissions / commitments / policies / violations / fines / ...

— [So, It is Still Enabling & Ruling Interaction

- but with a more “static”, structural flavour
- such that most people mix-up “static” and “organisation” improperly

— [Organisation in MAS

- first of all, a model of responsibilities
- roles (RBAC-MAS)
- requiring a model of actions
- communication and operation

Security

— [You May not Believe it,
but Also Security means Managing Interaction

— you cannot see/do/say this, you can say/do/see that

— [Typically, Security has Both “Static” and “Dynamic” Flavours
— a design- plus a run-time acceptance

— [But Tends to Enforce a “Negative” Interpretation over Interaction
— “this is not allowed”

— [It is Dual to Both Coordination and Organisation

— so, in MAS at least, it is just time to look at them altogether

Coordination, Organisation & Security

— [Can This Agent Speak to That Agent / Use That Artefact?
— in case, will it (deliberate to) speak / use?

— [Said that, We Have Seen them All

— but this is not the point today – just the starting point

— it is more generally “Managing Interaction in a MAS”

— [They All Mean to Shape the Space of Agent Interaction

— to define its admissible space at design time (Organisation/Security flavour)

— to govern its dynamics at run time (Coordination/Security flavour)

— [The Main Point Here is that Artefacts are our Conceptual Tools to
Articulate & Shape MAS Environment

— to define the agent workspace

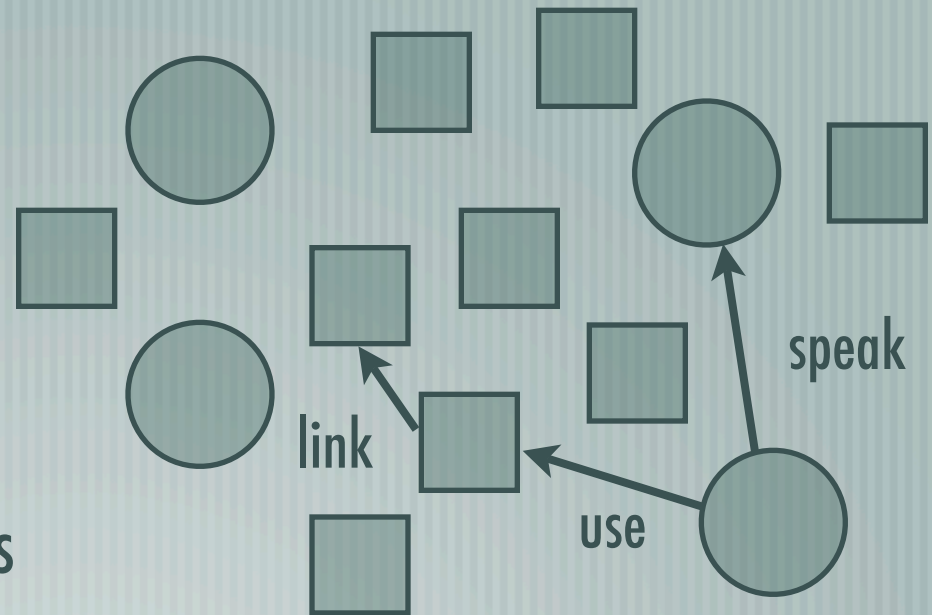
— to govern agent interaction

— [Here is Where Coordination/Organisation & Environment *Clash*

— Shaping the Environment with Artefacts embodying Coordination/Organisation

Languages & Artefacts for Governing MAS Interaction

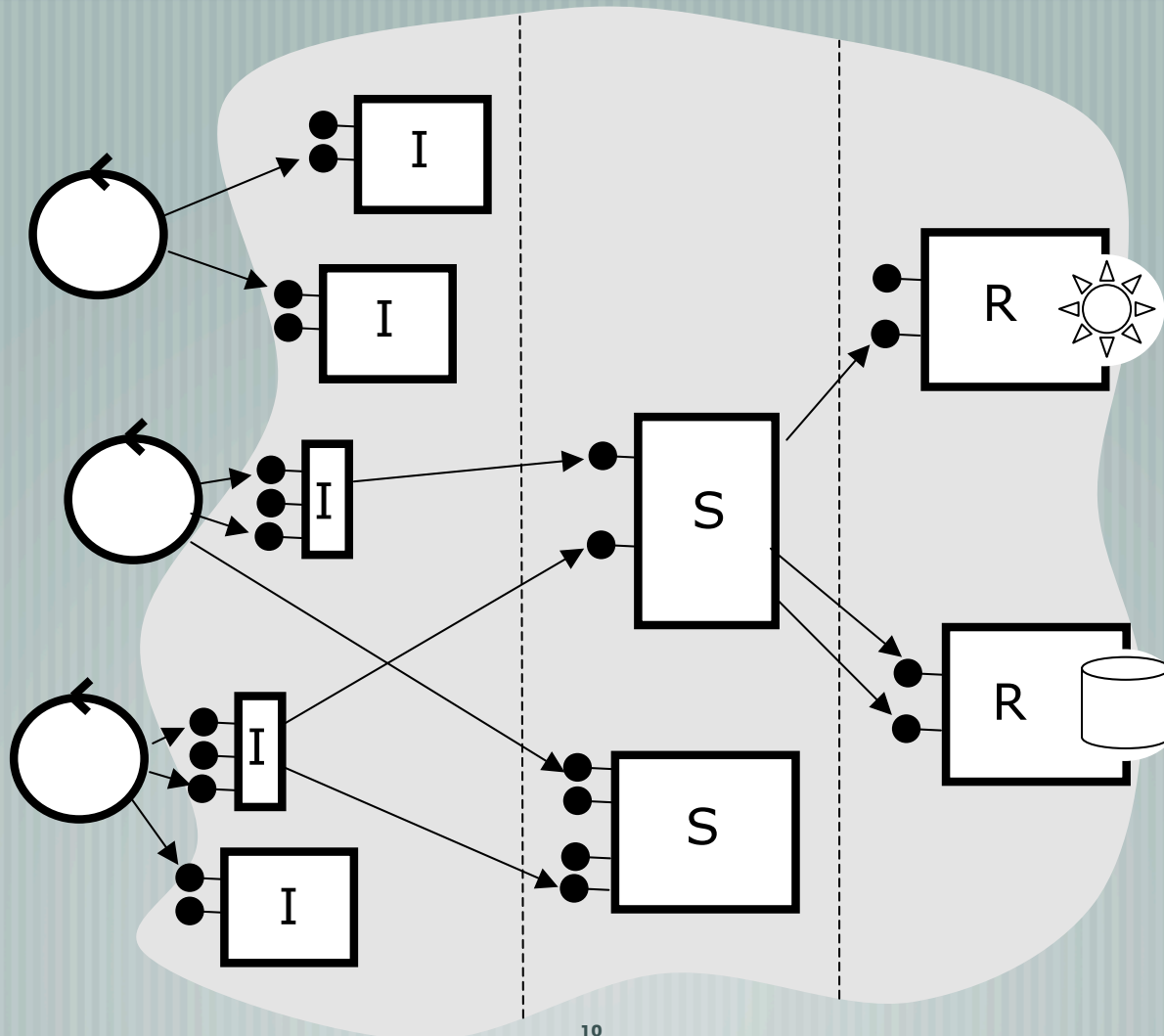
Again, What Happens in a MAS?



- Agents *Speak* with Agents
- Agents *Use* Artefacts
- Artefacts *Link* with Artefacts

- How to Manage Interaction within MAS with Artefacts?

Experiment: Layering the Workspace



A Sort of First-Level Taxonomy

- [**Taxonomy**

- **individual artefacts**

- handling a single agent's interaction

- **social artefacts**

- handling interaction among a number of agents / artefacts

- **resource artefacts**

- handling interaction with resources

Artefacts / Languages for “Organisation”

- [Individual Artefacts seem the Best Place for Ruling Individual Agent Interaction

- on the basis of “organisational” concerns

- [Our Example: Agent Coordination Contexts (ACC)

- filtering any perception / action / communication between the agent and the environment

- [Which Language Here?

- typically declarative – KR-style

- for our “quasi static” perception of organisation

- either process algebra denotation, or FOL rules [RBAC-MAS on AAECC]

- [“Declarative”, however, Does Not Mean Static

- “organisation” may change, agents may reason over (organisation) artefacts and change their state

Artefacts / Languages for “Coordination”

- [Social Artefacts seem the Best Place for Ruling Social Interaction
 - on the basis of objective coordination concerns

- [Our Example: Tuple Centres

- embodying coordination in their (coordinative) behaviour

- [Which Language Here?

- typically operational (event-driven)

- for our “dynamic” perception of coordination

- interaction happens, the artefacts react appropriately

- ReSpecT is FOL, however

- but the semantics is given operationally

- [“Operational”, Too, Does Not Mean Static

- “coordinative behaviour” may change over time, agents may reason over (coordination) artefacts and possibly change their behaviour

Artefacts / Languages for “Security”

— [Resource Artefacts may be a Good Place for Ruling Access to Resource
— on the basis of security concerns

— [Etc. etc.

— OK, fine with this, the picture is clear now.

Hey, Wait!: Behaviour in an Artefact?

— [Yes.

— [But, we already saw that this means in principle that *coordinative* behaviours could instead become *security*-oriented behaviours, or *organisation*-based behaviours.

— [Yes, and everything could be made in assembly language, too.

— [OK. Sorry for asking.

Which Languages Then?

— [Languages to Shape Artefacts

- language to program artefact behaviour, first of all
- and then, languages to use them, and link them

— [Roughly Speaking

- a coordination language à la Linda is a language to operate on artefacts
- Reo is mostly a language to link artefacts
- ReSpecT was born as a language for (programming) artefacts

— [Languages to Model Agent Responsibilities and Actions

- in terms of communications and operations

— [For instance

- RBAC-MAS models responsibilities in terms of agent roles and of an abstract action model
- ACLs typically models only communicative actions, with no regards for physical actions (operations) and responsibility

Our Traditional Example [Cut Down]

Schmidt's Artefacts

- [From CSCW Many Relevant Contributions

- Coordinative vs. Representation Artefacts in an Organisation Workspace [Schmidt 2005]

- Coordinative Artefacts have a Behaviour that is Used to Coordinate and Manage Activities in an Organisation

- Representation Artefacts have a (Possibly Inscribed) State that Records some Partion of the Organisation (Activity)

- [Coordination and Organisation Concerns are Quite Easily Mapped even though quite roughly

- [We May use our ReSpecT Tuple Centres for both as an experiment...

ReSpecT Tuple Centres

— [You Saw Already It Working Last PROMAS (A. Ricci)

— [Just to Say now

— looking at the Theory of Coordination makes it a coordinative artefact

— looking at the Theory of Communication makes it a representation artefact

— [Experiment: An Organisation Tuple Centre for each Workspace

— where the structure is represented declaratively as a FOL Theory

— ACCs released to requesting agents based on the OrgTC's current knowledge

— this is going to work as a basis of our Computational Institutions

[Preliminary] Conclusions

Summarising

- [**New SE Abstractions Open New Spaces for Programming Languages**

- The Agent-Artefact Metamodel for MAS Induces a New View of PLs
- both re-framing old ones, and paving the way for new ones

- [**Artefacts used**

- to shape the MAS environment, and
- to encapsulate organisational/coordination/security responsibilities

- [**Languages for Artefacts are the Key for Managing Interaction in MAS**

- capturing coordination, organisation & security concerns altogether
 - The Reactive Nature of Artefact Behaviours Lead to Event-driven, “Operational” Languages for Artefacts
 - The “Quasi-static” Nature of Organisation Calls for Declarative, KR-style Representations

- [**Need for a General Model for Agent Actions**

- Communication & Operation

Related Talks from Our Group Here

— [Mirko Viroli @ PROMAS/ENV TFG

= Mirko Viroli @ ENV TFG

= Ambra Molesini @ AOSE TFG

= Ambra Molesini @ CEEMAS

— Luca Gardelli @ SELFORG TFG



Spaces for Programming Languages in MASs

A Perspective on Coordination, Organisation & Security

Andrea Omicini

Alma Mater Studiorum–Università di Bologna a Cesena

andrea.omicini@unibo.it

Joint Research with Alessandro Ricci & Mirko Viroli