Organisation Oriented Programming with $\mathcal{M}$OISE$^+$

at the system and agent levels

Jomi F. Hübner
(collaboration with Olivier Boissier and Jaime S. Sichman)

ENS Mines Saint Etienne, France
hubner@emse.fr

LIP6 Seminars

Ecole Nationale Supérieure des Mines SAINT-ETIENNE

Outline

Reasons for organisation in MAS
'normative view'

- Multi-agent systems have two properties which seem contradictory:
  - a global purpose $\times$ autonomous agents

  While the autonomy of the agents is essential for MASs, it may cause loss in the global coherence of the system
- The organisation of a MAS is used to solve this problem constraining the agents' behaviour towards global purposes
- For example, when an agent adopts a role, it adopts a set of behavioural constraints that support a global purpose

Ecole Nationale Supérieure des Mines SAINT-ETIENNE

Constraining the agents' autonomy by **Norms** mechanisms

- Regimented norms: the organisation prevents their violation by the agents
  - e.g. messages that do not follow the protocol are discarded
- Enforced norms: agents decide to obey or not to them, the organisation lets the agents the possibility to violate them
  - e.g. a master thesis should be written in two years

$\leadsto$ Detection of violations, decision about sanctions

Reasons for organisation in MAS
'constitutive view'

- The organisation helps the agents to cooperate by defining common
  - global tasks
  - protocols
- For example, 'to bid' for a product on eBay is an institutional action only possible because the eBay defines the rules for that very action
  - the bid protocol is a constraint but it also creates the action

Programming organised MAS

- System approach:
  - Develop an organisational infrastructure that helps the agents to participate in the organisation
  - Develop an organisational infrastructure that ensures or enforce that the organisational norms will be followed
    - The agents have to respect the organisation despite their architecture
- Agent-centred approach:
  - Develop agent reasoning mechanisms that are aware of the organisation
  - Not suitable for all kinds of open systems (unknown agents may not behave well!)

▸ writing paper example

Ecole Nationale Supérieure des Mines SAINT-ETIENNE

Ecole Nationale Supérieure des Mines SAINT-ETIENNE

$\mathcal{M}$OISE$^+$– general view

- Organisation Modelling Language (OML)
  - ⇝ allows the designer to specify the organisation of a MAS along three dimensions (structural, functional, deontic)
- Organisational Infrastructure
  - ⇝ interprets the OML and then constraints/supports the agents in the specified organisation
    - by means of regimentation, enforcement, tools for cooperative tasks, ...
    - allows agents to interact with the organisation (agent programming issues)
- Support for agent programming

$\mathcal{M}$OISE$^+$– general view

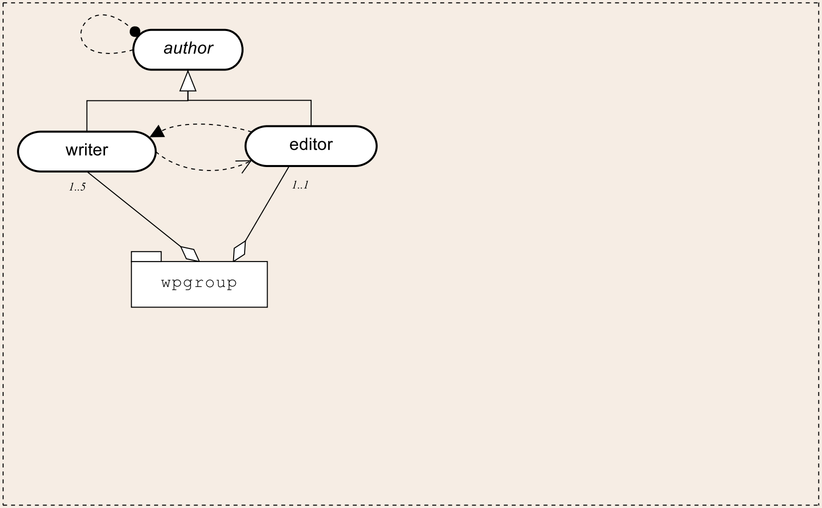
- Organisation Modelling Language (OML)
  - ⇝ allows the designer to specify the organisation of a MAS along three dimensions (structural, functional, deontic)
- Organisational Infrastructure
  - ⇝ interprets the OML and then constraints/supports the agents in the specified organisation
    - by means of regimentation, enforcement, tools for cooperative tasks, ...
    - allows agents to interact with the organisation (agent programming issues)
- Support for agent programming

Ecole Nationale Supérieure des Mines SAINT-ETIENNE

$\mathcal{M}$OISE^{+}– general view

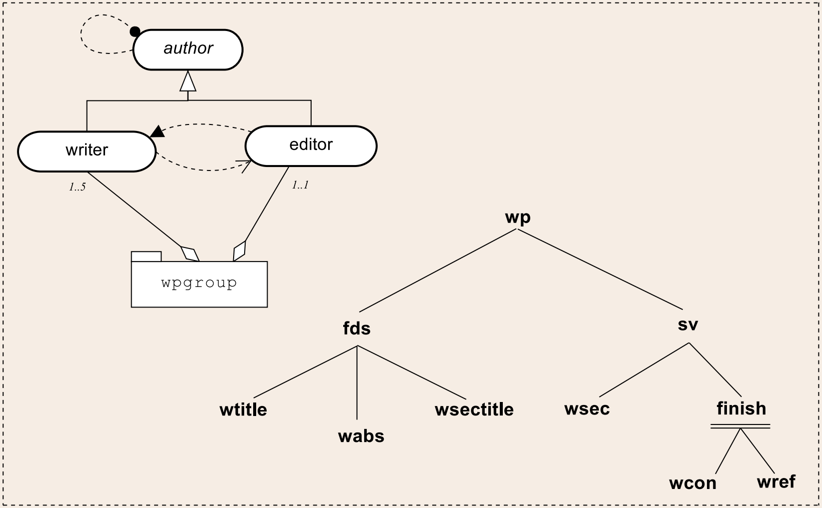
- Organisation Modelling Language (OML)
  - ⇝ allows the designer to specify the organisation of a MAS along three dimensions (structural, functional, deontic)
- Organisational Infrastructure
  - ⇝ interprets the OML and then constraints/supports the agents in the specified organisation
    - by means of regimentation, enforcement, tools for cooperative tasks, ...
    - allows agents to interact with the organisation (agent programming issues)
- Support for agent programming

$\mathcal{M}$OISE$^+$ by example: ‘writing a paper’



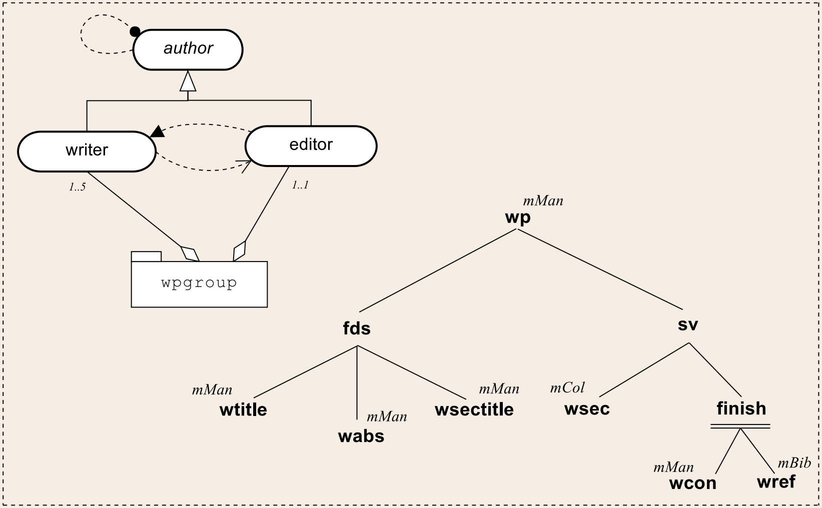
Structural Specification

$\mathcal{M}$OISE$^+$ by example: 'writing a paper'



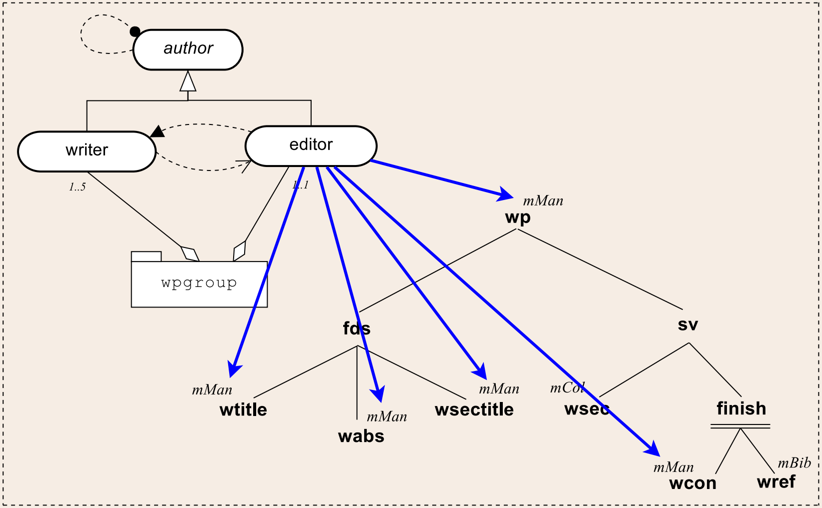
Functional Specification

$\mathcal{M}$OISE$^+$ by example: 'writing a paper'



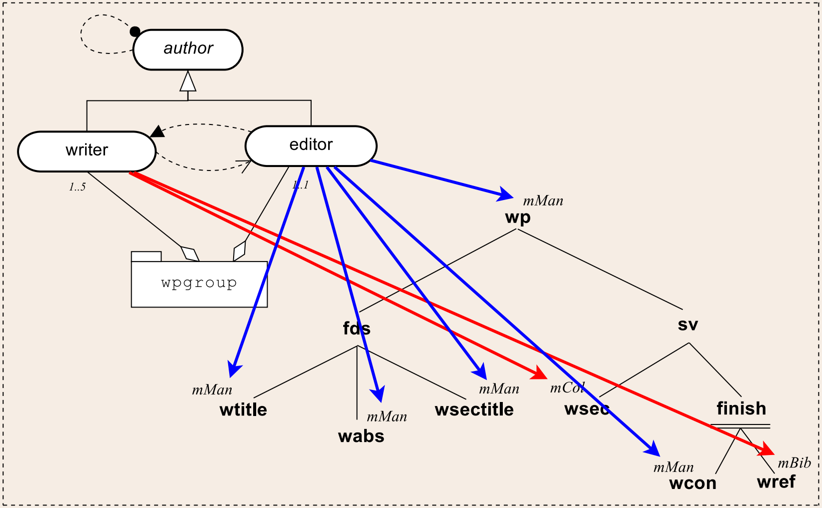
Missions

$\mathcal{M}$OISE$^+$ by example: ‘writing a paper’



Permissions

$\mathcal{M}$OISE$^+$ by example: 'writing a paper'



Obligations

$\mathcal{M}$OISE$^+$ software

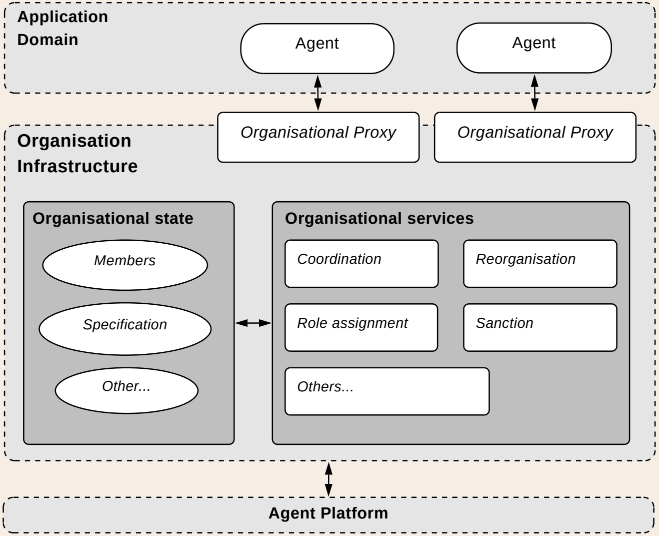
- Organisational infrastructures
  - $\mathcal{S}$-$\mathcal{M}$OISE$^+$ ('traditional' approach)
  - ORA4MAS (approach based on artifacts)
- Agent programming
  - $\mathcal{J}$-$\mathcal{M}$OISE$^+$ (BDI agent with ***Jason*** language)

agent level

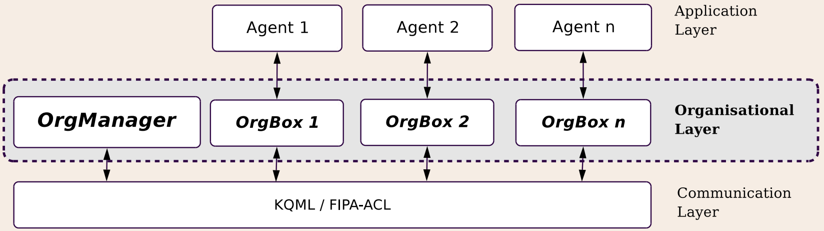
Ecole Nationale Supérieure des Mines SAINT-ETIENNE

Organisational Infrastructure (for open systems)

S-Moise+, Islander, steam, ...



$\mathcal{S}$-$\mathcal{M}$OISE$^+$: SACI + $\mathcal{M}$OISE$^+$



- Two main components: OrgManager and OrgBox

OrgBox

- The OrgBox is the interface that the agents use to access the organisational layer and thus the communication layer too
- OrgBox must be used to
  - Change the organisational entity (adopt a role, for instance)
  - Send a message to another agent
  - Get the organisational entity state
    - However, only a personalised version of the entity is given from OrgManager to OrgBox, respecting the acquaintance relation
- OrgManager notifies an agent's OrgBox about every change in the state of a scheme to which the agent has committed to
- No particular agent architecture is required

OrgManager

- Maintains the current state of the organisational entity
  - Created groups and schemes
  - Role assignments (Agents to Roles)
  - Mission assignments (Agents to Missions)
  - Change goal states (satisfied or not)
  - ...
- Maintains the current state of the organisational specification
- Receives messages from the other agents' OrgBoxes asking for changes in the organisational entity/specification
- Regiments some norms

Ecole Nationale Supérieure des Mines SAINT-ETIENNE

Organisational entity **dynamics**

The organisational entity is changed by requests coming from agents' OrgBoxes.

Examples of messages:

- `create_group("g1","wpgroup")`: a group called $g1$ is created using the 'wpgroup' group specification
- `create_scheme("wp", "g1")`: an instance of the 'wp' scheme specification is created; the agents in group $g1$ are responsible for this scheme's missions
- `adopt_role("bob", "editor", "g1")`: the agent 'bob' wants to adopt the role 'editor' in group 'g1'.
- ...

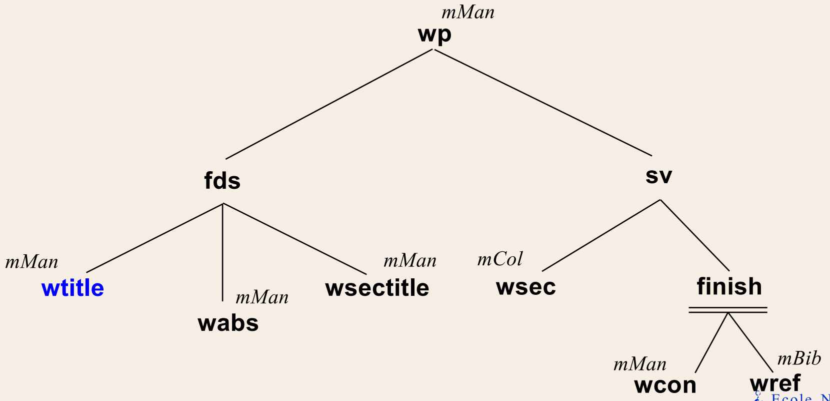
Regimentation of an organisational action
e.g. **role adoption**

The adoption of a role ρ by an agent α in group g has the following constraints:

- The role ρ must belong to the specification of group g
- The number of ρ players in g must be less than or equals to the maximum number of ρ players defined in the specification of group g
- For all roles ρ_i that agent α already plays in g, the roles ρ and ρ_i must be compatible in the specification of group g

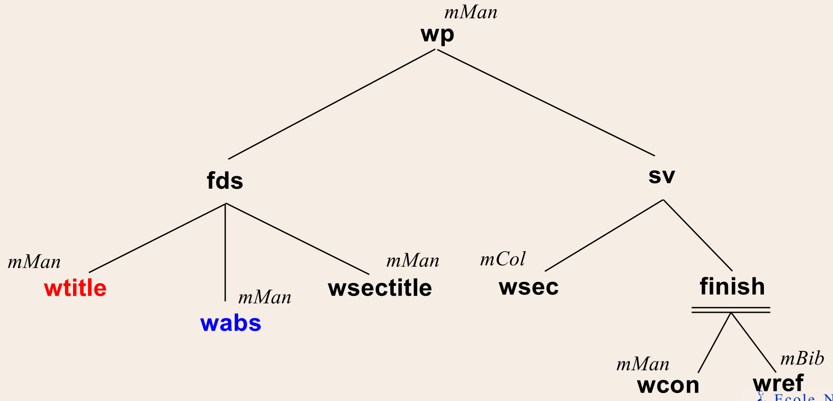
Goal's state and coordination

When an agent is committed to a mission, it is responsible for a number of goals. Only some of them might be possible at a given moment (those whose pre-condition goals are already satisfied)



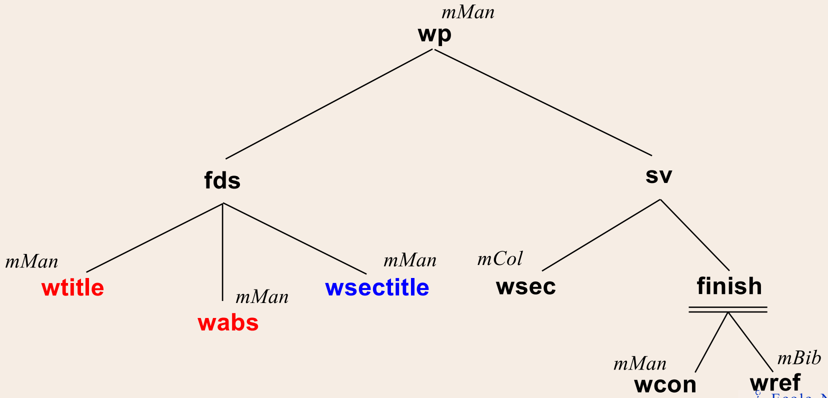
Goal's state and coordination

When an agent is committed to a mission, it is responsible for a number of goals. Only some of them might be possible at a given moment (those whose pre-condition goals are already satisfied)



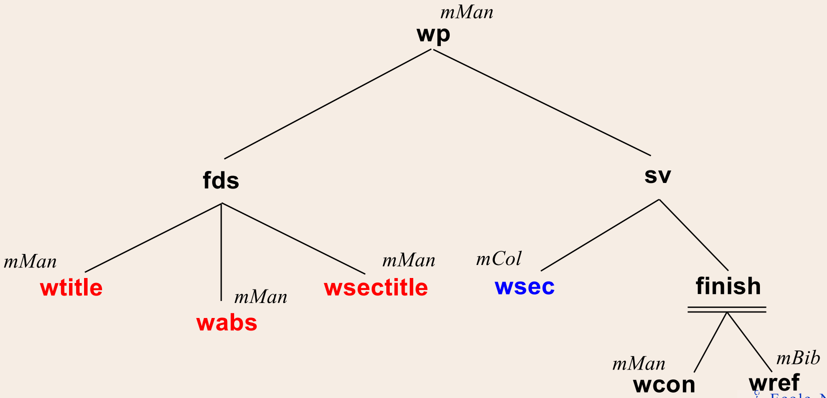
Goal's state and coordination

When an agent is committed to a mission, it is responsible for a number of goals. Only some of them might be possible at a given moment (those whose pre-condition goals are already satisfied)



Goal's state and coordination

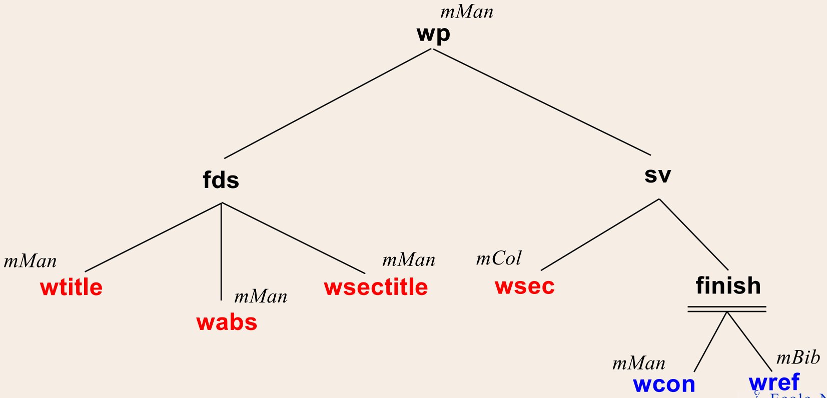
When an agent is committed to a mission, it is responsible for a number of goals. Only some of them might be possible at a given moment (those whose pre-condition goals are already satisfied)



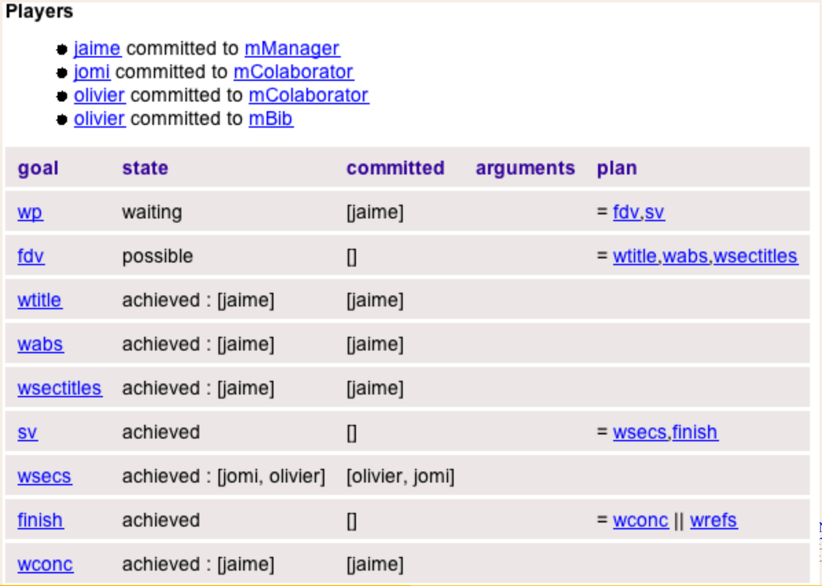
Ecole Nationale Supérieure des Mines SAINT-ETIENNE

Goal's state and coordination

When an agent is committed to a mission, it is responsible for a number of goals. Only some of them might be possible at a given moment (those whose pre-condition goals are already satisfied)



Useful tools — $\mathcal{M}$OISE$^+$ GUI



Players

- jaime committed to mManager
- jomi committed to mColaborator
- olivier committed to mColaborator
- olivier committed to mBib

goal	state	committed	arguments	plan
wp	waiting	[jaime]		= fdv,sv
fdv	possible	[]		= wtitle,wabs,wsectitles
wtitle	achieved : [jaime]	[jaime]		
wabs	achieved : [jaime]	[jaime]		
wsectitles	achieved : [jaime]	[jaime]		
sv	achieved	[]		= wsecs,finish
wsecs	achieved : [jomi, olivier]	[olivier, jomi]		
finish	achieved	[]		= wconc || wrefs
wconc	achieved : [jaime]	[jaime]		

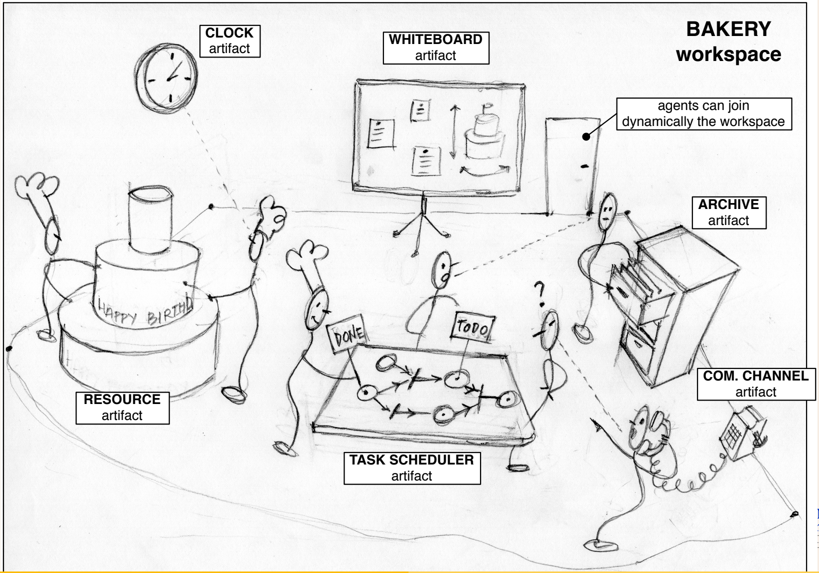
Motivations for another approach

- Organisational services are implemented as ‘special’ agents — which are conceptually different — agents doing services
- Organisational decisions are taken by the organisational infrastructure — the organisational infrastructure 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)

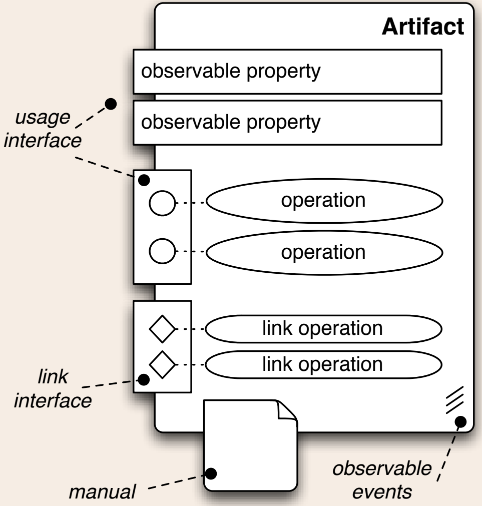
services taken decisions which are ‘closed’ for the agents

A&A model

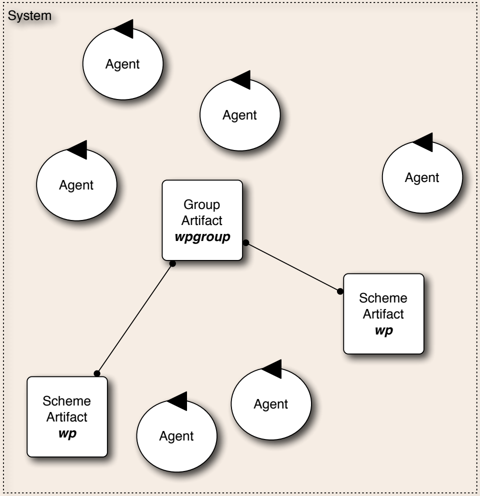
Artifacts, Agents, Workspaces [Ricci *et al.* 07]



Artifact model

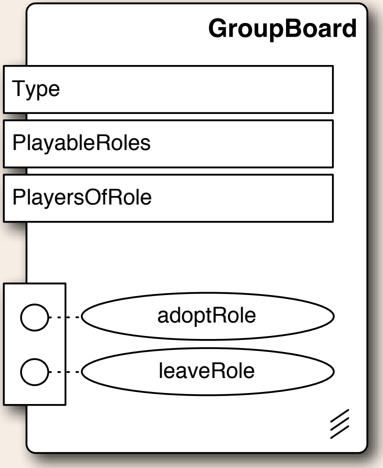


Organisational artifacts in ORA4MAS



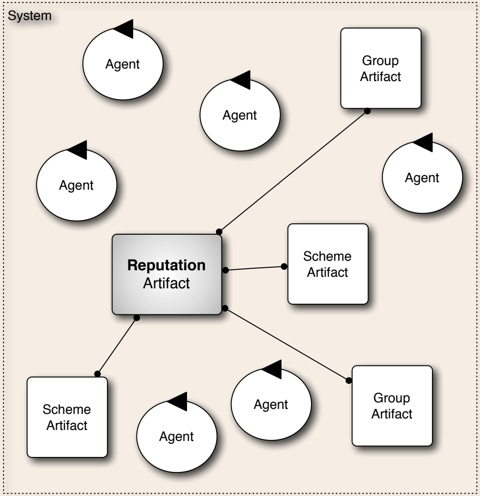
- based on A&A and $\mathcal{M}$OISE$^+$
- agents create and handle organisational artifacts
- artifacts in charge of regimentations, detection and evaluation of norms compliance
- agents are in charge of decisions about sanctions

Organisational artifacts in ORA4MAS



- based on A&A and $\mathcal{M}$OISE$^+$
- agents create and handle organisational artifacts
- artifacts in charge of regimentations, detection and evaluation of norms compliance
- agents are in charge of decisions about sanctions

Reputation Artifact in ORA4MAS



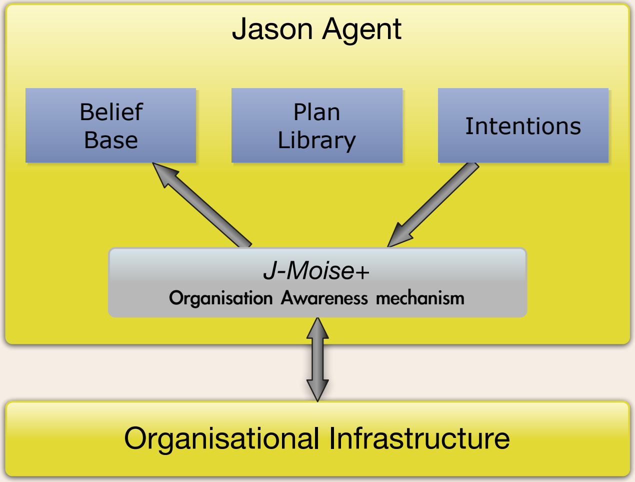
- Instrument to help in the enforcement of norms
- Indirect sanction system
- Considers the public character of the reputation process
- Publish an evaluation of the agents from the organisation point of view

Ecole Nationale Supérieure des Mines SAINT-ETIENNE

$\mathcal{J}$-$\mathcal{M}$OISE$^+$: ***Jason*** + $\mathcal{M}$OISE$^+$

- $\mathcal{S}$-$\mathcal{M}$OISE$^+$ and ORA4MAS provides general services for the agents to be organised, but does not help us to program the agents or the agents' reasoning about its organisation
- $\mathcal{J}$-$\mathcal{M}$OISE$^+$
  - Programming agents with AgentSpeak
  - BDI agents (reactive planning) – higher abstraction level
  - Help the programmer to determine when the agent should adopt a role, a mission, ...
  - Enable the agents to access organisational information
  - Independence from the distribution/communication layer
  - Using ***Jason***, an open-source interpreter for a variant of AgentSpeak, developed by Rafael Bordini and Jomi Hübner

General view


Jason Agent
Belief Base
Plan Library
Intentions
J-Moise+
Organisation Awareness mechanism
Organisational Infrastructure

Ecole Nationale Supérieure des Mines SAINT-ETIENNE

Organisational **actions** in AgentSpeak I

Example (AgentSpeak plan)

```
+some_event : some_context
  <- jmoise.create_group(wpgroup).
```

Some available Organisational Actions:

- For groups:
  - `create_group(<GrSpecId>[,<SuperGrId>])`
  - `remove_group(<GrId>)`
- For schemes:
  - `create_scheme(<SchSpecId> [,<groups>])`
  - `add_responsible_group(<SchId>,<GrId>)`
  - `remove_scheme(<SchId>)`
  - `set_goal_state(<SchId>,<Goal>,<State>)`

Ecole Nationale Supérieure des Mines SAINT-ETIENNE

Organisational **actions** in AgentSpeak II

- For Agents:
  - `adopt_role(<RoleId>,<GrId>)`
  - `remove_role(<RoleId>,<GrId>)`
  - `commit_mission(<MisId>,<SchId>)`
  - `remove_mission([<MisId>,] <SchId>)`
- Those actions are executed under regimentation (to avoid an inconsistent organisational state)
  e.g. the adoption of role is constrained by
  - the cardinality of the role in the group
  - the compatibilities of the roles played by the agent

Ecole Nationale Supérieure des Mines SAINT-ETIENNE

Handling organisational **events** in AgentSpeak

Whenever something changes in the organisation, the agent architecture updates the agent belief base accordingly producing events (belief update from perception)

Example (A new group is created)

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

or

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

Example (Some group is destroyed)

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

Available organisational events I

- +/- group(< *GrSpecId* >,< *GrId* >)
  [owner(< *AgName* >), super_gr(*G*)]:
  perceived by all agents when a group is created (event +) or removed (event −) by *AgName*
- +/- play(< *AgName* >, < *RoleId* >, < *GrId* >):
  perceived by the agents of *GrId* when an agent adopts (event +) or remove (event −) a role in group *GrId*
- +/- commitment(< *AgName* >, < *MisId* >, < *SchId* >):
  perceived by the *SchId* players when an agent commits or removes a commitment to a mission *MisId* in scheme *SchId*

Available organisational events II

- +/- scheme(< *SchSpecId* >,< *SchId* >)
  [owner(< *AgName* >)]:
  perceived by all agents when a scheme is created (+) or finished (−) by *AgName*
- + scheme_group(< *SchId* >,< *GrId* >):
  perceived by *GrId* players when this group becomes responsible for the scheme *SchId*
- + goal_state(< *SchId* >, < *GoalId* >, < *State* >):
  perceived by *SchId* players when the state of some goal changes

Available organisational events III

- $+$ obligation($<$ *SchId* $>$, $<$ *MisId* $>$) [role($<$ *RoleId* $>$),group($<$ *GrId* $>$)]: perceived by an agent when it has an organisational obligation for a mission. It has a role (*RoleId*) in a group (*GrId*) responsible for a scheme (*SchId*) and this role is obligated to a mission in this scheme
- $+$ permission($<$ *SchId* $>$, $<$ *MisId* $>$) [role($<$ *RoleId* $>$),group($<$ *GrId* $>$)]

Achieving organisational **goals**

An achievement goal event (+!g) is create when an organisational goal g is possible

Example (Organisational goal)

If an agent is committed to a mission with goal wsec, whenever this goal is possible (all its pre-condition goals are satisfied), the following plan may be selected:

```
+!wsec[scheme(Sch)]
  :  commitment(A, mBib, Sch)
  <- ..... actions to write the section .....;
     .send(A,tell,[references]);
     jmoise.set_goal_state(Sch, wsec, satisfied).
```

The context of this plan uses organisational information to constrain its execution.

Example: Writing paper

Organisation Specification

```
<organisational-specification
  <structural-specification>
     <role-definitions>
        <role id="author" />
        <role id="writer"> <extends role="author"/> </role>
        <role id="editor"> <extends role="author"/> </role>
     </role-definitions>

     <group-specification id="wpgroup">
        <roles>
           <role id="writer" min="1" max="5" />
           <role id="editor" min="1" max="1" />
        </roles>
        ...
```

Execution sample I

jaime	action: jmoise.create_group(wpgroup)
all	perception: group(wpgroup,g1)[owner(jaime)]
jaime	action: jmoise.adopt_role(editor,g1)
olivier	action: jmoise.adopt_role(writer,g1)
jomi	action: jmoise.adopt_role(writer,g1)

all	perception:
	play(jaime,editor,g1)
	play(olivier,writer,g1)
	play(jomi,writer,g1)

Execution sample II

jaime	action: jmoise.create_scheme(writePaperSch, [g1])
all	perception: scheme(writePaperSch,s1)[owner(jaime)]
all	perception: scheme_group(s1,g1)
jaime	perception: permission(s1,mManager)[role(editor),group(wpgroup)]
jaime	action: jmoise.commit_mission(mManager,s1)
olivier	perception: obligation(s1,mColaborator)[role(writer),group(wpgroup), obligation(s1,mBib)[role(writer),group(wpgroup)
olivier	action: jmoise.commit_mission(mColaborator,s1)
olivier	action: jmoise.commit_mission(mBib,s1)

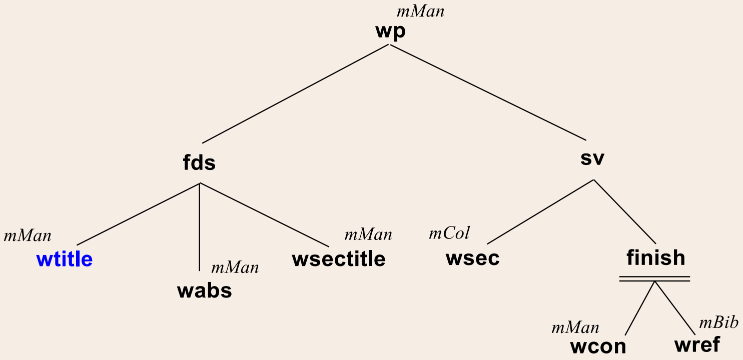
Execution sample III

jomi perception:
obligation(s1,mColaborator)[role(writer),group(wpgroup),
obligation(s1,mBib)[role(writer),group(wpgroup)]

jomi action: jmoise.commit_mission(mColaborator,s1)

all perception:
commitment(jaime,mManager,s1)
commitment(olivier,mColaborator,s1)
commitment(olivier,mBib,s1)
commitment(jomi,mColaborator,s1)

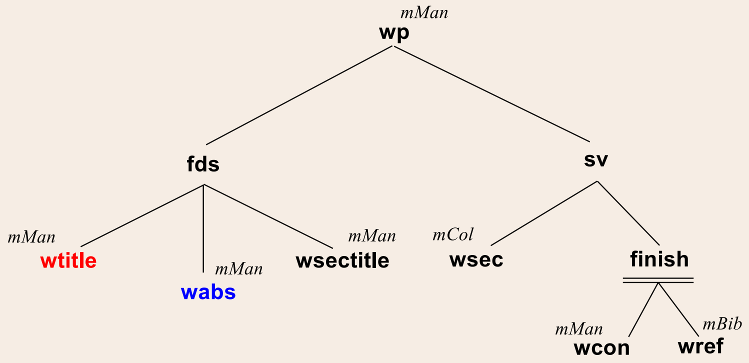
Execution sample IV



all perception: goal_state(s1,*,unsatisfied)

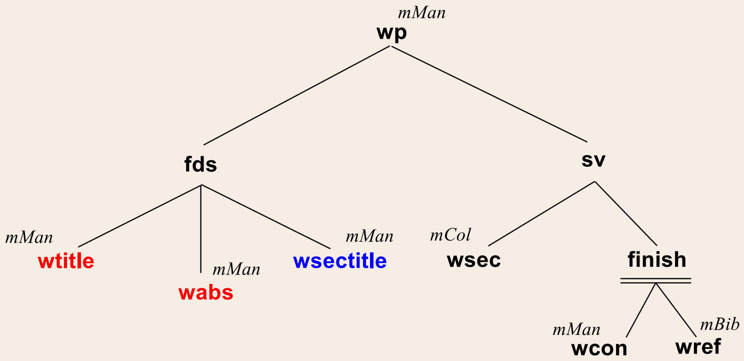
jaime (only wtitle is possible, Jaime should work)
event: +!wtitle
action: jmoise.set_goal_state(s1,wtitle,satisfied)

Execution sample V



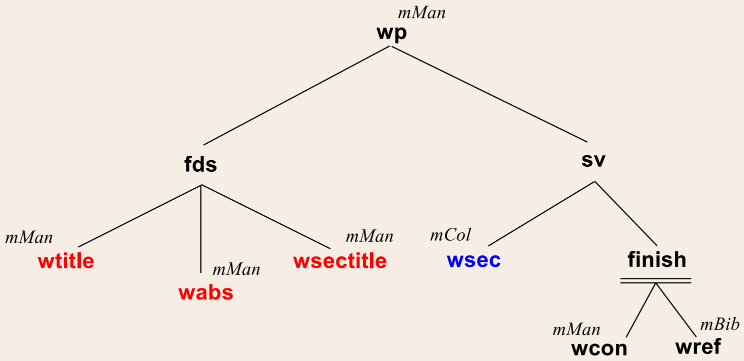
jaime event: +!wabs
action: jmoise.set_goal_state(s1,wabs,satisfied)

Execution sample VI



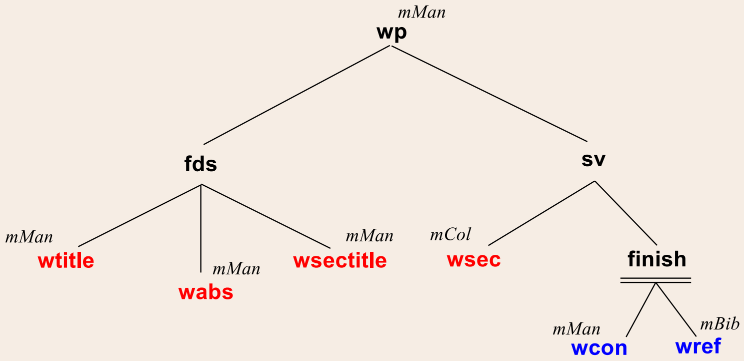
jaime event: +!wsectitles
action: jmoise.set_goal_state(s1,wsectitles,satisfied)

Execution sample VII



olivier, jomi event: +!wsecs
action: jmoise.set_goal_state(s1,wsecs,satisfied)

Execution sample VIII



jaime event: +!wcon; ...

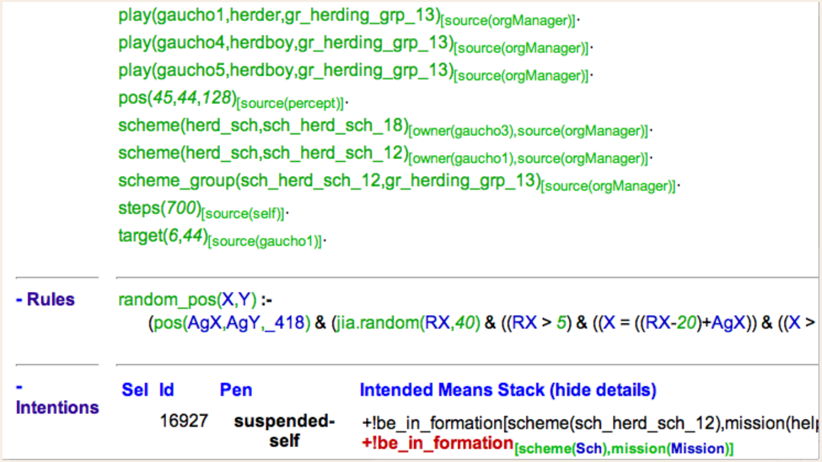
olivier event: +!wref; ...

Execution sample IX

all action: jmoise.remove_mission(s1)

jaime action: jmoise.jmoise.remove_scheme(s1)

Useful tools — Mind inspector



```
play(gaucho1,herder,gr_herding_grp_13)[source(orgManager)].
play(gaucho4,herdboy,gr_herding_grp_13)[source(orgManager)].
play(gaucho5,herdboy,gr_herding_grp_13)[source(orgManager)].
pos(45,44,128)[source(percept)].
scheme(herd_sch,sch_herd_sch_18)[owner(gaucho3),source(orgManager)].
scheme(herd_sch,sch_herd_sch_12)[owner(gaucho1),source(orgManager)].
scheme_group(sch_herd_sch_12,gr_herding_grp_13)[source(orgManager)].
steps(700)[source(self)].
target(6,44)[source(gaucho1)].
```

- Rules

```
random_pos(X,Y) :-
   (pos(AgX,AgY,_418) & (jia.random(RX,40) & ((RX > 5) & ((X = ((RX-20)+AgX)) & ((X >
```

- Intentions

Sel	Id	Pen	Intended Means Stack (hide details)
	16927	suspended-self	+!be_in_formation[scheme(sch_herd_sch_12),mission(hel +!be_in_formation[scheme(Sch),mission(Mission)]

Summary — $\mathcal{S}$-$\mathcal{M}$OISE$^+$

- Ensures that the agents follow some of the norms specified for the organisation (cardinality of groups, communication and acquaintance links, role and mission adoption, goal satisfaction)
- The organisation is interpreted at runtime, it is not hardwired in the agents code
- It has a synchronisation mechanism for scheme execution
- It is suitable for open systems as no specific agent architecture is required
- An implementation is available at `http://moise.sourceforge.net`

Summary — ORA4MAS

- Same services of $\mathcal{S}$-$\mathcal{M}$OISE$^+$
- Based on artifacts that agents can handle (non-autonomous part of the system)
- on going work — Rosine Kitio

Summary — $\mathcal{J}$-$\mathcal{M}$OISE$^+$

- Supports the development of organised agents using
  - Logic-based language
  - BDI architecture
  - AgentSpeak agent-oriented programming language
  - Declarative and goal oriented programming
  - Meta-programming
    `.drop_intention(_[role(writer)])`
- Approach based on
  - Organisational actions, events, and goals
- But, it is 'just' an integration,
  it still lacks organisational reasoning
  (ongoing work [Cosmin])

Summary — $\mathcal{J}$-$\mathcal{M}$OISE$^+$

- Supports the development of organised agents using
  - Logic-based language
  - BDI architecture
  - AgentSpeak agent-oriented programming language
  - Declarative and goal oriented programming
  - Meta-programming
    `.drop_intention(_[role(writer)])`
- Approach based on
  - Organisational actions, events, and goals
- But, it is 'just' an integration,
  it still lacks organisational reasoning
  (ongoing work [Cosmin])

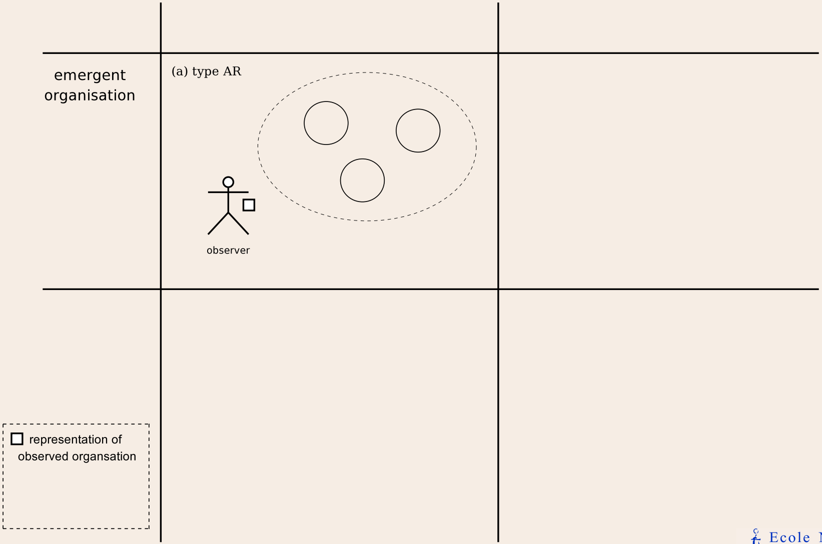
Ecole Nationale Supérieure des Mines SAINT-ETIENNE

More information

- `http://moise.sf.net`
- `http://jason.sf.net`

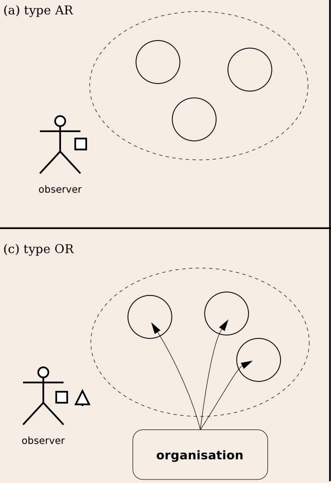
- J. F. Hübner, J. S. Sichman, and O. Boissier. Developing organised multi-agent systems using the $\mathcal{M}\text{OISE}^{+}$ model: Programming issues at the system and agent levels. *Int. J.Agent-Oriented Software Engineering*, 1(3/4):370–395, 2007.

Points of view on organisation



Points of view on organisation

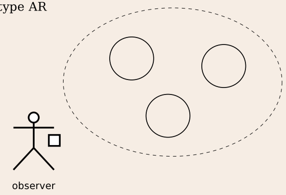
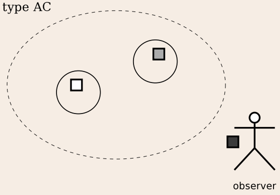
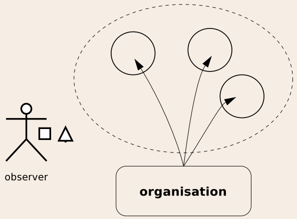
emergent organisation	(a) type AR	
pre-defined organisation	(c) type OR	



□ representation of observed organsation

△ representation of institutional organsation

Points of view on organisation

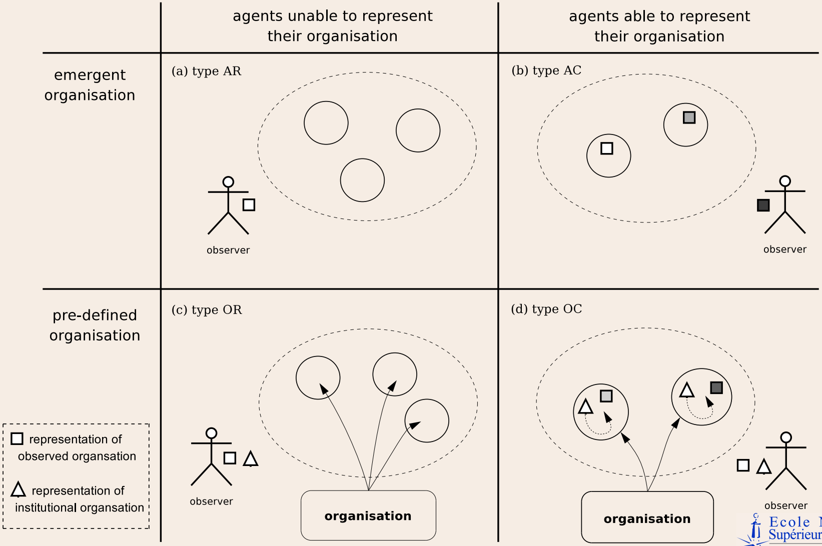
	agents unable to represent their organisation	agents able to represent their organisation
emergent organisation	(a) type AR 	(b) type AC 
pre-defined organisation	(c) type OR 	

□ representation of observed organsation

△ representation of institutional organsation

Ecole Nationale Supérieure des Mines SAINT-ETIENNE

Points of view on organisation


agents unable to represent their organisation
agents able to represent their organisation
emergent organisation
pre-defined organisation
(a) type AR
observer
(b) type AC
observer
(c) type OR
observer
organisation
(d) type OC
observer
organisation
□ representation of observed organsation
△ representation of institutional organsation