

# From Organisation Oriented Programming to Multi-Agent Oriented Programming

Olivier Boissier

ISCOD/Henri Fayol Institute & LSTI  
ENS Mines Saint-Etienne - France  
boissier@emse.fr

MATES, Berlin, October 06, 2011



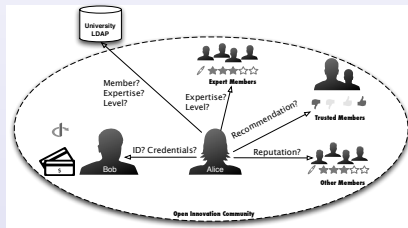
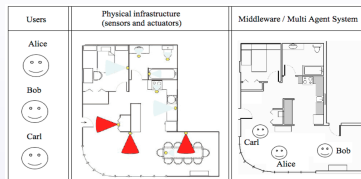
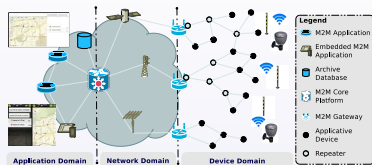
# Acknowledgements

- ▶ **R. H. Bordini**, Federal University of Rio Grande do Sul - Porto Alegre, Brazil (R.Bordini@inf.ufrgs.br)
- ▶ **J. F. Hübner**, Federal University of Santa Catarina - DAS, Florianópolis, Brazil ( jomi@das.ufsc.br)
- ▶ **J.S. Sichman**, Universidade de São Paulo - LTI-PCS, São Paulo, Brazil (jaime.sichman@poli.usp.br)
- ▶ **G. Picard**, ENS Mines St-Etienne, France (gauthier.picard@emse.fr)
- ▶ **M. Hannoun, B. Gâteau, G. Danoy, R. Kitio, C. Persson, R. Yaich**, ENS Mines St-Etienne, France
- ▶ **M. Piunti, A. Santi, A. Ricci**, Università degli studi di Bologna - DEIS, Bologna, Italy (a.ricci@unibo.it)
- ▶ **A. Ciortea, A. Sorici**, Politehnica University of Bucharest, Romania
- ▶ USP-COFECUB Project 98-04, FORTRUST Project ANR 06-10, CMIRA Project Rhône-Alpes Region 2010

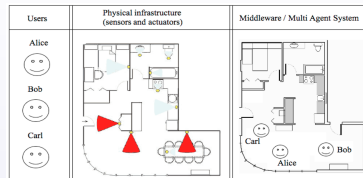
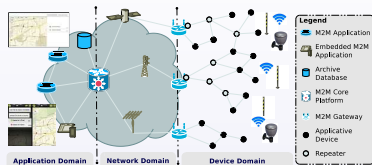
Some of the slides are modified versions of OEOP@EASSS'11



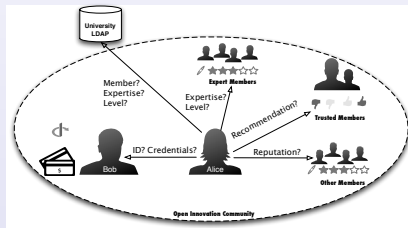
# Context & Motivations



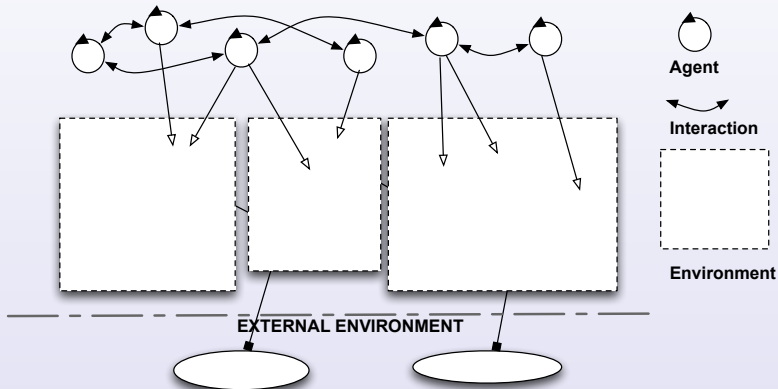
# Context & Motivations



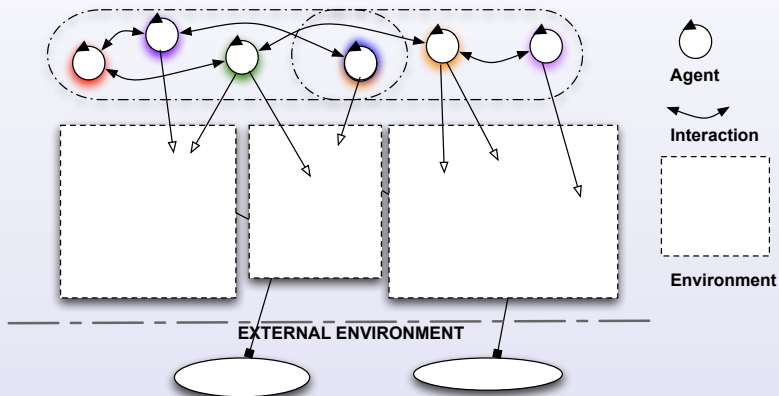
**How to program Open, Decentralized & Distributed Systems  
Operating in Dynamic and Complex Environments  
Taking into account issues such as Flexibility, Trust, ...**



# Agent Centred Approach

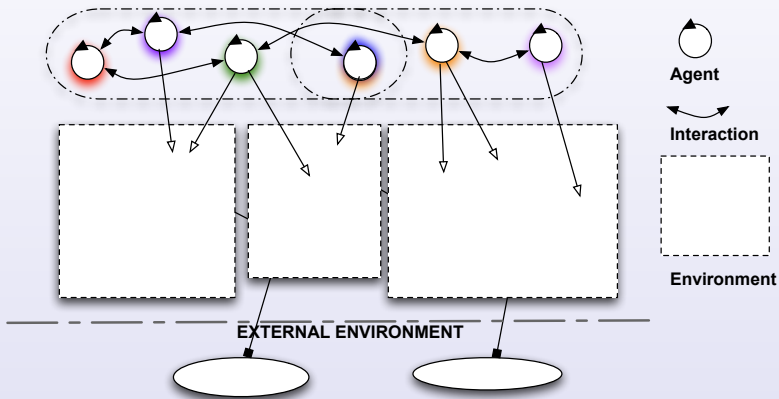


# Agent Centred Approach

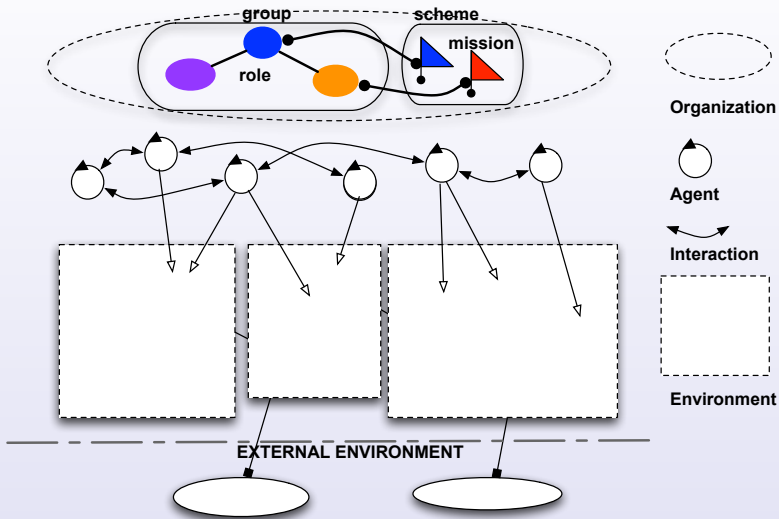


# Agent Centred Approach

Organization is in the "eyes" of the observer / agents

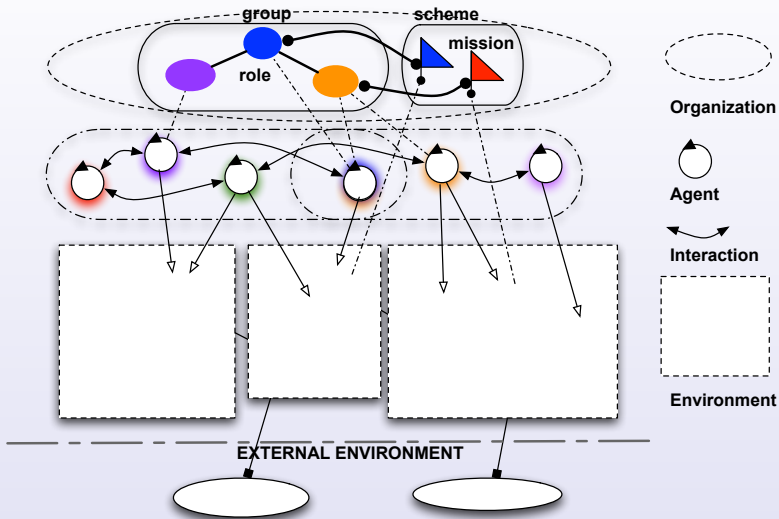


# Organization Centred Approach





# Organization Centred Approach



# Motivations for Organization Centred Approach

- ▶ Applicative motivations:
  - ▶ Increasing integration of human and technological communities (Socio-Technical Systems)
  - ▶ Heterogeneity, Openness, Scalability, Dynamicity, Autonomy are prevailing features
  - ▶ Governance of such systems is a challenge



# Motivations for Organization Centred Approach

- ▶ Applicative motivations:
  - ▶ Increasing integration of human and technological communities (Socio-Technical Systems)
  - ▶ Heterogeneity, Openness, Scalability, Dynamicity, Autonomy are prevailing features
  - ▶ Governance of such systems is a challenge
- ▶ Constitutive and Normative motivations:
  - ▶ To *help* the agents to cooperate with the other agents by defining *common* cooperation schemes
  - ▶ To constrain the agents' behaviour towards the global purposes of the organization, while explicitly addressing the autonomy of the agents within the organization



# Motivations for Organization Centred Approach

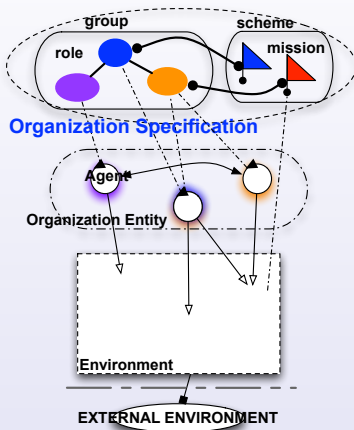
- ▶ **Applicative motivations:**
  - ▶ Increasing integration of human and technological communities (Socio-Technical Systems)
  - ▶ Heterogeneity, Openness, Scalability, Dynamicity, Autonomy are prevailing features
  - ▶ Governance of such systems is a challenge
- ▶ **Constitutive and Normative motivations:**
  - ▶ To *help* the agents to cooperate with the other agents by defining *common* cooperation schemes
  - ▶ To constrain the agents' behaviour towards the global purposes of the organization, while explicitly addressing the autonomy of the agents within the organization
- ▶ **Multiagent motivations:**
  - ▶ Agents need to reason about organization to enter/leave, adapt, obey/disobey the organization
  - ▶ Organization needs to govern agents, accept/refuse agents, accept/refuse modifications, ...



# Outline

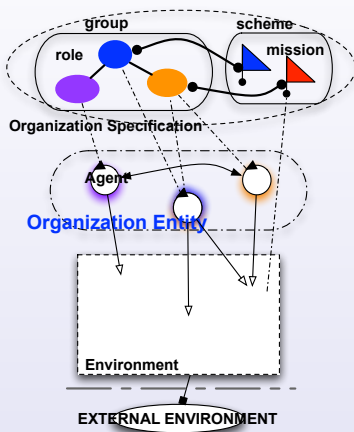
- 1 Introduction
- 2 OOP Perspective: *Moise* Framework
- 3 From OOP to MAOP
- 4 MAOP Perspective: JaCaMo Platform
- 5 Conclusions

# Organization Oriented Programming (OOP)



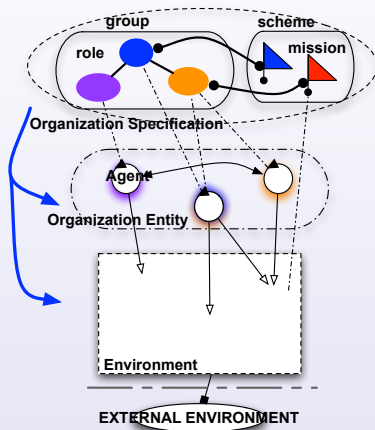
- ▶ Organization is a **first class entity**
  - ▶ Programmed outside the agents
  - ▶ Using organisational concepts to define cooperation patterns
- ▶ Program = **Organization Specification**
  - ▶ partially/totally accessible to the agents, to the environment, to the organization
  - ▶ By changing the specification, we can change the MAS overall behaviour

# Organization Oriented Programming (OOP)



- ▶ Current state of the enacted organization = **Organization entity**
- ▶ Representated in the mental state of the agents  
 ~ possible inconsistencies with the other agents' representations
- ▶ Represented globally/locally in the MAS  
 ~ difficulty to manage and build such a representation in a distributed and decentralized setting

# Organization Oriented Programming (OOP)



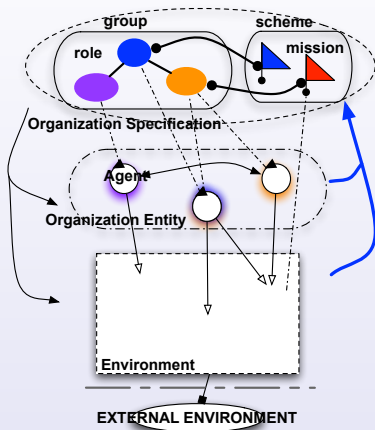
Organization **acts on** the Environment / Agents.

Agents can be:

- ▶ “Organization-Benevolent”
  - ~ Agents execute the program
- ▶ “Organization-Autonomous”
  - ~ Organization enforces the agents to follow the program
  - ~ Organization rewards the agents if they follow the program



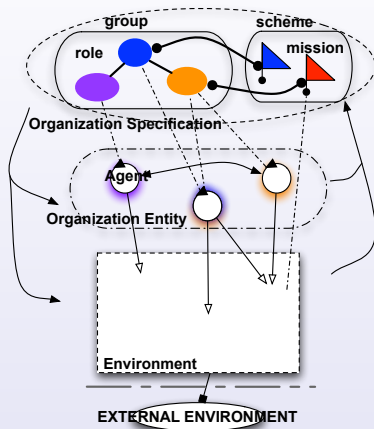
# Organization Oriented Programming (OOP)



Agents or Environment **act on** the Organization:

- ▶ Agents modify the Organization Entity by adopting/leaving roles, creating groups, committing to missions, ...
- ▶ Agents modify the Organization Specification by changing the structure, the cooperation patterns, the norms, ...

# Organization Oriented Programming (OOP)



## OOP Components:

- ▶ Programming language (OML)
- ▶ Platform (OMI)
- ▶ Integration to agent architectures and environment

# OOP Components:

## Organization Modelling Language (OML)

Language for the declarative specification of the organization(s)

- ▶ using multiple dimensions

e.g. structural, functional, dialogic, ...

- ▶ imposing constraints, norms and cooperation patterns on the members of the organization to achieve a global purpose
- ▶ based on an organization model

e.g. AGR [Ferber and Gutknecht, 1998],  
MOISE [Hannoun et al., 2000],  
TEAMCORE [Tambe, 1997],  
ISLANDER [Esteva et al., 2001],  
MOISE<sup>+</sup> [Hübner et al., 2002],  
OPERA [Dignum and Aldewereld, 2010],  
2OPL [Dastani et al., 2009a],  
THOMAS [del Val Noguera et al., 2010], ...



# OOP Components:

## Organization Management Infrastructure (OMI)

### Coordination mechanisms

Support infrastructure for helping the agents to coordinate with each other within the organization.

e.g. MADKIT [Gutknecht and Ferber, 2000], KARMA [Pynadath and Tambe, 2003], ...

### Regulation mechanisms

Governance infrastructure for ensuring the regulation of the agents functioning with respect to the organization norms.

e.g. AMELI [Esteva et al., 2004], S-Moise<sup>+</sup> [Hübner et al., 2006], ORA4MAS [Hübner et al., 2009b], THOMAS [del Val Noguera et al., 2010], ...

### Evolution/Adaptation mechanisms

Reorganization and openness management infrastructure for making the agents able to change their organization, to enter/exit of the organization.

e.g. [Hübner et al., 2004], [Kitio, 2011]



# OOP Components: Integration Mechanisms

## Agent integration mechanisms

They allow agents to be aware of/deliberate on: entering/exiting the organization, modification of the organization, obedience to/violation of norms, sanctioning/rewarding other agents

e.g.  $\mathcal{J}$ -Moise<sup>+</sup> [Hübner et al., 2007], Autonomy based reasoning [Carabelea, 2007], *ProsA<sub>2</sub>* Agent-based reasoning on norms [Ossowski, 1999], ...

## Environment integration mechanisms

They transform organization into embodied organization so that: (i) organization may act on the environment (e.g. enact rules, regimentation), (ii) environment may act on the organization (e.g. count-as rules)

e.g. [Piunti et al., 2009], [Okuyama et al., 2008]



# Some OOP approaches

- ▶ AGR/Madkit [Ferber and Gutknecht, 1998]
- ▶ STEAM/Teamcore [Tambe, 1997]
- ▶ ISLANDER/AMELI [Esteva et al., 2004]
- ▶ Opera/Operetta [Dignum and Aldewereld, 2010]
- ▶ PopOrg [Rocha Costa and Dimuro, 2009]
- ▶ 2OPL [Dastani et al., 2009a]
- ▶ THOMAS [del Val Noguera et al., 2010], ...



# MoISE Framework for OOP

- ▶ OML (language)
  - ▶ Tag-based language  
(issued from *MoISE* [Hannoun et al., 2000],  
*MoISE<sup>+</sup>* [Hübner et al., 2002],  
*MoISEINST* [Gâteau et al., 2005])
- ▶ OMI (infrastructure)
  - ▶ developed as a java-based middleware  
(*S-MoISE<sup>+</sup>* [Hübner et al., 2006] and *SYNAI* [Gâteau et al., 2005])
  - ▶ filters/hides/controls the access by the agents to the  
environment/communication resources
- ▶ Integrations
  - ▶ adhoc integration of Agent and Organization  
(*J-MoISE<sup>+</sup>* [Hübner et al., 2007])



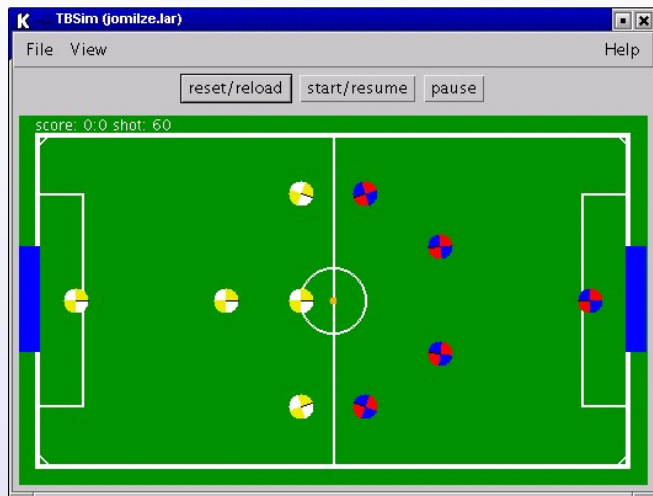
- ▶ Tag-based language for defining
  - ▶ organization specification (OS)
  - ▶ **and** organization entity (OE)
- ▶ Three independent dimensions [Hübner et al., 2007]  
( $\leadsto$  well adapted for reorganization concerns ):
  - ▶ *Structural*: Roles, Groups
  - ▶ *Functional*: Goals, Missions, Schemes
  - ▶ *Normative*: Norms (obligations, permissions, interdictions),  
 $\leadsto$  glue between Structural and Functional Dimensions
- ▶ Abstract description of the organization for
  - ▶ the designers
    - $\leadsto$  [Casare et al., 2010]
  - ▶ the agents
    - $\leadsto$   $\mathcal{J}$ -MoISE [Hübner et al., 2007]
  - ▶ the Organization Management Infrastructure
    - $\leadsto$   $S$ -MoISE<sup>+</sup> [Hübner et al., 2006], ORA4MAS [Hübner et al., 2009b]







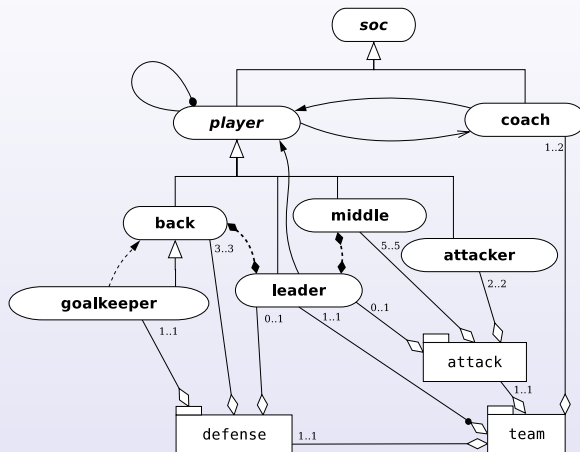
# JOJTeam Example [Hübner et al., 2002]



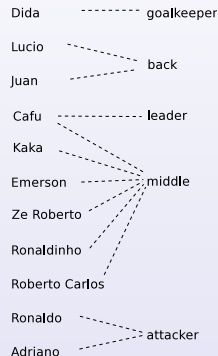
JOJTeam Example - Teambots Simulator

# JOJTeam Example (2/4)

## MOISE OS Structural Specification & OE



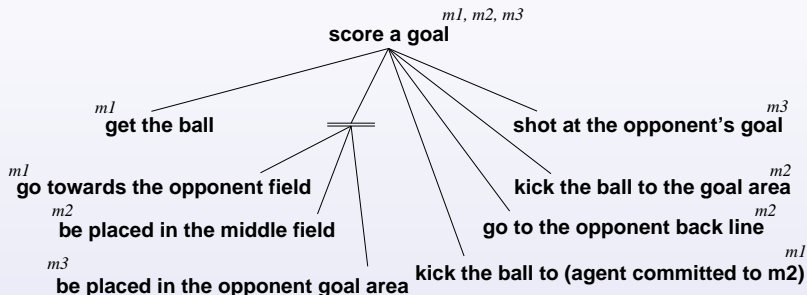
### Organizational Entity



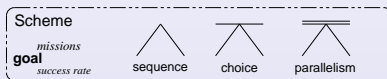
Graphical representation of “3-5-2” structural specification

# JOJTeam Example (3/4)

## Moise OS Functional Specification & OE



### Key



### Organizational Entity

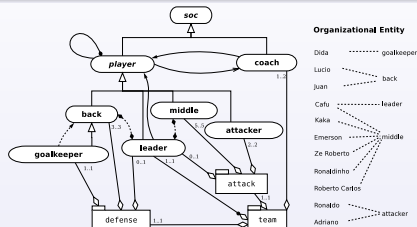


Graphical representation of "side\_attack" social scheme

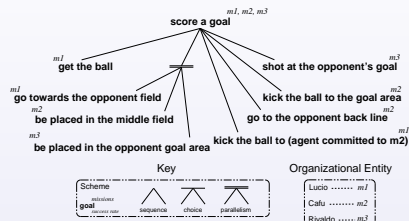


# JOJTeam Example (4/4)

## MOISE OS Normative Specification



Graphical representation of “3-5-2” structural specification



Graphical representation of “side\_attack” social scheme

role	deontic	mission	TTF
back	obliged	m1	get the ball, go ...
left	obliged	m2	be placed at ..., kick ...
right	obliged	m2	1 day
attacker	obliged	m3	kick to the goal, ...
			30 seconds

Partial view of normative specification for JOJTeam

MATES 2011

From OOP to MAOP

# iTV Game Example [Gâteau et al., 2005]



The screenshot displays the iTV Game interface. The top left shows a 3D game scene with a stage and audience. The top right shows four player avatars: Bob (red cap), Ralph (green cap), Jo (blue cap), and Jim (yellow cap). Below the avatars is a large white 'C' with a red arrow pointing down to Bob. The bottom section contains a question and answer area.

**Question 1:**

Quel prénom féminin qui désigne un col rond et plat?

**Réponse:**

1 Claudine	6 Maria
2 Françoise	7 Claudia
3 <b>Magdalena</b>	8 Christine
4 Sophie	9 Germaine
5 Isabelle	

**Score**

**30**

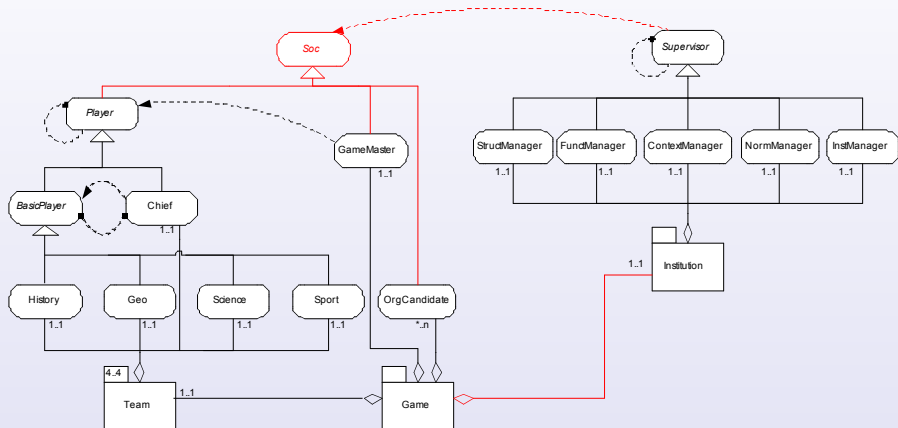
**Ranking**

**128**

Multiagent based iTV Game support infrastructure

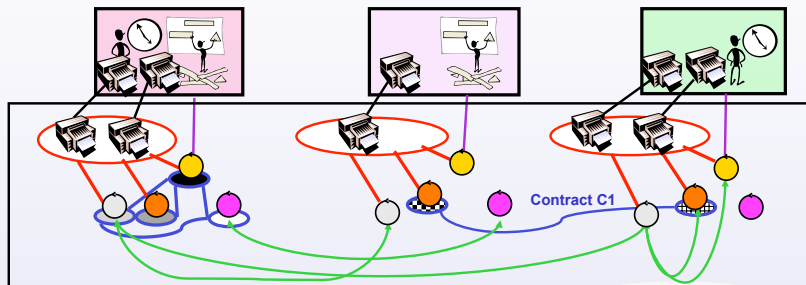
# iTV Game Example (2/2)





## MOISE OS structural specification (Regulation)



Graphical view of the OS Structural Specification

# E-alliance Example [Hübner et al., 2005]



-  Agent for the management of the Alliance
-  Agent for the management of the Contracts on behalf of a Printshop
-  Agent for the management of Negotiations on behalf of a Printshop
-  Agent for the user-interaction with a Printshop

 Environment     Interaction     Organisation

Ecole Nationale  
Supérieure des Mines  
SAINT-ETIENNE

**Alliance**

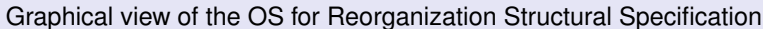
XRCE

réunir pour réussir

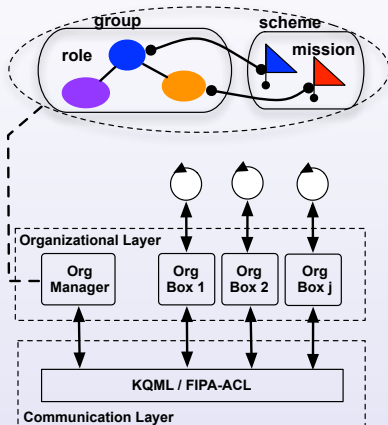
réunir pour réussir

[Castellani 03]  
[Hübner 05]



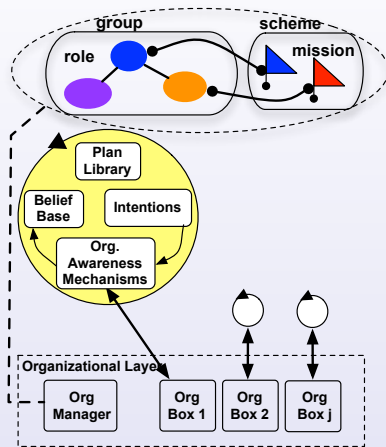


# Moise OMI: *S-Moise*<sup>+</sup> [Hübner et al., 2006]



- ▶ Developed as a java-based middleware
- ▶ OrgBox:
  - ▶ Interface between the agents and the Org Manager, and the Communication layer
- ▶ Org Manager:
  - ▶ Maintains the current state of the OE (adoption of roles, creation of groups, ...)
  - ▶ Ensures the fulfillment of Permissions/Obligations

# Moise Integration: $\mathcal{T}$ -Moise<sup>+</sup> [Hübner et al., 2007]



Dedicated Java library in  
*Jason* [Bordini et al., 2007]

- ~ agents are aware of the Organization by the way of:
  - *organizational events* encapsulated in messages sent by the OrgManager
  - *organizational actions* encapsulated in messages sent by the agents to the OrgManager
- ~ agents can reason on the organization:
  - to achieve *organizational goals*
  - by developing *organizational plans*

# Example

## Example (A new group is created)

```
+group(wpgroup,GId) : true  
  <- jmoise.adopt_role(editor,GId).
```

or

```
+group(wpgroup,GId)[owner(0)] : my_friend(0)  
  <- jmoise.adopt_role(editor,GId).
```

## Example (Some group is deleted)

```
-group(wpgroup,GId) <- .print("Group removed!").
```



# OOP Applications with the *Moise* framework

- ▶ JOJTeam Teambots Simulator [Hübner et al., 2002], [Hübner et al., 2004]
- ▶ Multiagent based iTV Game support infrastructure [Gâteau et al., 2005]
- ▶ A Multi-Agent Approach for Hybrid and Dynamic Coevolutionary Genetic Algorithms : Organizational Model and Real-World Problems Applications [Danoy et al., 2010]
- ▶ Crisis Management Application [Boissier et al., 2011]
- ▶ Smart Home Management Application [Castebrunet et al., 2010]



# Synthesis

Moise Framework is a good support for Organization Centred Approaches, for OOP

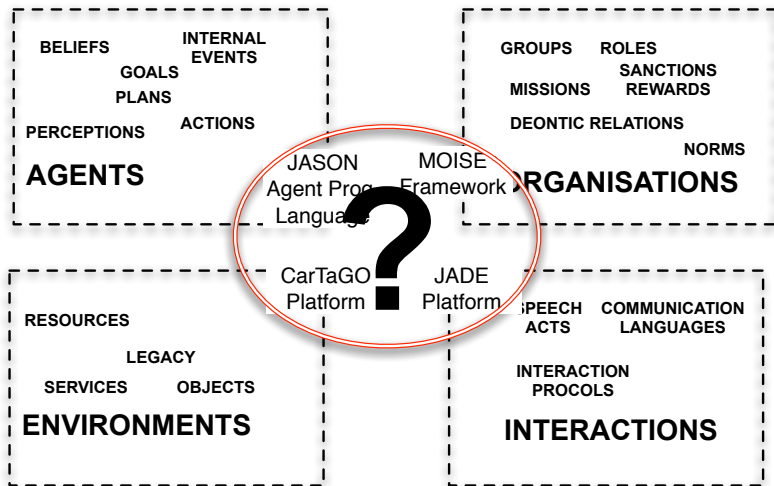
However

- ▶ Organization Management Infrastructure
  - ▶ layer external to the agents
  - ▶ has too much power: it embeds decisions that should be at the agent level
- ▶ Integration Mechanisms
  - ▶ Adhoc integration of the Organization to Agents
  - ▶ Environment as a adhoc entity, Adhoc integration to the Organization
- ▶ Difficulty to combine Agent centred and Organization centred approaches, where global and local levels are in constant interaction, installing a continuous cycle of:
  - ▶ Top-down functioning (normative cooperation patterns imposed on individuals)
  - ▶ Bottom-up functioning (individuals design new cooperation patterns)
- ▶ Lack of suitable high level abstractions for programming applications



- 1 Introduction
- 2 OOP Perspective: *Moise* Framework
- 3 From OOP to MAOP**
- 4 MAOP Perspective: JaCaMo Platform
- 5 Conclusions

# How do we go to MAOP?

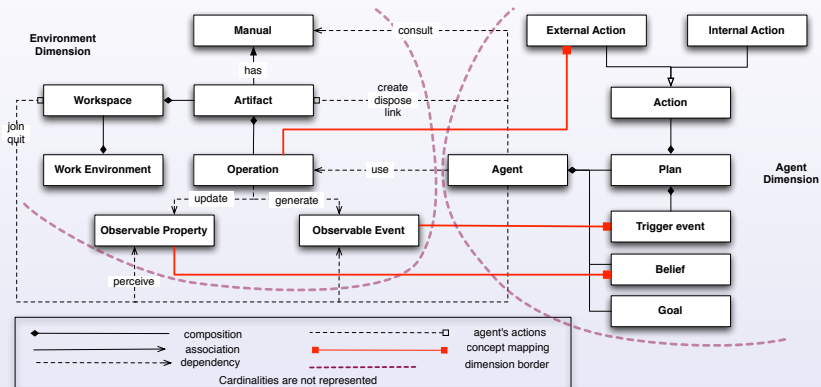


**Which platform, Which Programming Environment?**





# First Step: A+E meta-model



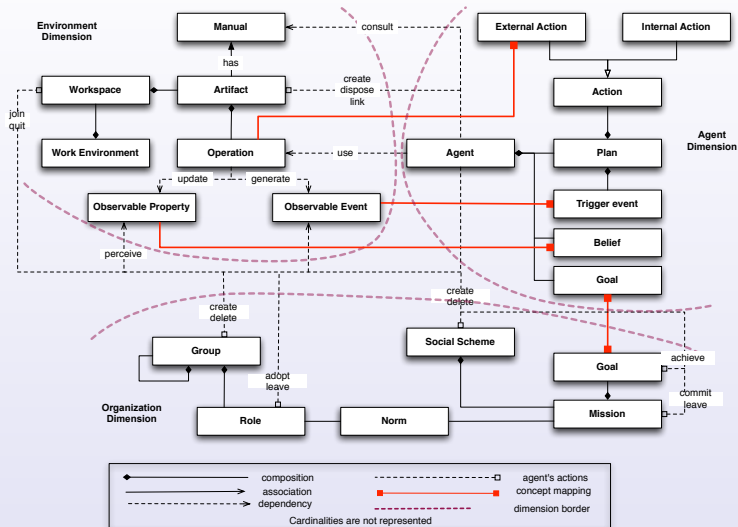
Based on A&A [Omicini et al., 2008], Jason meta-models

# What do we get? (1)

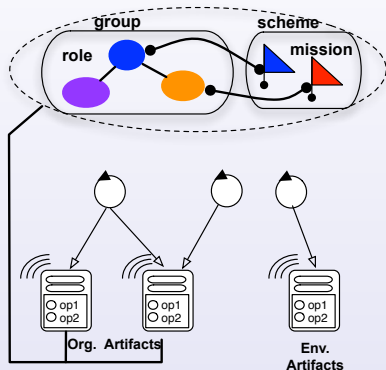
- ▶ Mapping
  - ▶ agent actions into environment operations (e.g. CArtAgO)
  - ▶ environment observable state/events into agent beliefs
- ▶ Outcome
  - ▶ agents with dynamic action repertoire, extended/reshaped by agents themselves
  - ▶ uniformly implementing any mechanisms (e.g. coordination mechanism) in terms of actions/percepts
    - ▶ no need to extend agents with special purpose primitives
  - ▶ exploiting a new type of agent modularity, based on *externalization* [Ricci et al., 2009a]



# Second Step: A+E+O meta-model



# Third Step: Refactoring *Moise* OMI



~> ORA4MAS [Hübner et al., 2009b, Kitio, 2011] Artifact-based working environment

► *organizational artifacts* provide:

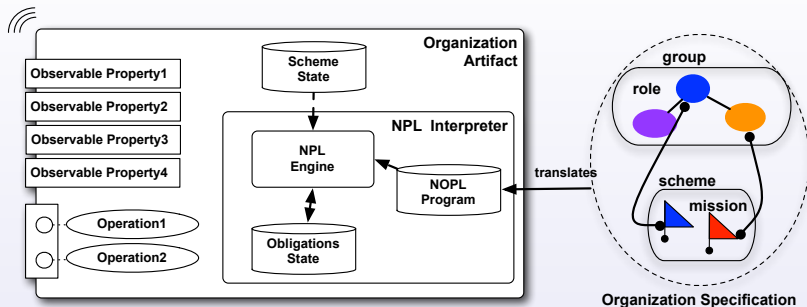
- the access to “institutional” actions
- the access to the visible state of the OE
- synchronization, regimentation and violation detection
- uniform interactions with agents by actions and perceptions

► *agents* have the power back to:

- decide about the organization management
- execute “meta-institutional” actions (e.g. applying sanctions, reorganization, ...)



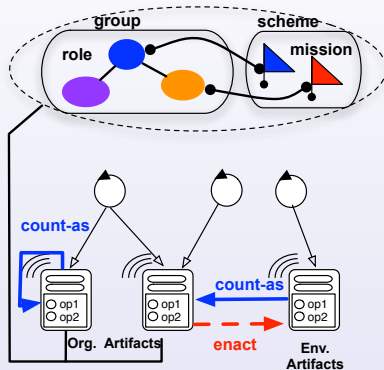
# Organizational Artifact Architecture



- ▶ Interpreter of Normative Programs written in NOPL issued from the automatic translation of the OS written in OML [Hübner et al., 2011]
- ▶ generating *Signals* ( $o$  = obligation(to whom, reason, what, deadline)):
  - ▶  $obl\_created(o)$  – the obligation  $o$  is created – ,  $obl\_fulfilled(o)$  – the obligation  $o$  is fulfilled – ,  $obl\_unfulfilled(o)$  – the obligation  $o$  is unfulfilled – ,  $obl\_inactive(o)$  – the obligation  $o$  is inactive – ,  $norm\_failure(f)$  – the failure  $f$  has happened –



## Fourth Step: E & O Integration



- ▶ Env. Artifacts provide operations on shared resources
- ▶ Org. Artifacts provide organizational operations
- ▶ Both artifacts bound by count-as, enact constitutive rules [Piunti et al., 2009]
- ~ Org-agnostic agents may indirectly act on the organization
- ~ Environment can act on the organization
- ~ Organization is embodied, situated in the environment



# Constitutive rules

## Count-As rule

An event occurring on an artifact, in a particular context, may “count-as” an institutional event

- ▶ transforms the events created in the working environment into activation of an organizational operation
- ↪ indirect automatic updating of the organization

## Enact rule

An event produced on an organizational artifact, in a specific institutional context, may “enact” change and updating of the working environment (i.e., to promote equilibrium, avoid undesirable states)

- ▶ Installing automated control on the working environment
- ▶ Even without the intervention of organizational/staff agents (regimenting actions on physical artifacts, enforcing sanctions, ...)



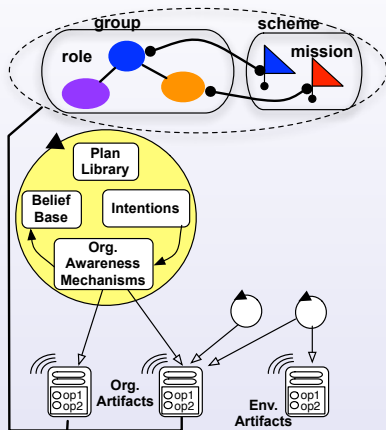
# What do we get? (2)

- ▶ Organisation infrastructure is grounded in the environment
  - ▶ implemented using environment abstractions
  - ▶ ... that agents perceive then as first-class entities of their world
- ▶ Mapping
  - ▶ organisational state reified by the environment computational state
  - ▶ organisational actions/perceptions reified by actions/percepts on the environment state
  - ▶ organisational functionalities encapsulated by suitably designed environment abstractions
  - ▶ environment and organization are connected
- ▶ “The power is back to agents” [Kitio et al., 2008]
- ▶ Extensible set of artifacts
  - ▶ Communication management Artifact [Ciorte, 2011]
  - ▶ Openness Management Artifact [Kitio, 2011]
  - ▶ Reorganisation Artifact [Sorici, 2011]
  - ▶ Evaluation Artifact (kind-of reputation artifact) [Hubner et al., 2009]





# Fifth Step: Refactoring A & O Integration



- ▶ Exploit the uniform access to artifacts

~ agents are aware of the Organization by the way of:

- ▶ *organizational events*
- ▶ *organizational actions*

~ Agents can reason on the organization:

- ▶ to achieve *organizational goals*
- ▶ by developing *organizational plans*



# Example

## Example (Adoption of Role)

...

```
+!discover_art(ToolName)
  <- joinWorkspace("HouseBuildingWsp");
     lookupArtifact(ToolName,ToolId);
     focus(ToolId).

+!contract("SitePreparation",GroupBoardId)
  <- adoptRole(site_prep_contractor)
     focus(GroupBoardId).

+!site_prepared
  <- ... // actions to prepare the site..
```



# What do we get? (3)

- ▶ Normative deliberative agents
  - ▶ possibility to define mechanisms for agents to evolve within an organisation/several organisations
  - ▶ possibility to define proper mechanisms for deliberating on the internalisation/adoption/violation of norms
- ▶ Reorganisation, adaptation of the organisation
  - ▶ possibility to define proper mechanisms for diagnosing/evaluating/refining/defining organisations
- ▶ “Deliberative” Organisations
  - ▶ possibility to define dedicated organisational strategies for the regulation/adaptation of the organisation behaviour (organisational agents)



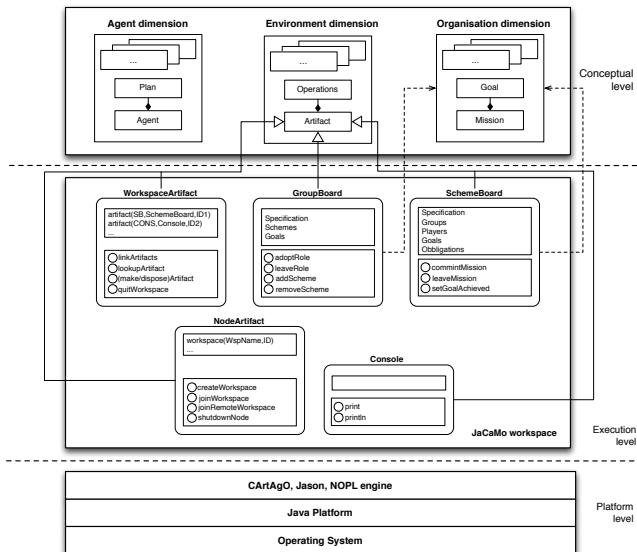
- 1 Introduction
- 2 OOP Perspective: *Moise* Framework
- 3 From OOP to MAOP
- 4 MAOP Perspective: JaCaMo Platform**
- 5 Conclusions

# JaCaMo Platform

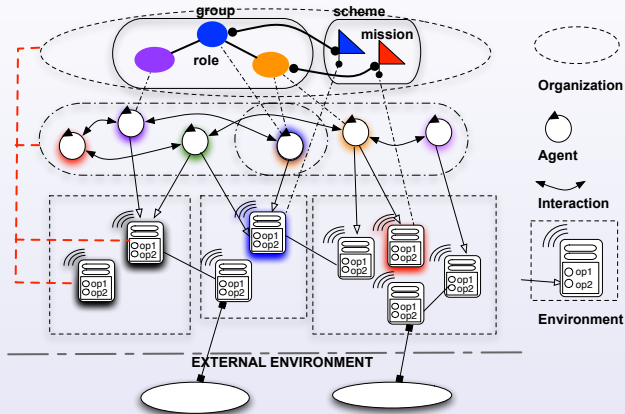
- ▶ Agent: *Jason* agents [Bordini et al., 2007]
- ▶ Environment: CArtAgO platform [Ricci et al., 2009b]
- ▶ Organization: based on an extended/refactored version of the *Moise* Framework
- ▶ Seamless Integration in the JaCaMo platform with dedicated bridges for
  - ▶ A–E (c4Jason, c4Jadex [Ricci et al., 2009b])
  - ▶ E–O (count-as/enact rules [Piunti et al., 2009])
  - ▶ A–O ( $\mathcal{T}$ -*Moise*<sup>+</sup> [Hübner et al., 2007] Action/Perception *Moise*<sup>+</sup> organizational events and actions integration within *Jason*)



# JaCaMo Platform



# JaCaMo Platform



JaCaMo Platform: <http://jacamo.sourceforge.net>

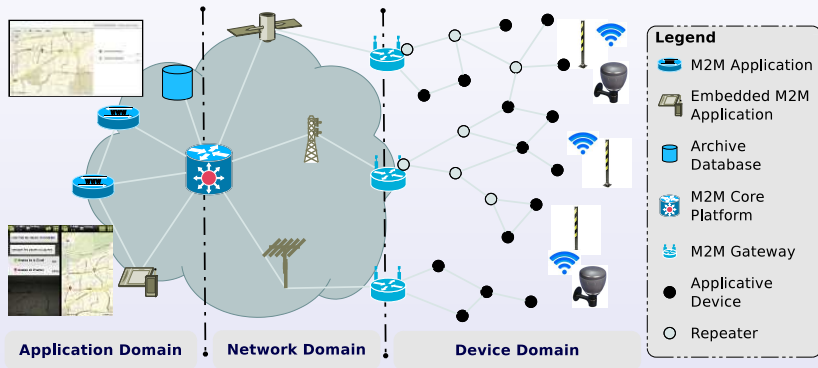
# Ongoing and Related Research

- ▶ Unifying agents, environments and organisation perspectives
  - ▶ Volcano platform [Ricordel and Demazeau, 2002]
  - ▶ MASK platform [Occello et al., 2004]
  - ▶ MASQ [Stratulat et al., 2009], extending AGRE and AGREEN
  - ▶ Embodied organisations [Piunti, 2010]
  - ▶ Situated E-Institutions [Campos et al., 2009]
- ▶ Normative programming and infrastructures [Hübner et al., 2009a, Tinnemeier et al., 2009, Dastani et al., 2009b]

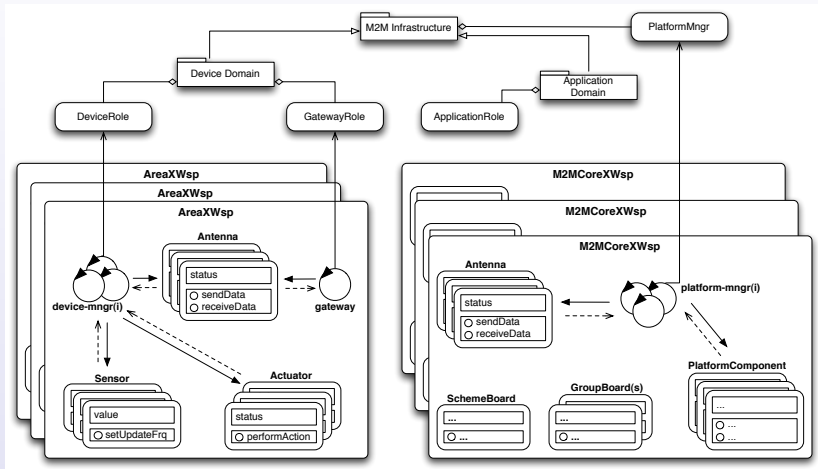




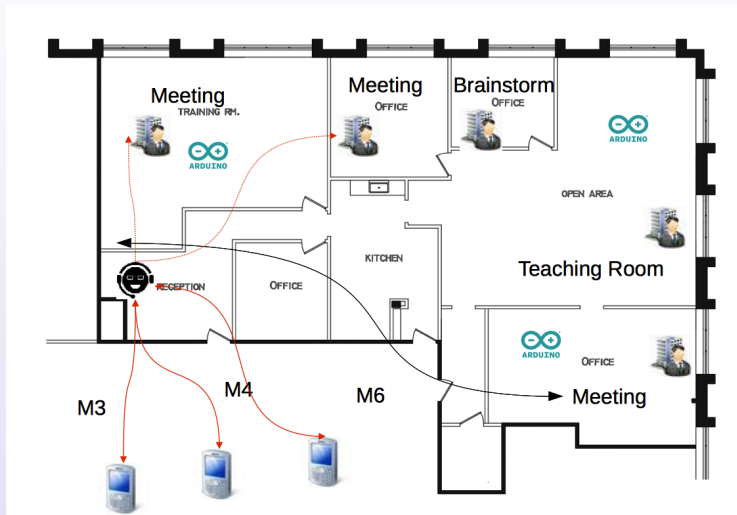
# Application Example with JaCaMo: M2M Agile Governance



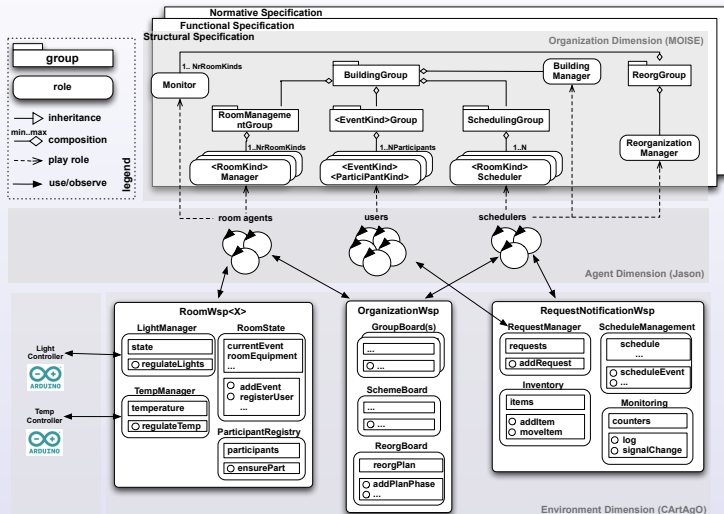
# Application Example with JaCaMo: M2M Agile Governance (2)



# Application Example with JaCaMo: RoomBooking Management



# Application Example with JaCaMo: RoomBooking Management (2)



# Conclusions

- ▶ From a monolithic OOP view to the orthogonal integration of AOP & EOP & OOP dimensions
  - ↪ improvement of *separation of concerns*
    - ▶ using the best abstraction level and tools to tackle the specific dimensions, avoiding design pitfalls, such as using agents to implement either non-autonomous entities (e.g., a blackboard agent) or a collection of autonomous entities (group agent)
  - ↪ promoting openness and heterogeneity
    - ▶ E.g., heterogeneous agents working in the same organisation, heterogeneous agents working in the same environment, the same agent working in different and heterogeneous organisations, the same agent working in different heterogeneous environments
- ↪ Outcome from a programming point of view
  - ▶ code more clean and understandable
  - ▶ improving modularity, extensibility, reusability



# Open Issues

- ▶ AOSE Methodology for MAOP, Integrating such a methodology with the JaCaMo platform
- ▶ Introducing heterogeneity in the platform: multiple different Agent Architectures, different Environment Models, different Organization Models
- ▶ Introducing the Interaction Dimension in the JaCaMo platform
- ▶ Going back to Organizations:
  - ▶ Bottom-up AND Top-Down functioning within organizations  $\equiv$  Emergence AND Normative functioning
  - ▶ Management of Multiple Organizations, of Open Organizations
  - ▶ Scalability, Robustness
  - ▶ Hybrid Systems / Socio-Technical Systems
  - ▶ ...



# Bibliography I



Boissier, O., Balbo, F., and Badeig, F. (2011).

Controlling multi-party interaction within normative multi-agent organizations.



Bordini, R. H., Hübner, J. F., and Wooldrige, M. (2007).

*Programming Multi-Agent Systems in AgentSpeak using Jason.*

Wiley Series in Agent Technology. John Wiley & Sons.



Campos, J., López-Sánchez, M., Rodriguez-Aguilar, J. A., and Esteva, M. (2009).

Formalising situatedness and adaptation in electronic institutions.

In *Coordination, Organizations, Institutions and Norms in Agent Systems IV*, volume 5428/2009 of LNCS. Springer Berlin / Heidelberg.



Carabelea, C. (2007).

*Reasoning about autonomy in open multi-agent systems - an approach based on the social power theory.*

in french, ENS Mines Saint-Etienne.



Casare, S. J., Brandão, A. A. F., and Sichman, J. S. (2010).

A semiotic perspective for multiagent systems development.

In AAMAS, pages 1373–1374.



# Bibliography II



Castebrunet, M., Boissier, O., Giroux, S., and Rialle, V. (2010).

Organization nesting in a multi-agent application for ambient intelligence.

In Demazeau, Y. and Dignum, F., editors, *Proceedings of the 8th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS'2010)*, Advances in Intelligent and Soft-Computing. Springer.



Ciortea, A. (2011).

Modeling relationships for privacy preservation in virtual communities.

Master's thesis, University Politehnica of Bucharest.



Danoy, G., Bouvry, P., and Boissier, O. (2010).

A multi-agent organizational framework for coevolutionary optimization.



Dastani, M., Grossi, D., Meyer, J.-J., and Tinnemeier, N. (2009a).

Normative multi-agent programs and their logics.

In Meyer, J.-J. and Broersen, J., editors, *Knowledge Representation for Agents and Multi-Agent Systems*, volume 5605 of *Lecture Notes in Computer Science*, pages 16–31. Springer Berlin / Heidelberg.





# Bibliography III



Dastani, M., Tinnemeier, N., and Meyer, J.-J. C. (2009b).

A programming language for normative multi-agent systems.

In *Multi-Agent Systems: Semantics and Dynamics of Organizational Models*.  
IGI-Global.



del Val Noguera, E., Criado, N., Carrascosa, C., Julián, V., Rebollo, M., Argente, E.,  
and Botti, V. J. (2010).

Thomas: a service-oriented framework for virtual organizations.

In van der Hoek, W., Kaminka, G. A., Lespérance, Y., Luck, M., and Sen, S., editors,  
*AAMAS*, pages 1631–1632. IFAAMAS.



Dignum, V. and Aldewereld, H. (2010).

Operetta: Organization-oriented development environment.

In *Proceedings of LADS @ MALLOW 2010*, pages 14–20.



Esteva, M., Rodriguez-Aguilar, J. A., Sierra, C., Garcia, P., and Arcos, J. L. (2001).

On the formal specification of electronic institutions.

In Dignum, F. and Sierra, C., editors, *Proceedings of the Agent-mediated Electronic  
Commerce*, LNAI 1191, pages 126–147, Berlin. Springer.



# Bibliography IV



Esteva, M., Rodríguez-Aguilar, J. A., Rosell, B., and Arcos, J. L. (2004).

AMELI: An agent-based middleware for electronic institutions.

In Jennings, N. R., Sierra, C., Sonenberg, L., and Tambe, M., editors, *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'2004)*, pages 236–243, New York. ACM.



Ferber, J. and Gutknecht, O. (1998).

A meta-model for the analysis and design of organizations in multi-agents systems.

In Demazeau, Y., editor, *Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS'98)*, pages 128–135. IEEE Press.



Gâteau, B., Boissier, O., Khadraoui, D., and Dubois, E. (2005).

Moiseinst: An organizational model for specifying rights and duties of autonomous agents.

In *Third European Workshop on Multi-Agent Systems (EUMAS 2005)*, pages 484–485, Brussels Belgium.



Gutknecht, O. and Ferber, J. (2000).

The MadKit agent platform architecture.

In *Agents Workshop on Infrastructure for Multi-Agent Systems*, pages 48–55.



# Bibliography V



Hannoun, M., Boissier, O., Sichman, J. S., and Sayettat, C. (2000).

Moise: An organizational model for multi-agent systems.

In Monard, M. C. and Sichman, J. S., editors, *Proceedings of the International Joint Conference, 7th Ibero-American Conference on AI, 15th Brazilian Symposium on AI (IBERAMIA/SBIA'2000), Atibaia, SP, Brazil, November 2000*, LNAI 1952, pages 152–161, Berlin. Springer.



Hübner, J., Boissier, O., and Bordini, R. (2011).

A normative programming language for multi-agent organisations.

*Annals of Mathematics and Artificial Intelligence*, pages 1–27.  
10.1007/s10472-011-9251-0.



Hübner, J. F., Boissier, O., and Bordini, R. H. (2009a).

Normative programming for organisation management infrastructures.

In *MALLOW Workshop on Coordination, Organization, Institutions and Norms in Agent Systems in Online Communities (COIN-MALLOW 2009)*.



Hübner, J. F., Boissier, O., Kitio, R., and Ricci, A. (2009b).

Instrumenting Multi-Agent Organisations with Organisational Artifacts and Agents.

*Journal of Autonomous Agents and Multi-Agent Systems*.



# Bibliography VI



Hübner, J. F., Boissier, O., and Sichman, J. S. (2005).

Using a multi-agent organization description language to describe contract dynamics in virtual enterprises.

In Skowron, A., Barthès, J.-P. A., Jain, L. C., Sun, R., Morizet-Mahoudeaux, P., Liu, J., and Zhong, N., editors, *IAT*, pages 672–678. IEEE Computer Society.



Hübner, J. F., Sichman, J. S., and Boissier, O. (2002).

A model for the structural, functional, and deontic specification of organizations in multiagent systems.

In Bittencourt, G. and Ramalho, G. L., editors, *Proceedings of the 16th Brazilian Symposium on Artificial Intelligence (SBIA'02)*, volume 2507 of *LNAI*, pages 118–128, Berlin. Springer.



Hübner, J. F., Sichman, J. S., and Boissier, O. (2004).

Using the *MOISE*<sup>+</sup> for a cooperative framework of MAS reorganisation.  
pages 506–515.

# Bibliography VII



Hübner, J. F., Sichman, J. S., and Boissier, O. (2006).

S-MOISE+: A middleware for developing organised multi-agent systems.

In Boissier, O., Dignum, V., Matson, E., and Sichman, J. S., editors, *Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems*, volume 3913 of *LNCS*, pages 64–78. Springer.



Hübner, J. F., Sichman, J. S., and Boissier, O. (2007).

Developing Organised Multi-Agent Systems Using the MOISE+ Model: Programming Issues at the System and Agent Levels.

*Agent-Oriented Software Engineering*, 1(3/4):370–395.



Hubner, J. F., Vercouter, L., and Boissier, O. (2009).

Instrumenting Multi-Agent Organisations with reputation artifacts.

In Hubner, J. F., Matson, E., Boissier, O., and Dignum, V., editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems IV*, volume 5428 of *LNAI*, pages 96–110. Springer.



Kitio, R. (2011).

*Gestion de l'ouverture au sein d'organisations multi-agents. Une approche basée sur des artefacts organisationnels.*

PhD thesis, ENS Mines Saint-Etienne.



# Bibliography VIII



Kitio, R., Boissier, O., Hübner, J. F., and Ricci, A. (2008).

Organisational artifacts and agents for open multi-agent organisations: “giving the power back to the agents”.

In *Coordination, Organizations, Institutions, and Norms in Agent Systems III*, volume 4870 of *LNCS*, pages 171–186. Springer.



Occello, M., Baeijs, C., Demazeau, Y., and Koning, J.-L. (2004).

MASK: An AEIO toolbox to design and build multi-agent systems.

In et al., C., editor, *Knowledge Engineering and Agent Technology*, IOS Series on Frontiers in AI and Applications. IOS press, Amsterdam.



Okuyama, F. Y., Bordini, R. H., and da Rocha Costa, A. C. (2008).

A distributed normative infrastructure for situated multi-agent organisations.

In *DALT*, pages 29–46.



Omicini, A., Ricci, A., and Viroli, M. (2008).

Artifacts in the A&A meta-model for multi-agent systems.

*Autonomous Agents and Multi-Agent Systems*, 17(3):432–456.

# Bibliography IX



Ossowski, S. (1999).

*Co-ordination in Artificial Agent Societies: Social Structures and Its Implications for Autonomous Problem-Solving Agents*, volume 1535 of LNAI.

Springer.



Piunti, M. (2010).

Embodied organizations: a unifying perspective in programming agents, organizations and environments.

PhD thesis.



Piunti, M., Ricci, A., Boissier, O., and Hubner, J. (2009).

Embodying organisations in multi-agent work environments.

In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT 2009)*, Milan, Italy.



Pynadath, D. V. and Tambe, M. (2003).

An automated teamwork infrastructure for heterogeneous software agents and humans.

*Autonomous Agents and Multi-Agent Systems*, 7(1-2):71–100.



# Bibliography X



Ricci, A., Piunti, M., and Viroli, M. (2009a).

Externalisation and internalization: A new perspective on agent modularisation in multi-agent system programming.

In Dastani, M., Fallah-Seghrouchni, A. E., Leite, J., and Torroni, P., editors, *LADS*, volume 6039 of *Lecture Notes in Computer Science*, pages 35–54. Springer.



Ricci, A., Piunti, M., Viroli, M., and Omicini, A. (2009b).

Environment programming in CArtaGO.

In *Multi-Agent Programming: Languages, Platforms and Applications, Vol.2*. Springer.



Ricordel, P. and Demazeau, Y. (2002).

VOLCANO: a vowels-oriented multi-agent platform.

In Dunin-Keplicz and Nawarecki, editors, *Proceedings of the International Conference of Central Eastern Europe on Multi-Agent Systems (CEEMAS'01)*, volume 2296 of *LNAI*, pages 252–262. Springer Verlag.



Rocha Costa, A. C. d. and Dimuro, G. (2009).

A minimal dynamical organization model.

In Dignum, V., editor, *Multi-Agent Systems: Semantics and Dynamics of Organizational Models*, chapter XVII, pages 419–445. IGI Global.



# Bibliography XI



Sorici, A. (2011).

Agile governance in an ambient intelligence environment.

Master's thesis, University Politehnica of Bucharest.



Stratulat, T., Ferber, J., and Tranier, J. (2009).

MASQ: towards an integral approach to interaction.

In *AAMAS (2)*, pages 813–820.



Tambe, M. (1997).

Towards flexible teamwork.

*Journal of Artificial Intelligence Research*, 7:83–124.



Tinnemeier, N., Dastani, M., Meyer, J.-J., and van der Torre, L. (2009).

Programming normative artifacts with declarative obligations and prohibitions.

In *IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT 2009)*.

