

# Organisational Artifacts and Agents For Open Multi-Agent Organisations

“Giving the power back to the agents”

Rosine Kitio<sup>1</sup>    Olivier Boissier<sup>1</sup>  
Jomi F. Hübner<sup>1,2</sup>    Alessandro Ricci<sup>3</sup>

<sup>1</sup>ENSM.SE, France & <sup>2</sup>FURB, Brazil & <sup>3</sup>DEIS, Italy

EUMAS 2007

# Outline

- Introduction
- ORA4MAS Infrastructure
- Shaping ORA4MAS on  $\mathcal{MOISE}^+$
- Conclusion and perspectives

# MAS Organisation

## and its importance for MAS

- A multiagent system has two properties which seem controversial:

- a **global** purpose  $\times$  **autonomous** agents

While the autonomy of the agents is essential for the MAS, it may cause the looseness of the global congruence

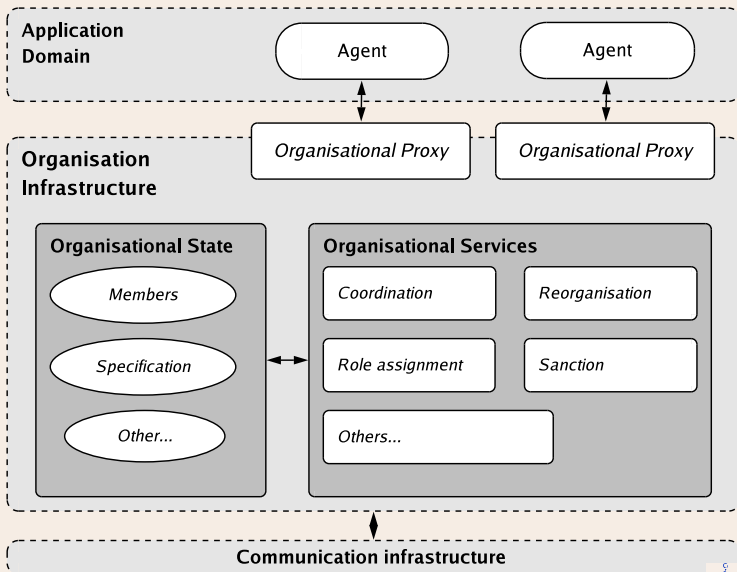
- The **organisation** of an MAS is used to solve this conflict constraining the agents' behaviour towards global purposes
- Example: when an agent adopts the student role, he indeed adopts a set of behavioural constraints that helps the achievement of the global purpose of the school

# Open MAS

## and the organisation

- New (heterogeneous) agents should be able to
  - enter and leave a running MAS
  - know the organisation – **system level**, e.g. the system has a service that provides a description of the organisation
  - reason about it – **agent level**
- New agents have to follow some organisational rules – **organisational enforcement** can be implemented
  - by the agents
  - by the system

# Current approaches – enforcement at the system level



# Current approaches

and some problems — organisational enforcement only at system level

- ① Organisational services are implemented as “special” agents — which are **conceptually different** — agents doing services
- ② Organisational decisions are taken in the services layer — the **organisation middleware has too much power**
  - For example, if some agent performs a forbidden action, the middleware **detects** it as a violation and **decides** to apply a sanction (or even disable the execution of the action)

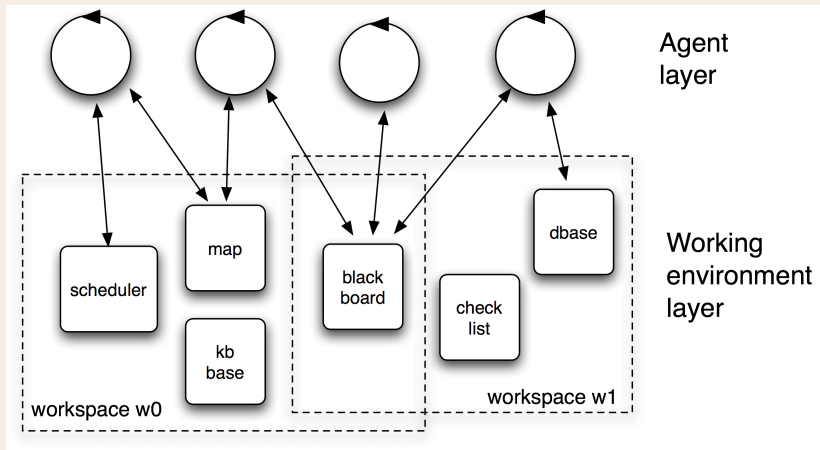
services taken decisions which are “closed” for the agents

- ③ Organisational proxies are different for each implementation (the agent should be developed for a specific proxy)

# Objective

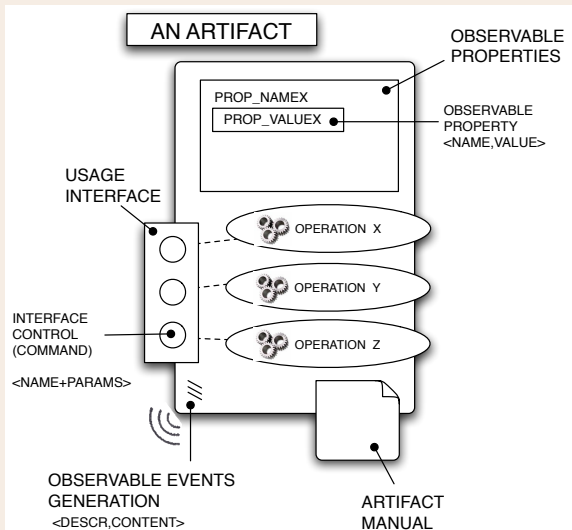
- Provide a conceptual and architectural step towards the simplification of the development of open and organised MAS
- Proposal: ORA4MAS
  - Approach based on the simple concept of artifacts
  - While agents model the **decisions** of the system, the artifacts model its **functions**

# Agents & Artifacts





# Artifacts

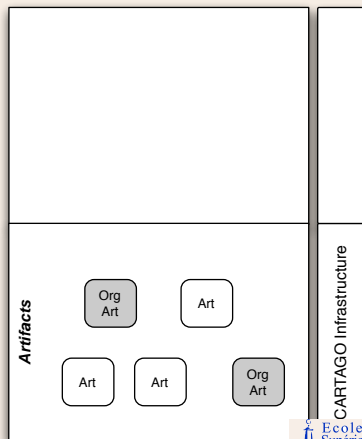


# ORA4MAS

To engineer the organisational infrastructure in terms of organisational artifacts and agents

## Organisational Artifacts

- they are discovered and used by the agents to work inside the organisation
- encapsulate organisational functionalities, that agents of an organisation may want/have to use in order to participate in organisational activities

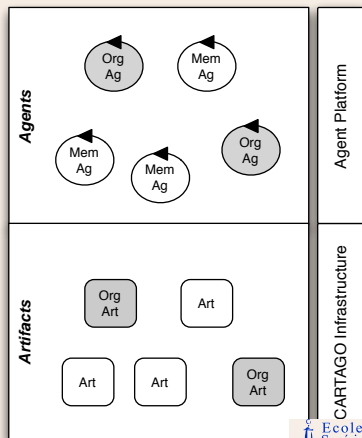


# ORA4MAS

To engineer the organisational infrastructure in terms of organisational artifacts and agents

## Organisational Agents

- they dynamically manage and adapt the organisation by creating and linking artifacts
- responsible of management of activities inside the organisation (observing, monitoring, reasoning and deciding about organisation)



# Shaping ORA4MAS artifacts upon $\mathcal{MOISE}^+$

- $\mathcal{MOISE}^+$  will be used as a first case study for the ORA4MAS approach
- It is an organisational model that allows us to specify three dimensions of the organisation
  - structural specification (roles and groups)
  - functional specification (goals, missions, schemes)
  - deontic specification (obligations and permissions)

# Organisational Artifacts for MOISE<sup>+</sup>

## ORG-BOARD

org-spec

current-groups

current-schemes

- ☐ enterOrg
- ☐ leaveOrg
- ☐ registerGroup( $G, GB$ )
- ☐ removeGroup( $G$ )
- ☐ registerScheme( $S, SB$ )
- ☐ removeScheme( $S$ )

## GROUP-BOARD

role-assignments

- ☐ adoptRole( $R$ )
- ☐ giveUpRole( $R$ )
- ☐ sendMsg( $A, M$ )

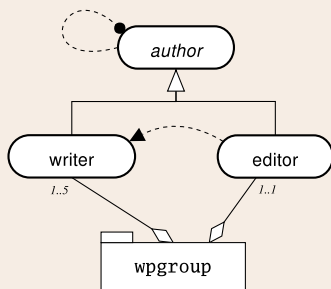
## SCHEME-BOARD

scheme-state

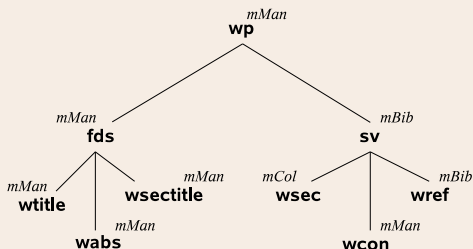
missions-committed

- ☐ commitMission( $M$ )
- ☐ setMissionState( $M, S$ )
- ☐ setGoalState( $G, S$ )

# Example: “writing a paper” — specification



(a) Structural Specification

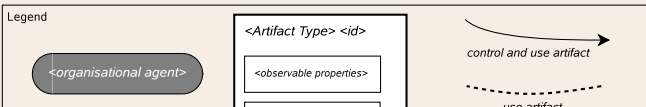
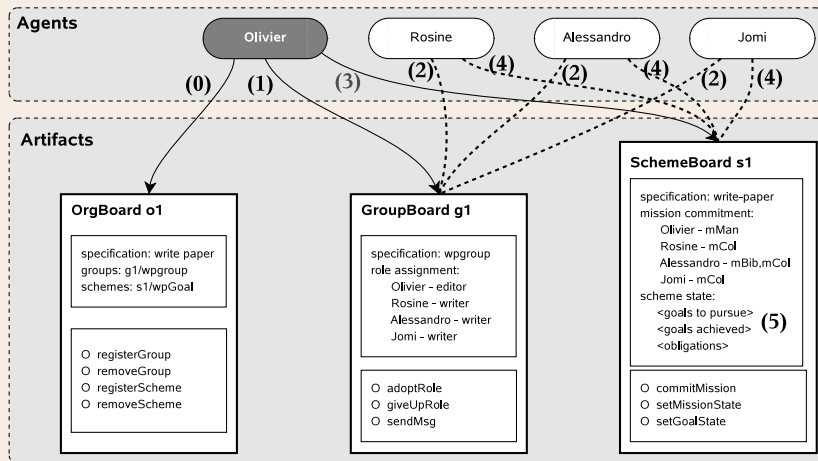


(b) Functional Specification -  
“write-paper”

editor	<i>permission</i>	$mMan$
writer	<i>obligation</i>	$mCol$
writer	<i>obligation</i>	$mBib$

(c) Deontic Specification

# Example: “writing a paper” — dynamics



# ORA4MAS platform

This approach is being implemented using

- CARTAGO for the artifacts infrastructure (provides distribution of the artifact in a network)
- $\mathcal{S}\text{-}\mathcal{MOISE}^+$  for the artifact operations (provides algorithms for the operations of the artifacts)
- $\mathcal{J}\text{-}\mathcal{MOISE}^+$  for agent's programming (using the CARTAGO - **Jason** integration to provides artifact access to **Jason** agents)



# Conclusion – “Power back to agents”

- Encapsulation of organisational **functions** in artifacts
- Organisational **decisions** are given to the agents
- Decentralised management (several agents and artifacts)
- Organisational enforcement is done by agents (based on particular domain requirements)
- Openness
  - Manuals can be read for new coming agents
  - The organisational decisions are open to agents

# Perspectives

- A generic solution
  - Apply ORA4MAS to other organisational models (ISLANDER, OMNI,  $\mathcal{M}\text{OISE}^{Inst}$ , AGR, ...)
- An adaptative solution
  - Study the reorganisation process of an MAS with organisational artifacts
- Control the access to organisational artifacts
  - Define a meta-organisation for the ORA4MAS, having special roles for organisational agents that give them access to the organisational artifacts