Organisation-Oriented Programming of MAS

From Closed to Open MAS

O. Boissier(1)

J.F. Hübner(1,2), R. Kitio(1), A. Ricci(3), J.S. Sichman(4)

(1) MAS/G2I/ENSM.SE
France

(2) GIA/DSC/FURB
Brazil

(3) DEIS/Univ. Bologna
Italy

(4) LTI/USP
Brazil

Organisations in MAS

- Organisation in MAS is

  a supra-agent pattern

  of **emergent** or (**pre**)**defined** agents cooperation,

  for a purpose,

  that could be defined by the designer or by the agents themselves.

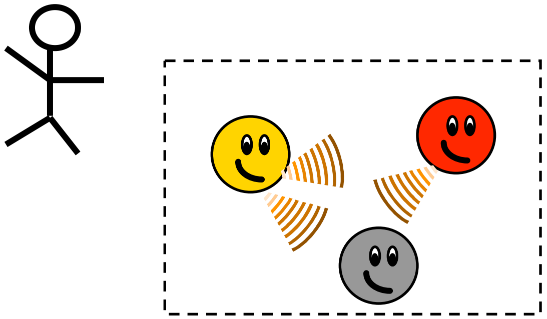
→ Pattern of emergent/potential cooperation

- Organisation entity, institution, social relations, commitments

→ Pattern of (pre)defined cooperation

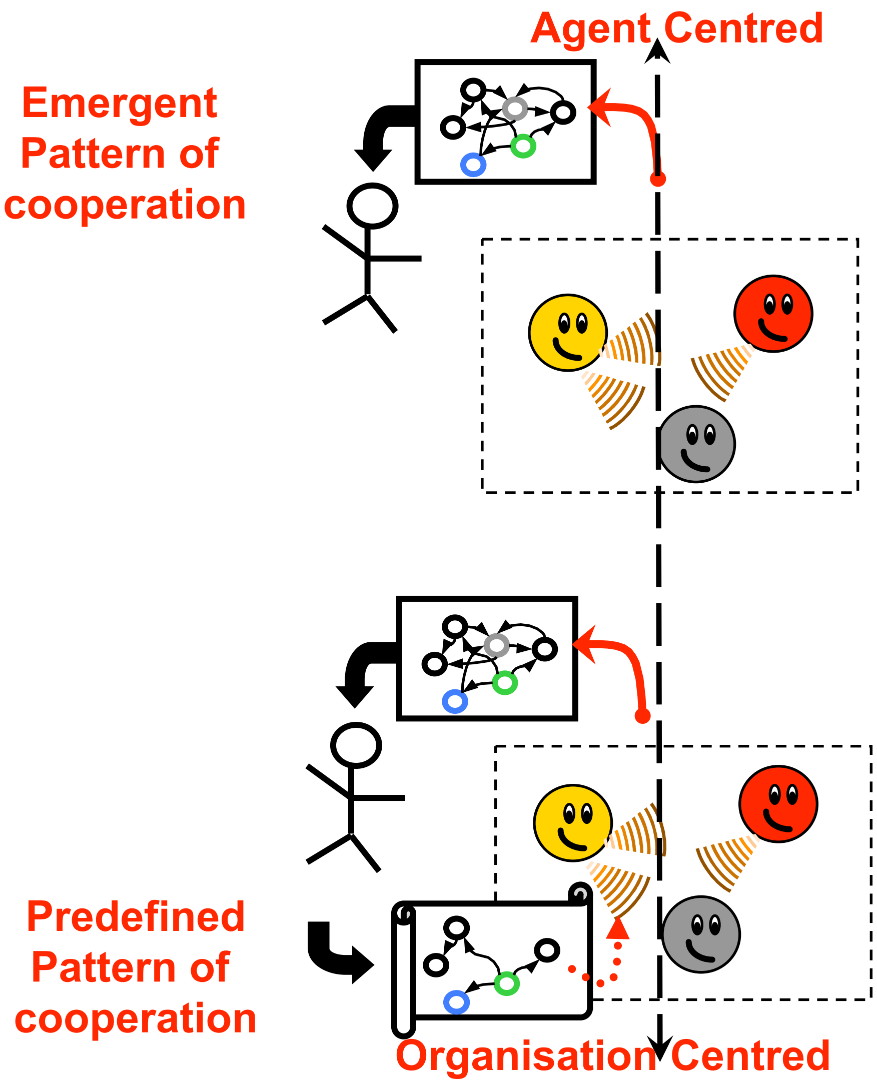
- Organizational structure, norms, …

Organisations in MAS (global picture)



Designer / Observer

Organisations in MAS (global picture)



Agent Centred

Emergent Pattern of cooperation

Predefined Pattern of cooperation

Organisation Centred

Organization Specification

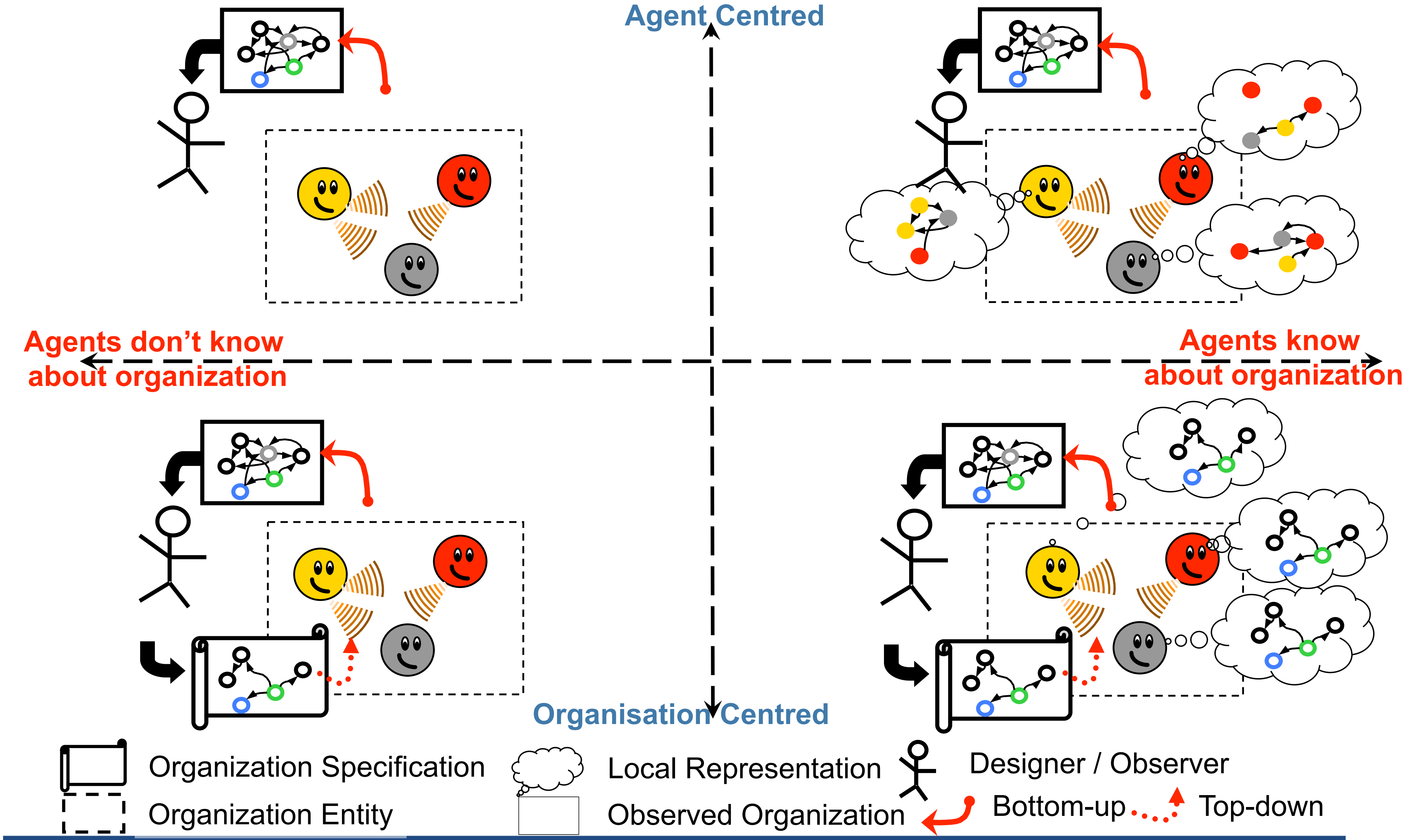
Organization Entity

□ Observed Organization

Designer / Observer

Bottom-up ... Top-down

Organisations in MAS (global picture)



Agent Centred

Organisation Centred

Agents don’t know about organization

Agents know about organization

Organization Specification

Organization Entity

Local Representation

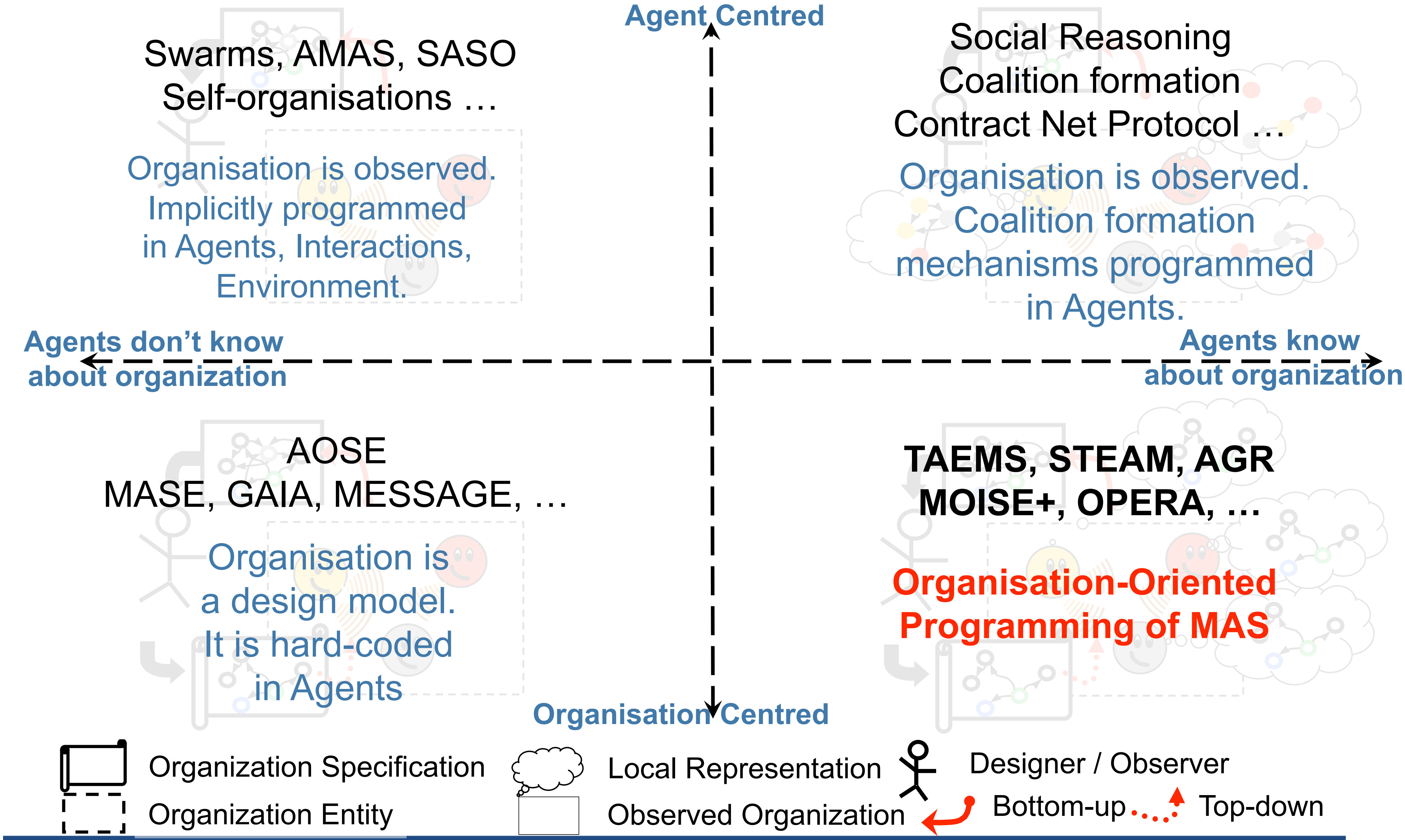
Observed Organization

Designer / Observer

Bottom-up

Top-down

Organisations in MAS (approaches)



Agent Centred

Organisation Centred

Agents don't know about organization

Agents know about organization

Swarms, AMAS, SASO
Self-organisations …

Organisation is observed.
Implicitly programmed
in Agents, Interactions,
Environment.

Social Reasoning
Coalition formation
Contract Net Protocol …

Organisation is observed.
Coalition formation
mechanisms programmed
in Agents.

AOSE
MASE, GAIA, MESSAGE, …

Organisation is
a design model.
It is hard-coded
in Agents

TAEMS, STEAM, AGR
MOISE+, OPERA, …

Organisation-Oriented Programming of MAS

Organization Specification

Organization Entity

Local Representation

Observed Organization

Designer / Observer

Bottom-up

Top-down

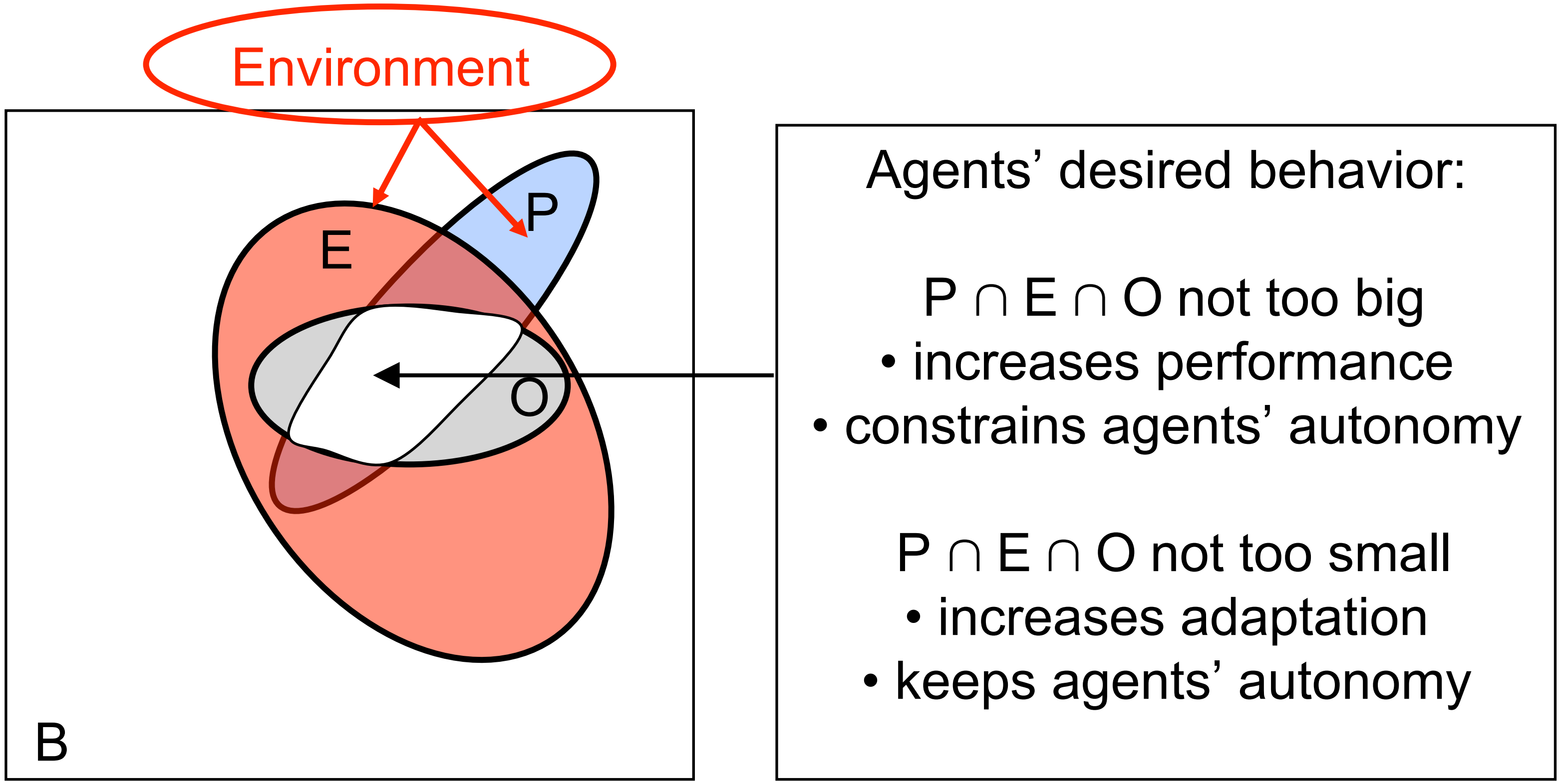
Outline

1. Introduction
2. **Organisation Oriented Programming**
3. Organisation Modeling Language
4. Organisation Infrastructure
5. Conclusion and Perspectives

Motivations

- Organisations are usefull to program multi-agent systems:
  - To deal with heterogeneous agents' autonomy i.e. controlling their actions, goals, … while keeping their autonomy
  - To help heterogeneous agents to coordinate to achieve common goals
- e.g.
  when someone adopts the role of master student in a laboratory, she remains autonomous to perform its research but should follow some rules of the laboratory.
  These rules can be:
  ''*the access to computers requires an username*''
  ''*a master thesis should be written in two years*''
- The master student is free to adopt the role, but once adopted the organisation expects her to limit her autonomy.

Multi-Agent Organisation and Autonomy



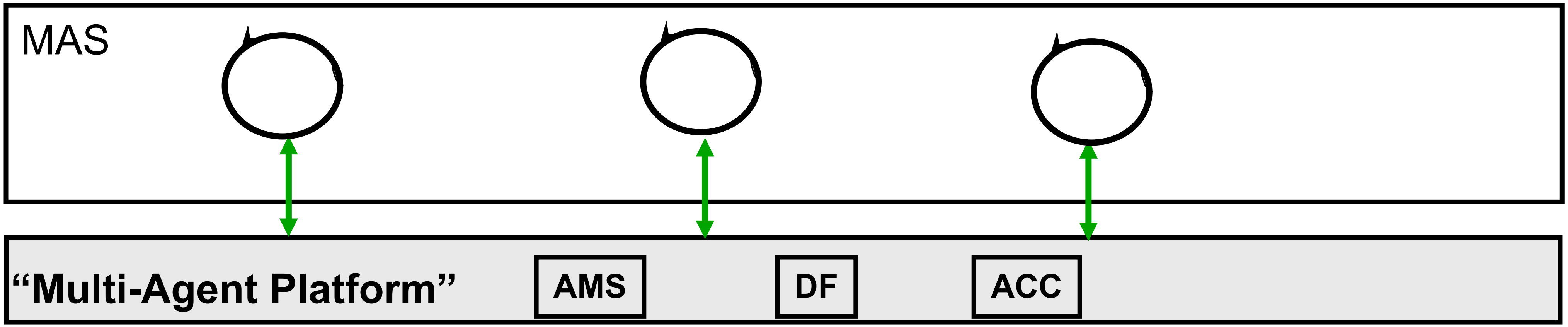
Agents’ desired behavior:

P ∩ E ∩ O not too big
- increases performance
- constrains agents’ autonomy

P ∩ E ∩ O not too small
- increases adaptation
- keeps agents’ autonomy

B: agents’ possible behaviors
P: agents’ behaviors that lead to global purpose
E: agents’ possible behaviors constrained by the environment
O: agents’ possible/permitted/obliged behaviors constrained by the organisation

Organisation Oriented Programming

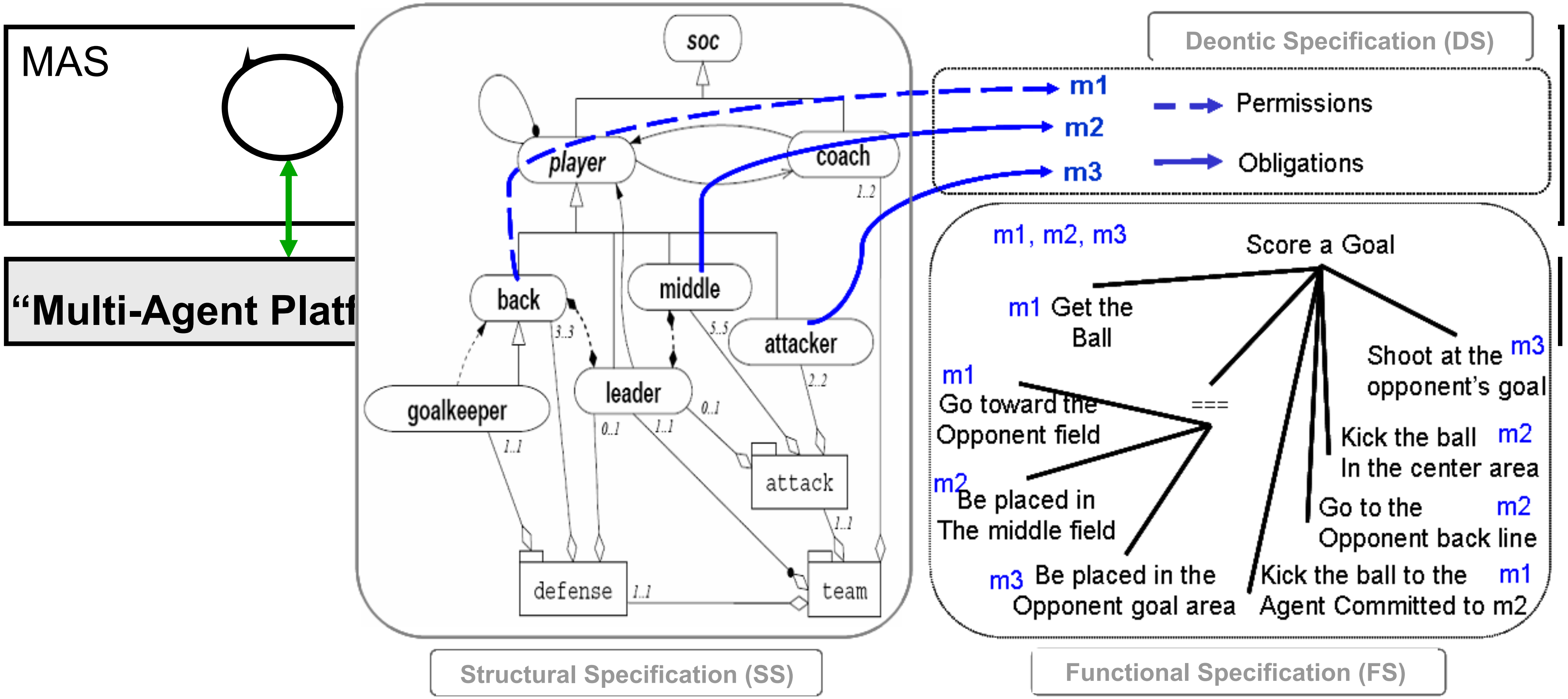


Infrastructure/Platforms for MAS in which domain agents operate on top of a middleware layer

- Supporting the implementation of agents
- Providing fundamental global generic services

e.g. JADE, FIPA-OS, SACI, …

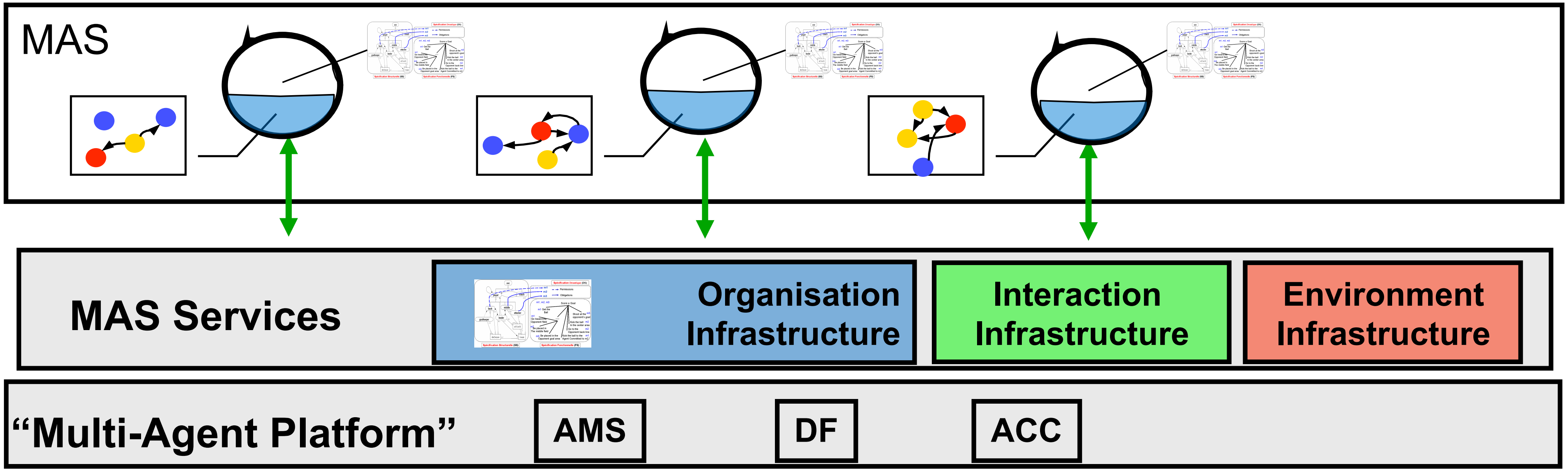
Organisation Oriented Programming


MAS
“Multi-Agent Platf
soc
player
coach
back
middle
attacker
goalkeeper
leader
attack
defense
team
1..2
3..3
5..5
2..2
0..1
0..1
1..1
1..1
1..1
1..1
Structural Specification (SS)
Deontic Specification (DS)
m1
m2
m3
Permissions
Obligations
m1, m2, m3
Score a Goal
m1 Get the Ball
m1 Go toward the Opponent field
m2 Be placed in The middle field
m3 Be placed in the Opponent goal area
===
Shoot at the opponent’s goal m3
Kick the ball In the center area m2
Go to the Opponent back line m2
Kick the ball to the Agent Committed to m2 m1
Functional Specification (FS)

Organisation Modeling Language (OML): Declarative specification of Multi-Agent Organisation(s), collecting and expressing specific constraints and cooperation patterns imposed on the agents

e.g. AGR [Ferber 98], Teamwork [Tambe 98], Islander [Esteva 01], Moise+ [Hubner 02], …

Organisation Oriented Programming



MAS

MAS Services

Organisation Infrastructure

Interaction Infrastructure

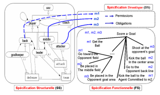
Environment Infrastructure

“Multi-Agent Platform”

AMS

DF

ACC



Org. Specification written with Organisation Modeling Language (OML)

Organisation Implementation Architecture (OIA)

supports the management of the **Organisation Entity**

Organisation-awareness agent mechanisms transforming agents into organisation-deliberative agent (e.g. Jason-MOISE [Hubner 06], Autonomy based reasoning [Carabelea 07])

Organisation Infrastructure e.g. Madkit [Gutknecht 98], Karma [Pynadath 03], Ameli [Esteva 04], S-Moise+ [Hubner 05], SYNAI [Gateau 07], …

Outline

1. Introduction
2. Organisation Oriented Programming
3. **Organisation Modeling Language**

    MOISE+, MOISE^{inst}, …

4. Organisation Infrastructure
5. Conclusion and Perspectives

Motivations

- Organisation is explicitly represented as:
  - Organisation specification,
  - Organisation entity,
  - Local representations of organisation entity

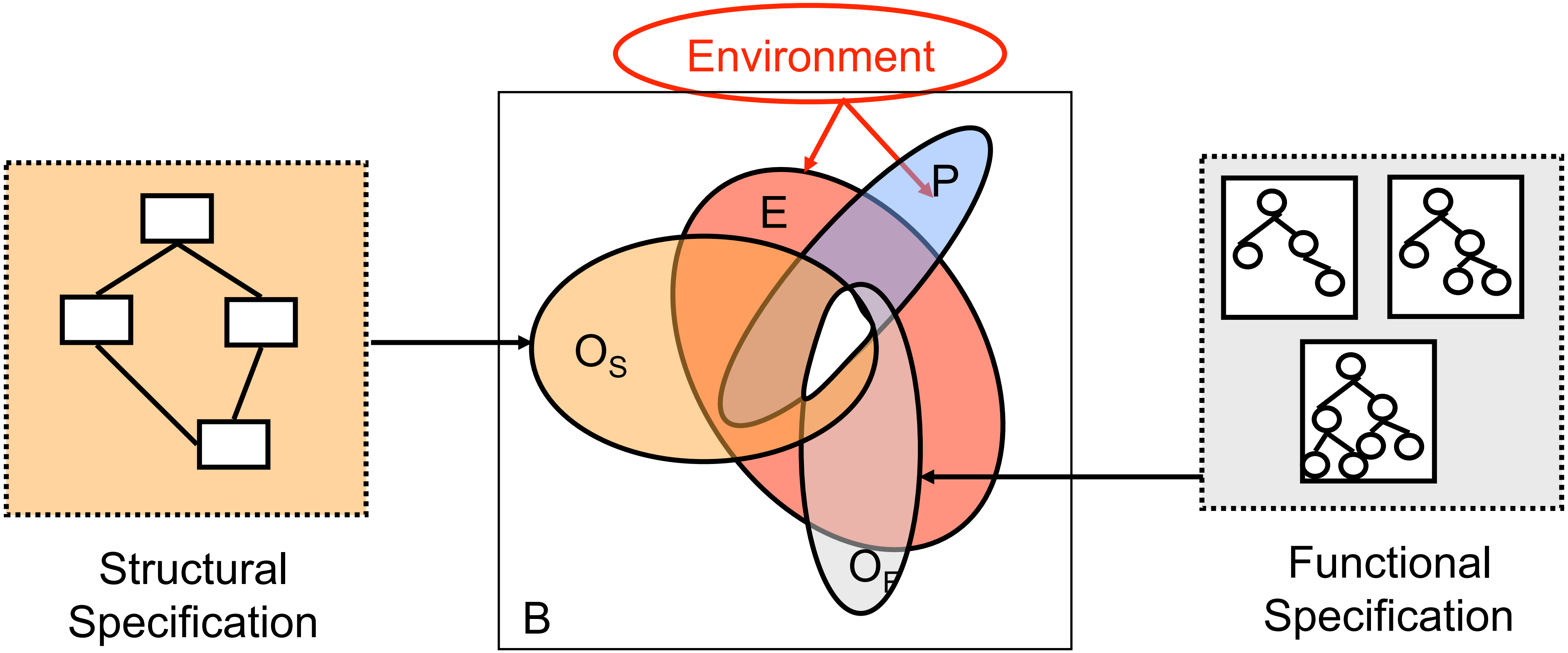
- Agents can “reason” about the organisation
  - to enter into/leave from the organisation entity during execution
  - to change/adapt the organisation entity, organisation specification
  - to obey/disobey the organisation
    - organisation is no more a regimentation

- Organisation Infrastructure can
  - Manage the organisation entity and support the agents in their cooperation
  - Monitor and regulate the agents in the organisation entity

MOISE+ Organisation Modeling Language

- Model of Organisation for multI-agent SystEms [Hannoun 98, Hübner 03]
- http://moise.sourceforge.net/
- Distinguishes three main dimensions in the organisation of a Multi-Agent System:
  - Structural Specification
  - Functional Specification
  - Deontic Specification

MOISE+ OML Dimensions



Structural Specification

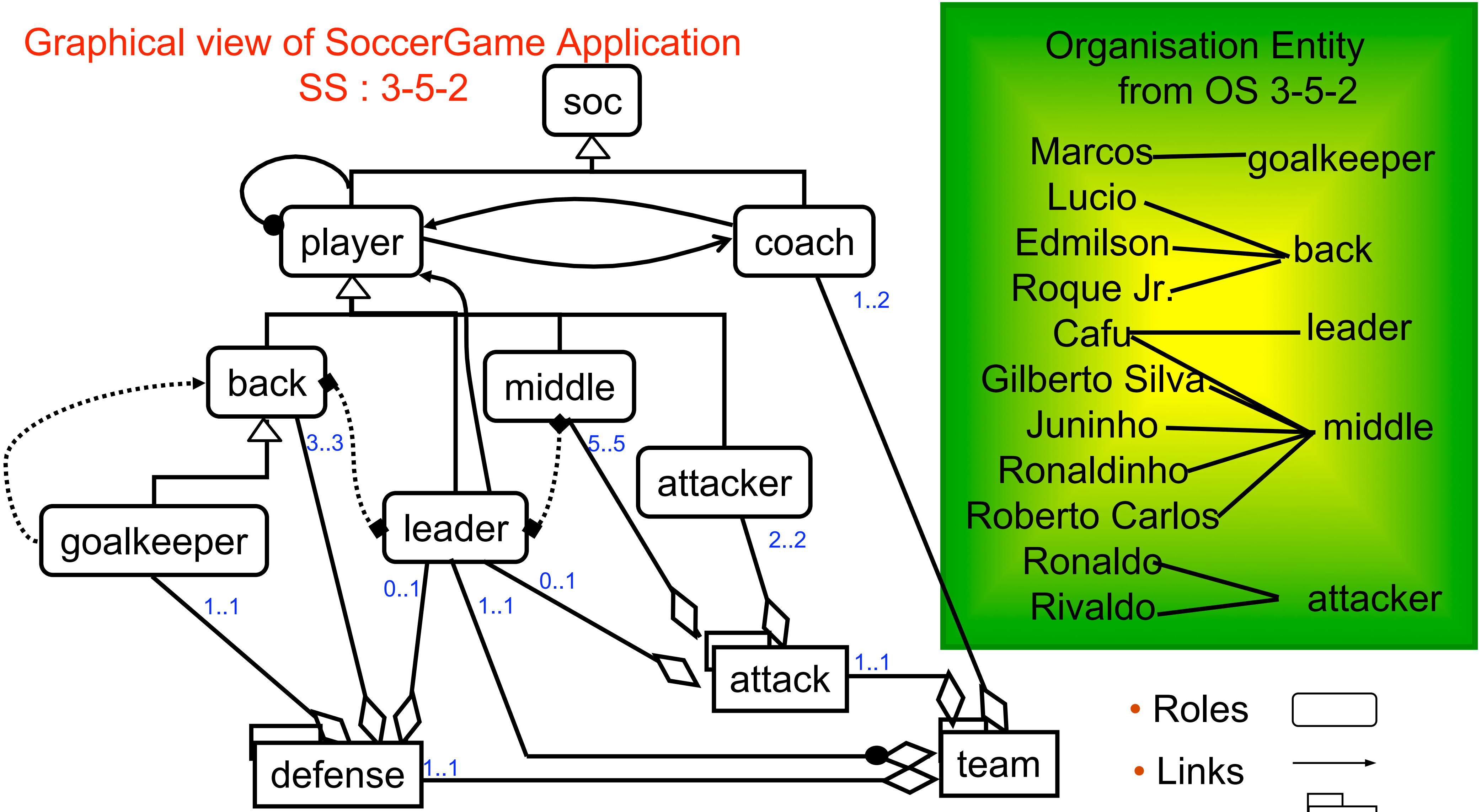
Functional Specification

Groups, links, roles
Compatibilities, multiplicities inheritance

Global goals, plans, Missions, schemas, preferences

Deontic Specification:
Permissions, Obligations
Agents' autonomy is considered!

MOISE+ OML: Structural Specification



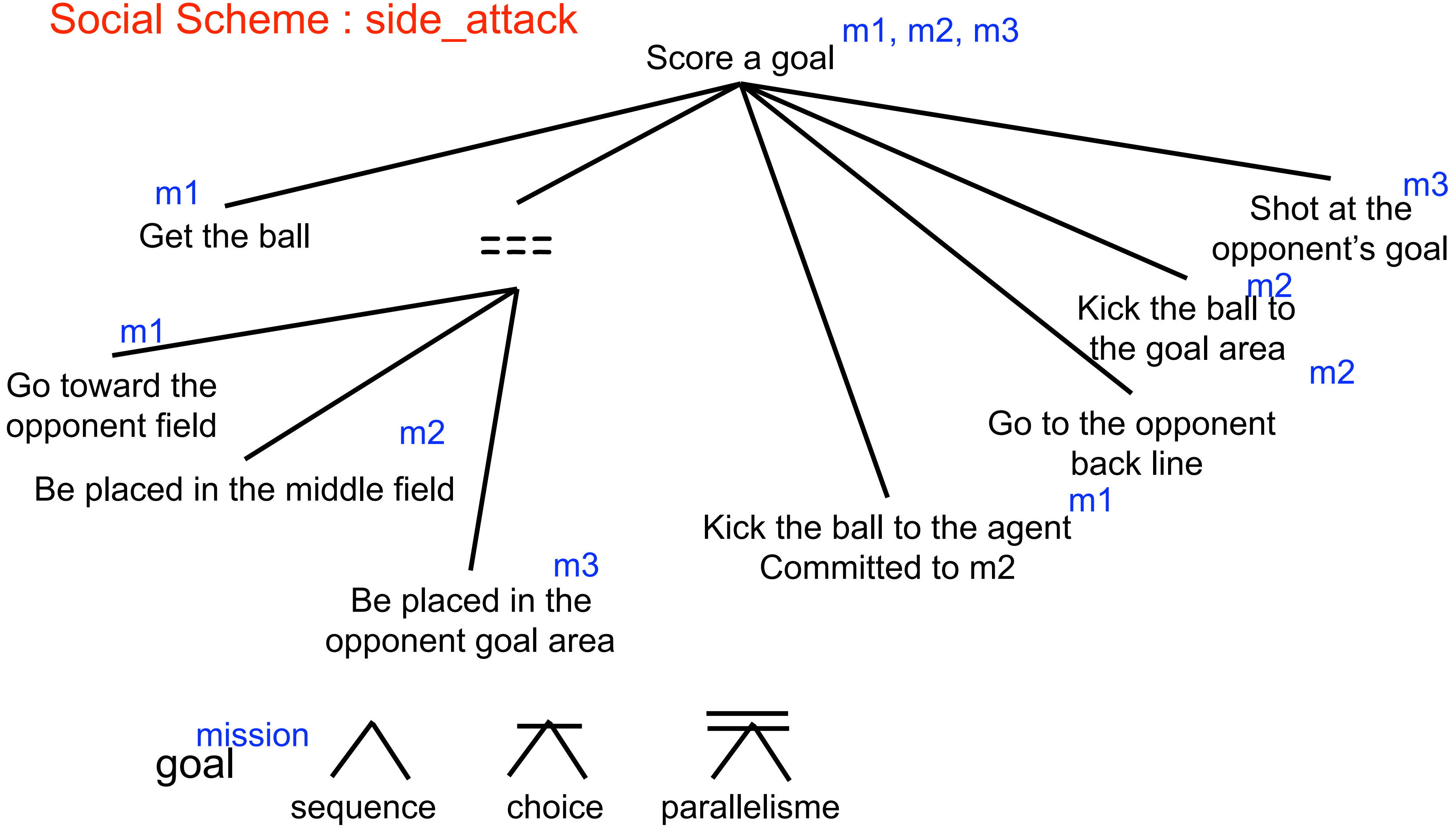
Graphical view of SoccerGame Application
SS : 3-5-2

- Roles
- Links
- Groups

MOISE+ OML: Functional Specification

Graphical view of SoccerGame Application FS

Social Scheme : side_attack

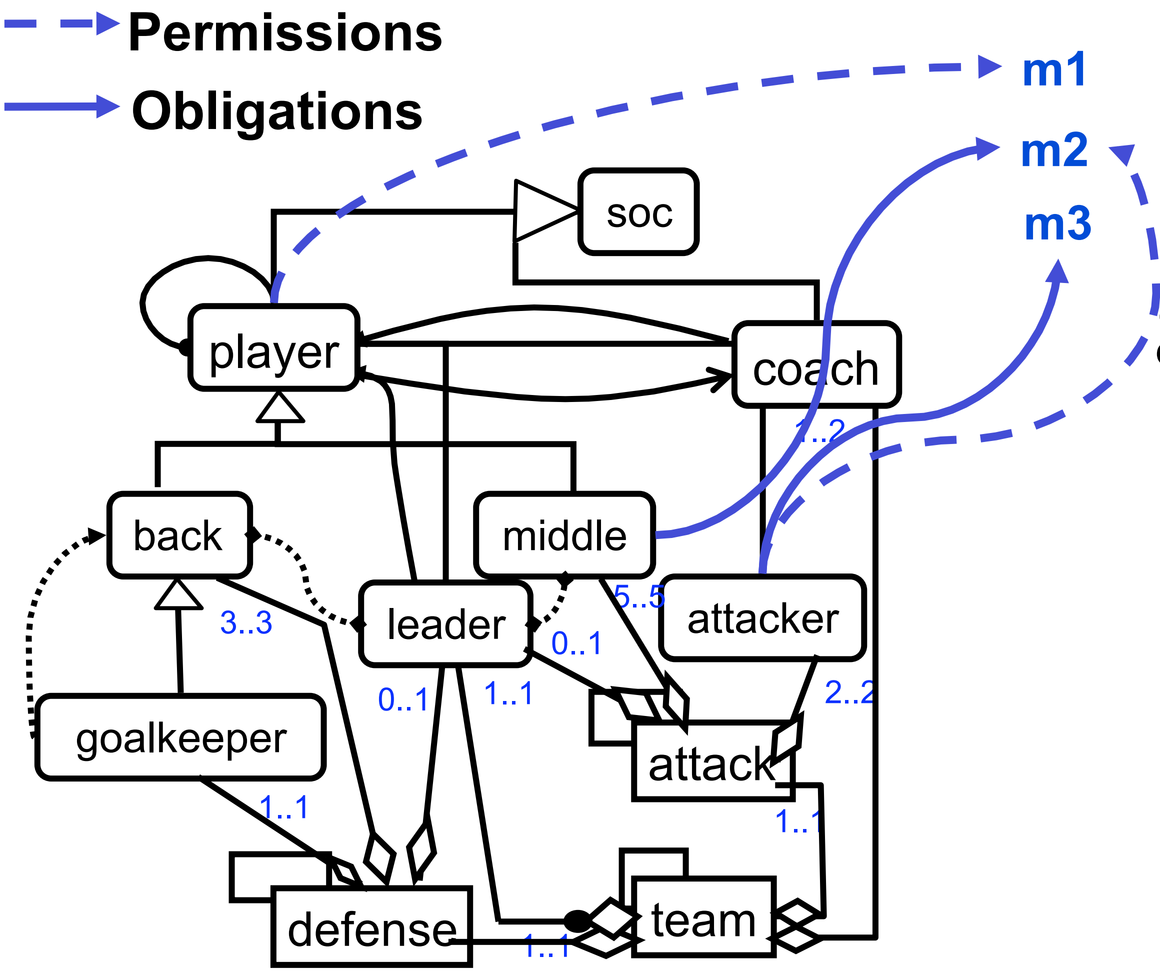

m1, m2, m3
Score a goal
m1
Get the ball
m1
Go toward the opponent field
m2
Be placed in the middle field
m3
Be placed in the opponent goal area
Kick the ball to the agent Committed to m2
m1
Go to the opponent back line
m2
Kick the ball to the goal area
m2
Shot at the opponent's goal
m3
mission
goal
sequence
choice
parallelisme

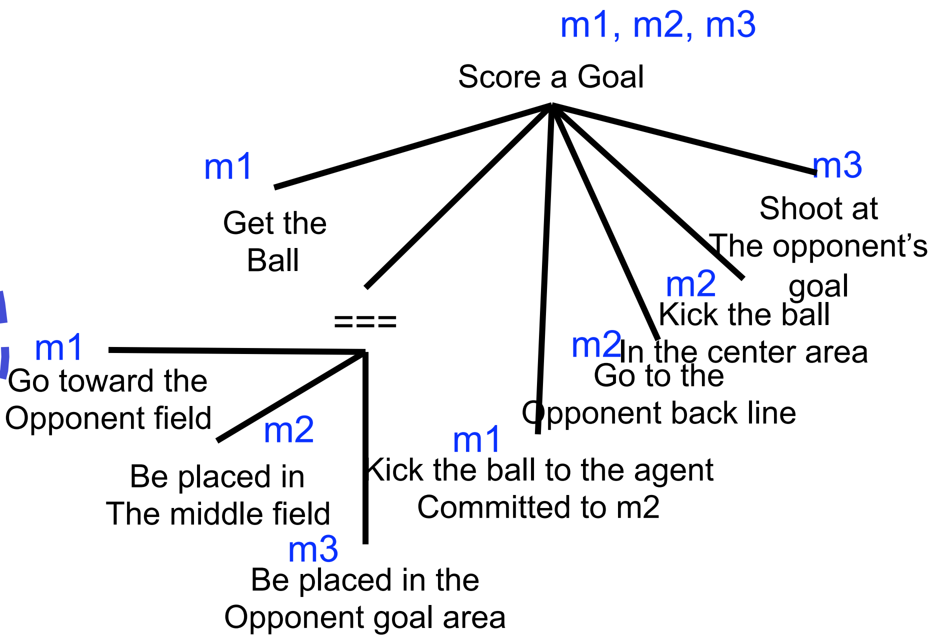
MOISE+ OML: Deontic Specification

- Explicit relation between the functional and structural specifications
  - Permissions and obligations to commit to missions in the context of a role
  - To make explicit the normative dimension of a role

Role	Deontic Relation	Mission	Temporal Constraint (cf. [carron 01])
Back	Permission	m1	In [0 30]
Middle	Obligation	m2	during [Attacker]
Attacker	Obligation	m3	Any

MOISE+ OML: Deontic Specification



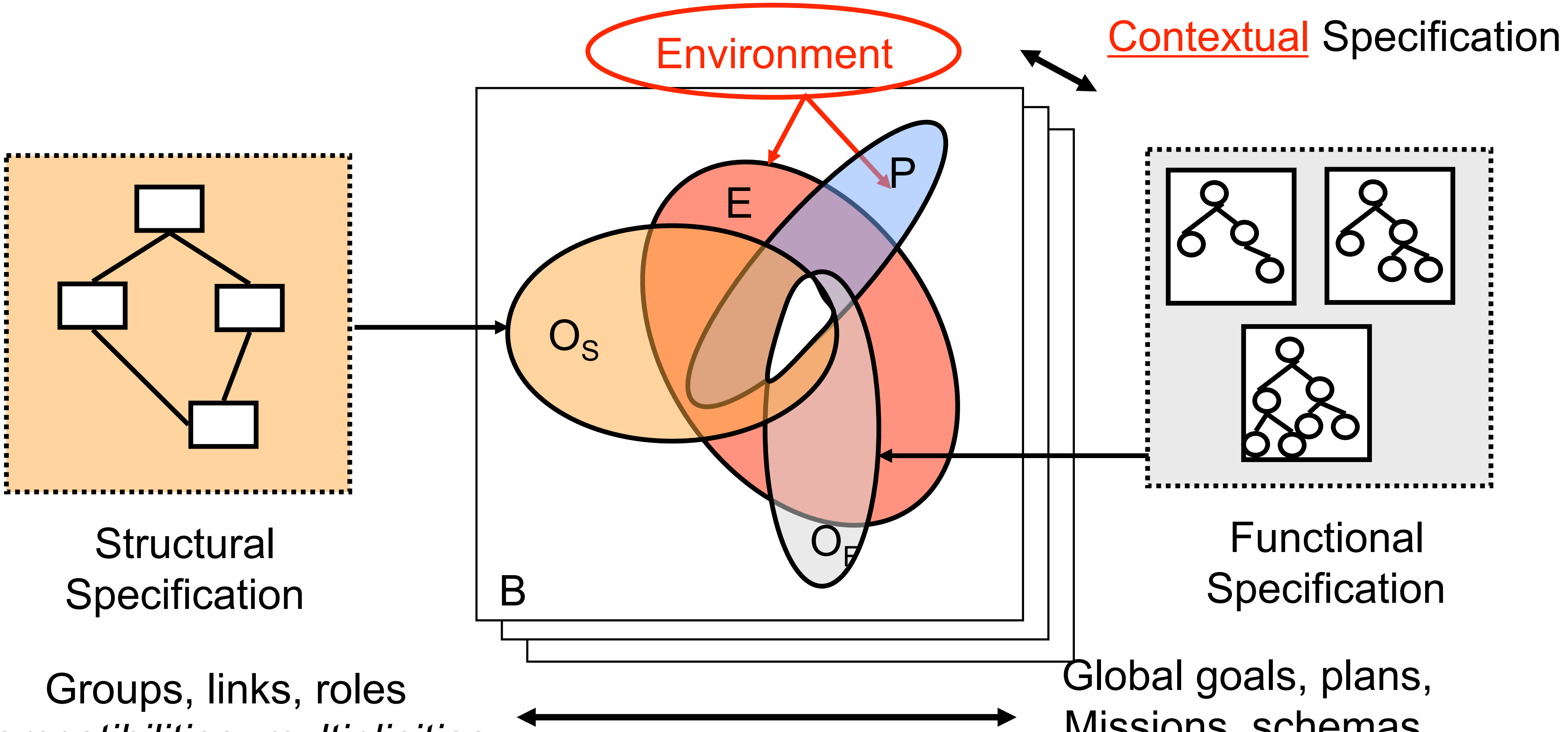


m1, m2, m3
Score a Goal
m1
Get the Ball
===
m1
Go toward the Opponent field
m2
Be placed in The middle field
m3
Be placed in the Opponent goal area
m1
Kick the ball to the agent Committed to m2
m2
Go to the Opponent back line
m2
Kick the ball In the center area
m3
Shoot at The opponent's goal

Organisation Entity
Lucio ----- m1
Cafu ----- m2
Rivaldo ----- m3

Graphical view of SoccerGame Application DS

MOISE^{inst} OML Dimensions [Gateau 07]



Contextual Specification

Structural Specification

Functional Specification

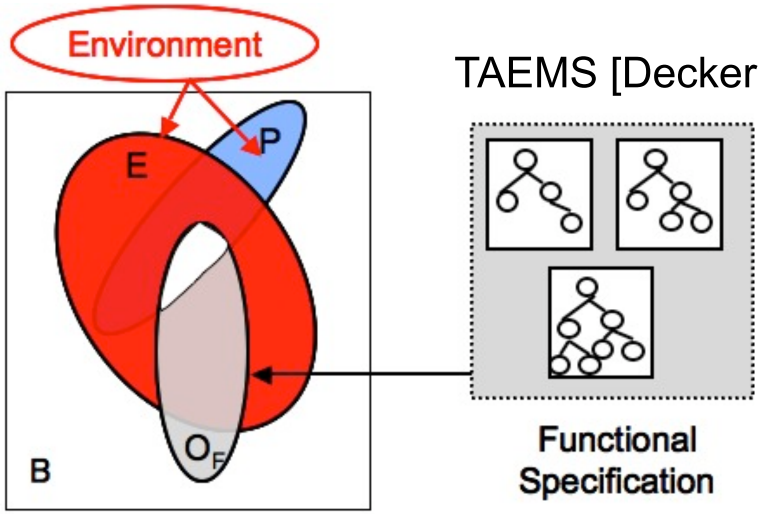
Groups, links, roles
Compatibilities, multiplicities inheritance

Normative Specification:
Permissions, Obligations

Global goals, plans,
Missions, schemas,
preferences

Landscape of existing OMLs

TAEMS [Decker 96]



Environment

E

P

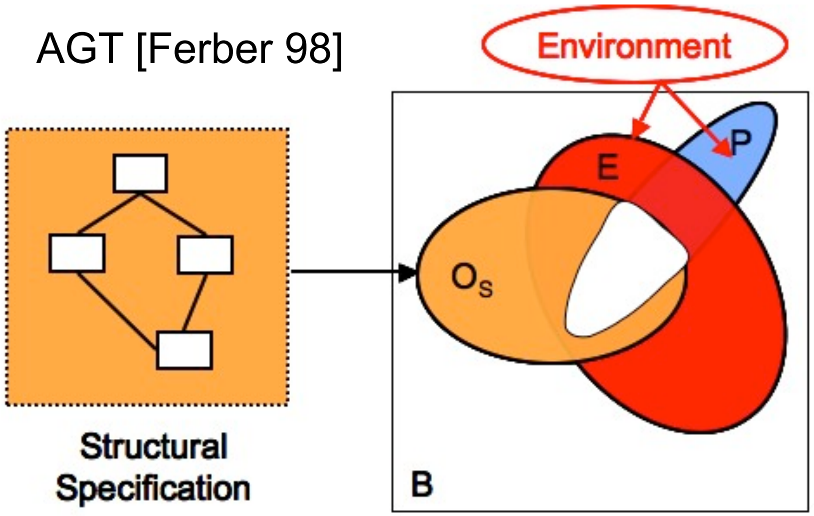
O_F

B

Functional Specification

a) O_F: agents' behaviors functionally constrained by the organization

AGT [Ferber 98]



Environment

E

P

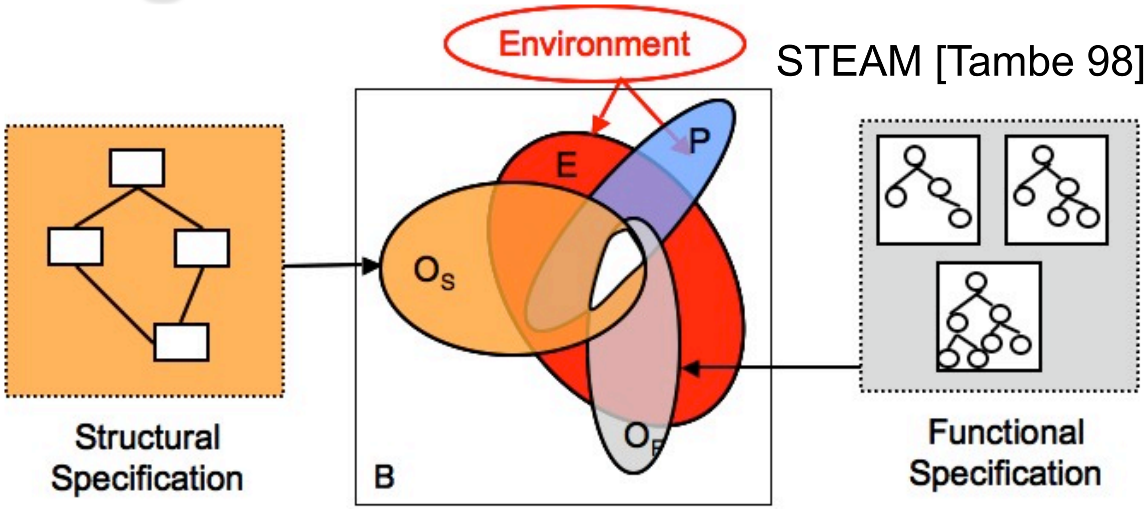
O_S

B

Structural Specification

b) O_S: agents' behaviors structurally constrained by the organization

STEAM [Tambe 98]



Structural Specification

Environment

E

P

O_S

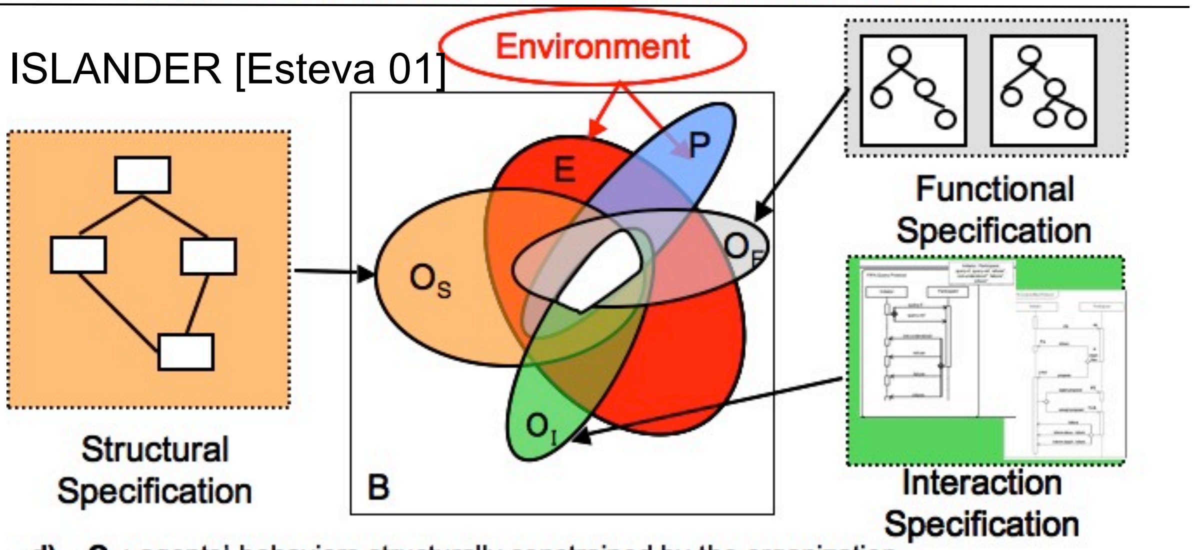
O_F

B

Functional Specification

c) O_S: agents' behaviors structurally constrained by the organization
O_F: agents' behaviors functionally constrained by the organization

ISLANDER [Esteva 01]



Environment

Functional Specification

Structural Specification

E

P

O_S

O_F

O_I

B

Interaction Specification

d) O_S: agents' behaviors structurally constrained by the organization
O_I: agents' behaviors interactionally constrained by the organization
O_F: agents' **permitted/obliged** behaviors functionally constrained by the organization

Outline

1. Introduction

2. Organisation Oriented Programming

3. Organisation Modeling Language

4. Organisation Infrastructure

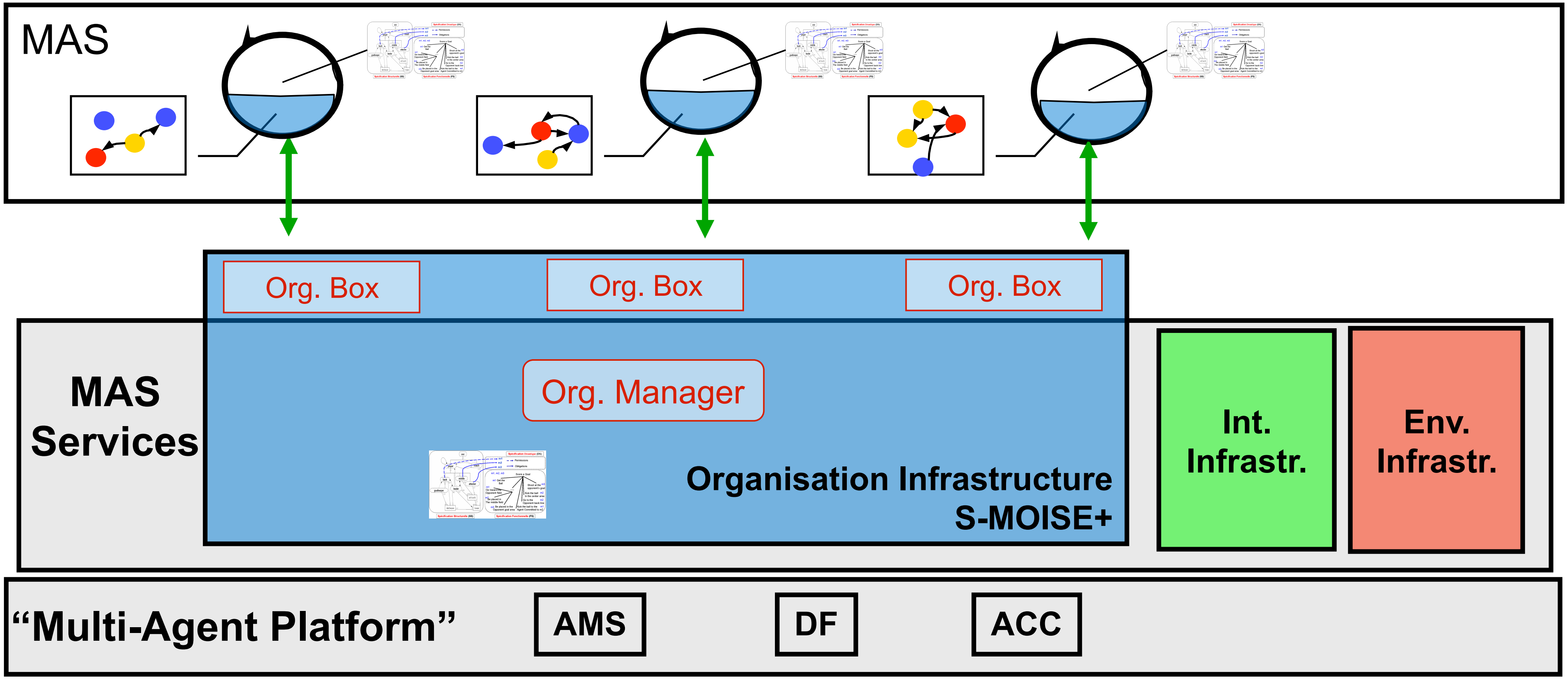
S-MOISE+, SYNAI, ORA4MAS

5. Conclusion and Perspectives

Motivations

- To help heterogeneous agents to dynamically participate in the organisation
  - Agents have to respect the organisation despite their architecture, their designer, …
- To ensure or enforce that the organisation is respected
  - Agents have to respect the organisation despite their architecture, their designer, …
- Interpretation of the OML
- Use of regimentation, enforcement, tools for cooperative tasks, …
- Allows agents to interact with the organisation (agent programming issues)

MOISE+ Organisation Infrastructure



- OML: MOISE+ [Hubner 02]
- OIA: S-MOISE+ [Hubner 05], J-MOISE+ [Hubner 06]

S-MOISE+ OrgBox Services

- Is attached to an agent and connected to the OrgManager

- Interface used by the agent:
  - to change the organisation entity (e.g. adopt a role)
  - to get the state of the organisation entity
    - However, only a personalised version of the entity is given from OrgManager to OrgBox, respecting the acquaintance relation
  - to send/receive a message to/from another agent
- Interface used by the OrgManager:
  - to notify the agent about every change in the state of a scheme to which the agent has committed to

- No particular agent architecture is required

S-MOISE+ OrgManager

- Maintains the current state of the organisation 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 organisation specification
- Receives messages from the other agents' OrgBoxes asking for changes in the organisation entity/ specification
- Regiments some norms (cardinality, compatibility …)

S-MOISE+ Organisation Events

- Organisation life-cycle
  - Entrance/Exit of an agent
  - Creation/Deletion of an Organisation entity
  - Change of Organisation specification
- Structural Organisation life-cycle
  - Creation/Deletion of a group
  - Adoption/Release of a role by an agent
- Functional Organisation life-cycle
  - Creation/End of a schema
  - Commitment/Release of a mission by an agent
  - Change of a global goal state
- Obligation life-cycle
  - Start/end of obligation

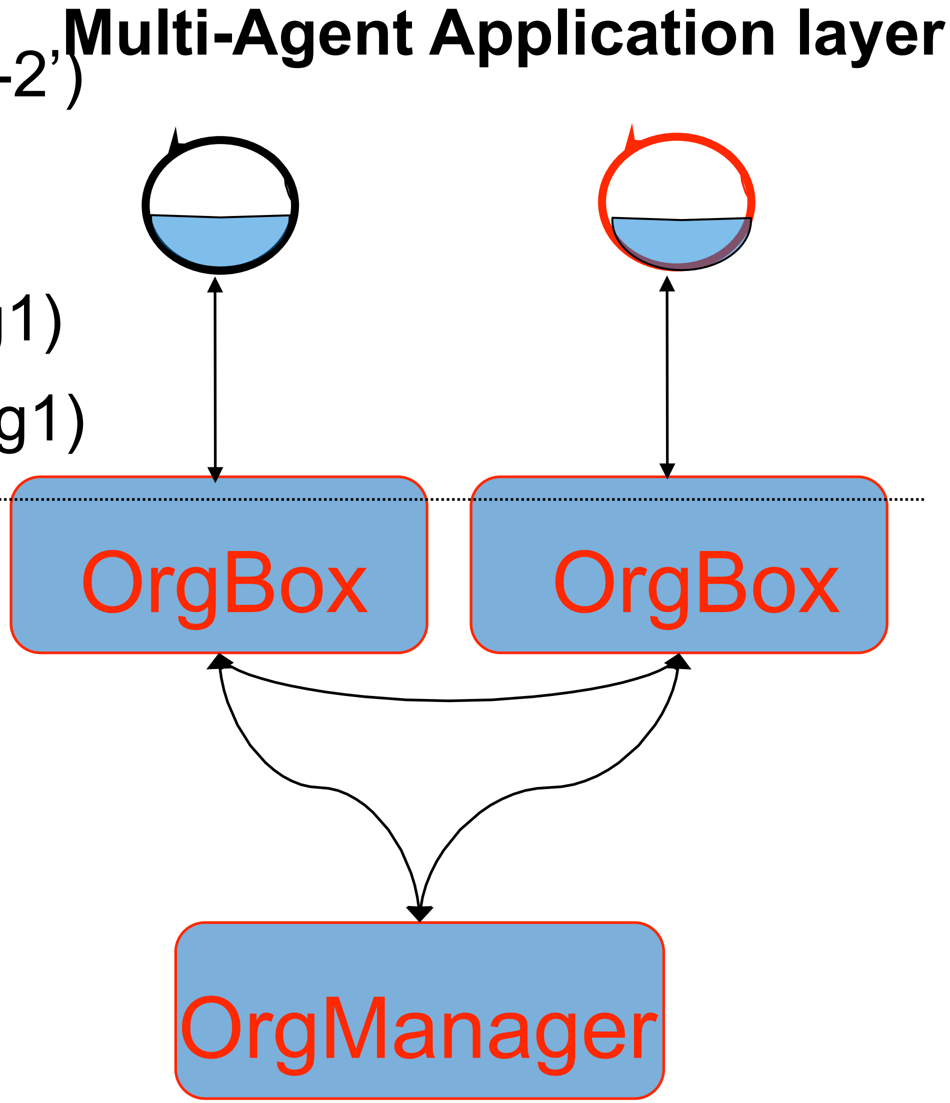
S-MOISE+ example

1. oe1=createOE('score a goal', '3-5-2')

2. g1=createGroup('team',oe1)

3. g1-1=createSubgroup('defense', g1)

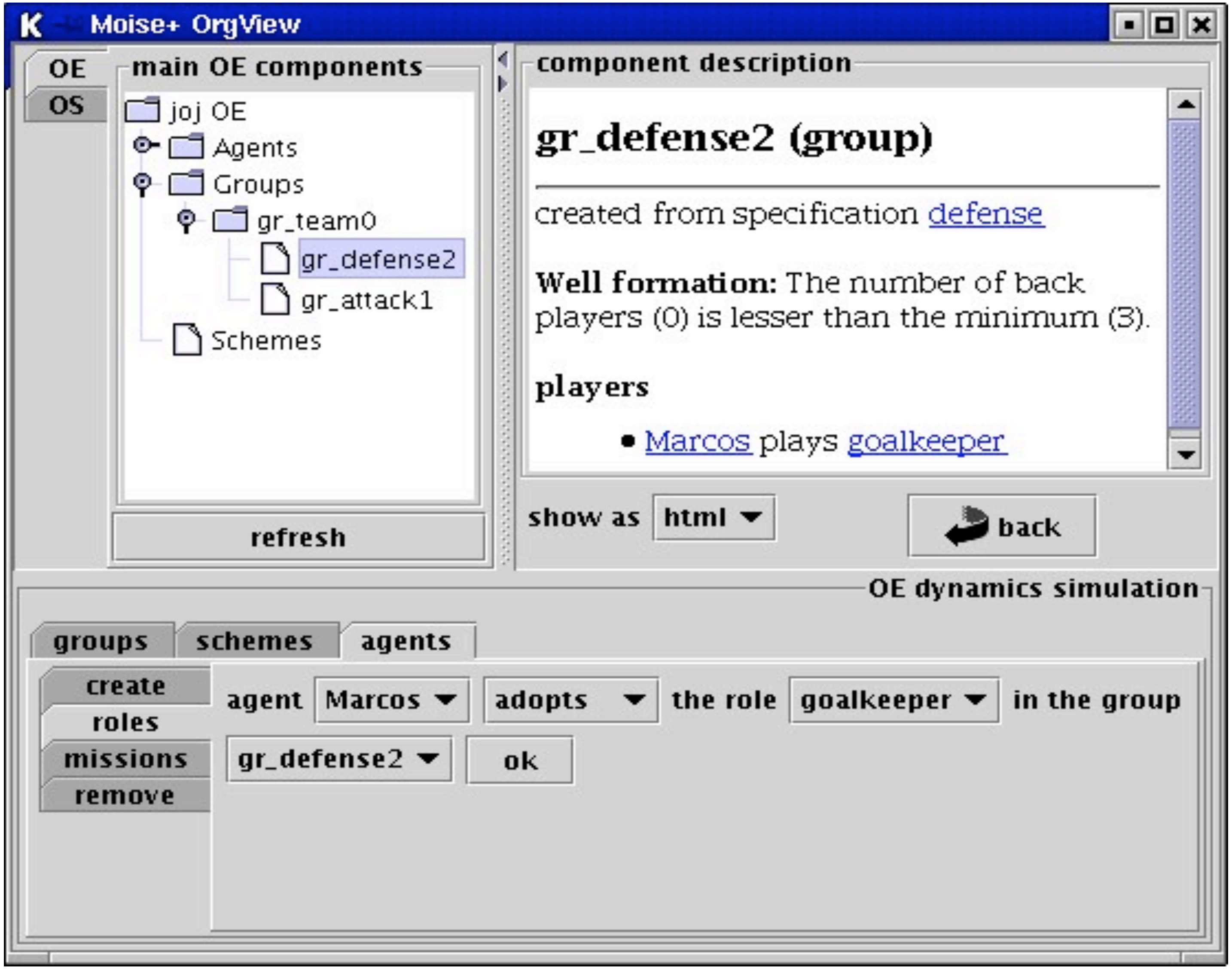
4. sc1=createScheme('side_attack', g1)



S-MOISE+: Org. Manager View

Organizational Entity after the events

The defense group is not **well formed** since there is no agent playing ‘back’ role



gr_defense2 (group)

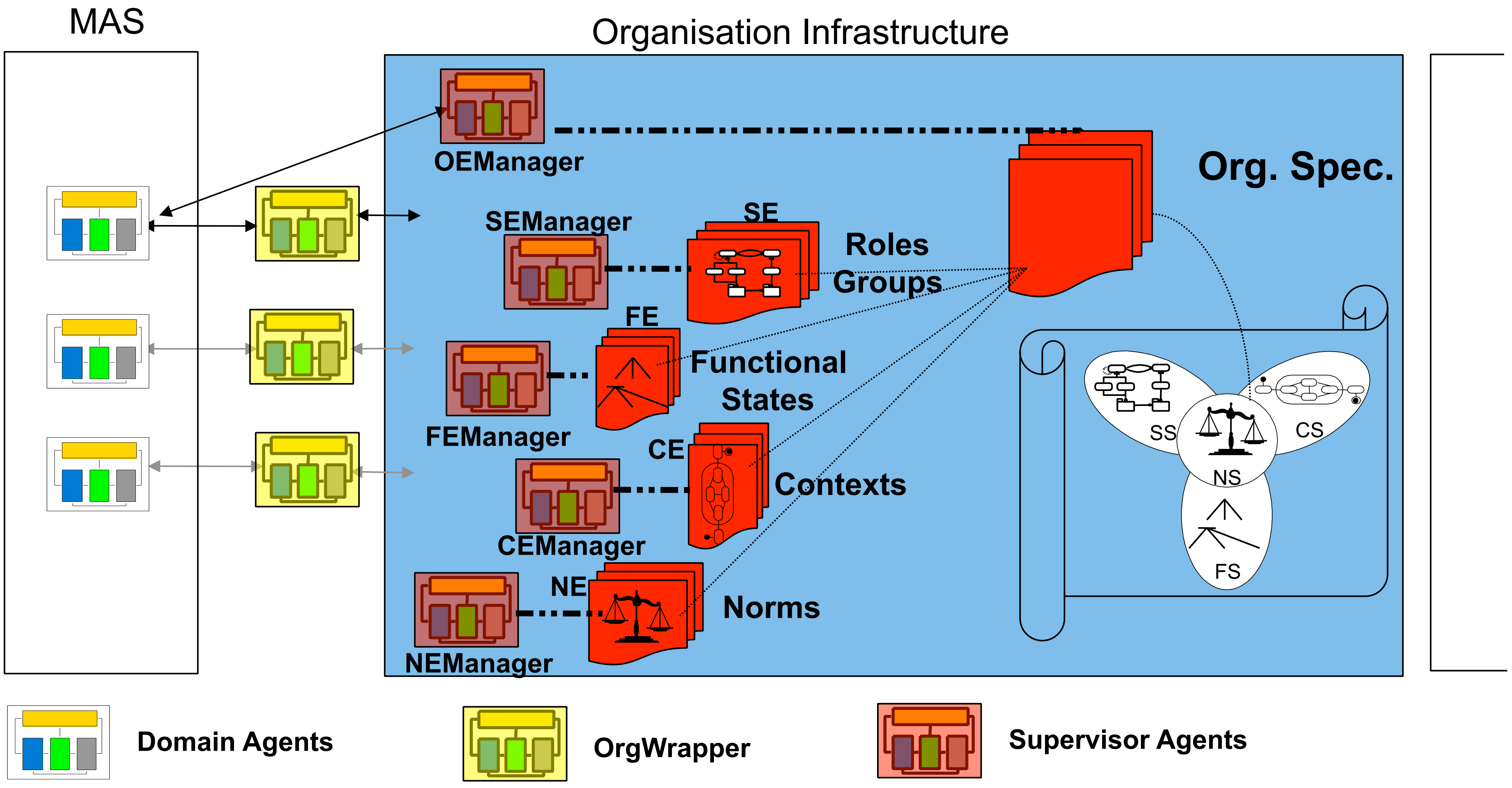
created from specification defense

Well formation: The number of back players (0) is lesser than the minimum (3).

players

- Marcos plays goalkeeper

MOISE^{inst}: Organisation Infrastructure [Gateau 07]

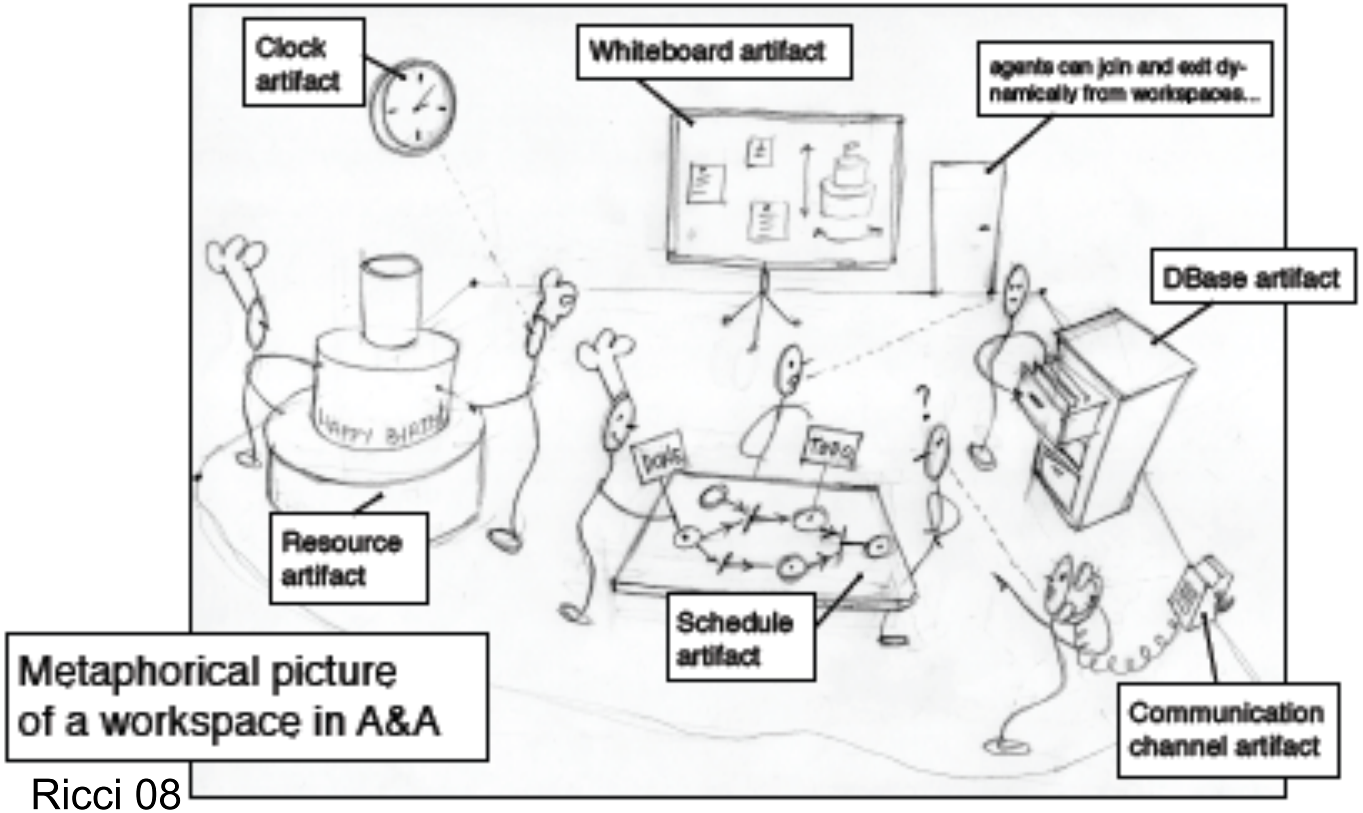


ORA4MAS [Kitio 07]: Motivations

- Regimentations vs Enforcements in Multi-Agent Organisations
  - Regimentations prevent agents' autonomy
    - e.g. ''*the access to computers requires an username*'', "roles which cardinality is full cannot be adopted, agents cannot adopt incompatible roles"
  - Norms regulate agents' autonomy
    - e.g. ''*a master thesis should be written in two years*'' "based on its roles, an agent is obliged to commit to certain missions, based on its missions, an agent is obliged to achieve some goals"
- Limitations of current approaches:
  - Management of regimentations and norms realised by mechanisms in dedicated services/agents within the agent platform
  - Organisation is half on its way of being a first class entity in MAS!!!
- Requirements:
  - "Give the power back to the agents'', i.e. decision on Norms violation should be realised within agents and no more in the agent platform
  - Distributed mediation between agents to access the organisation state and to cooperate for norm management and enforcement

ORA4MAS: Foundations

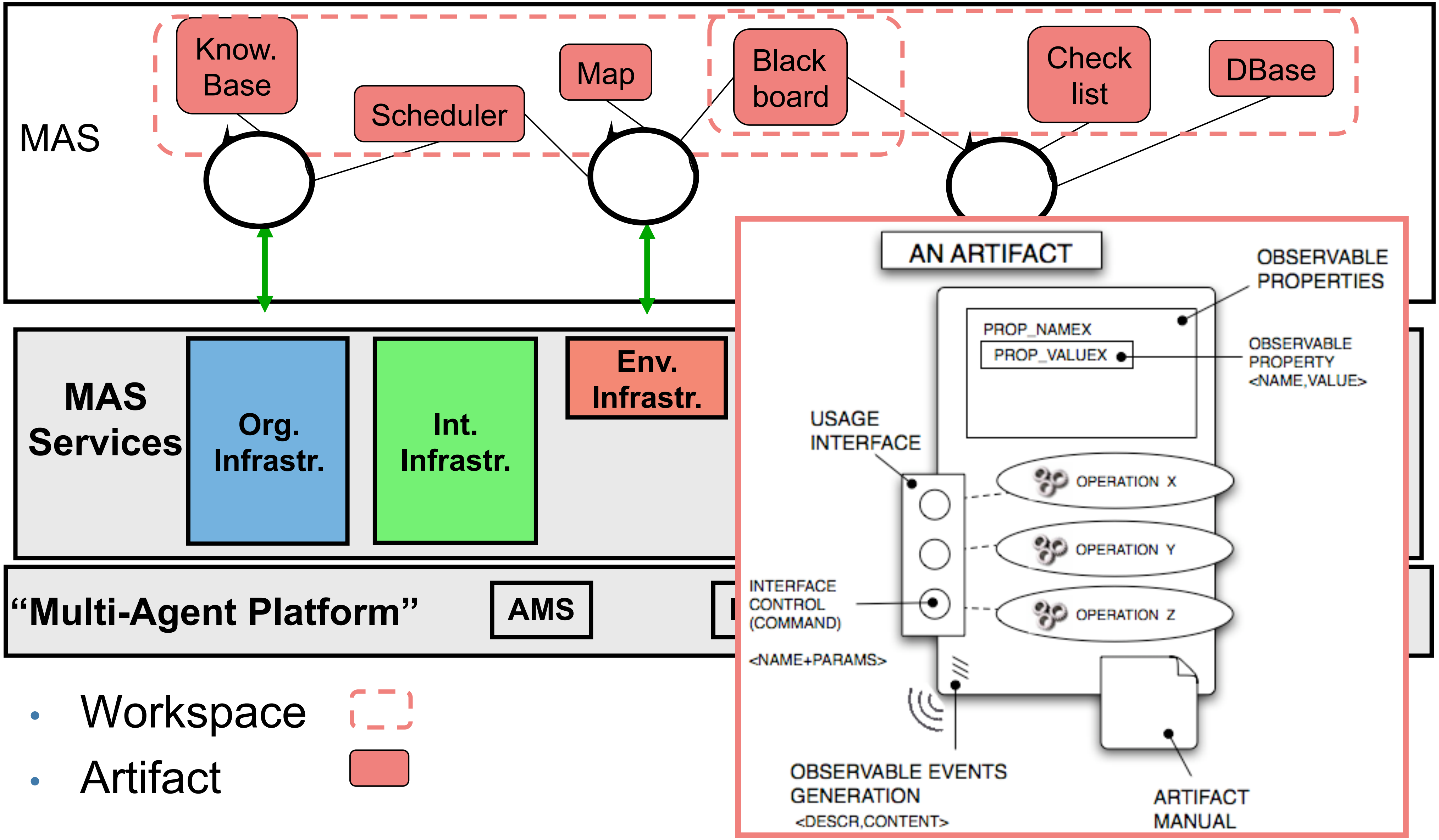
- Artifacts, Agents, Workspaces [Ricci 07]
- Background in Activity Theory and Distributed Cognition



Metaphorical picture of a workspace in A&A

Ricci 08

ORA4MAS: Foundations

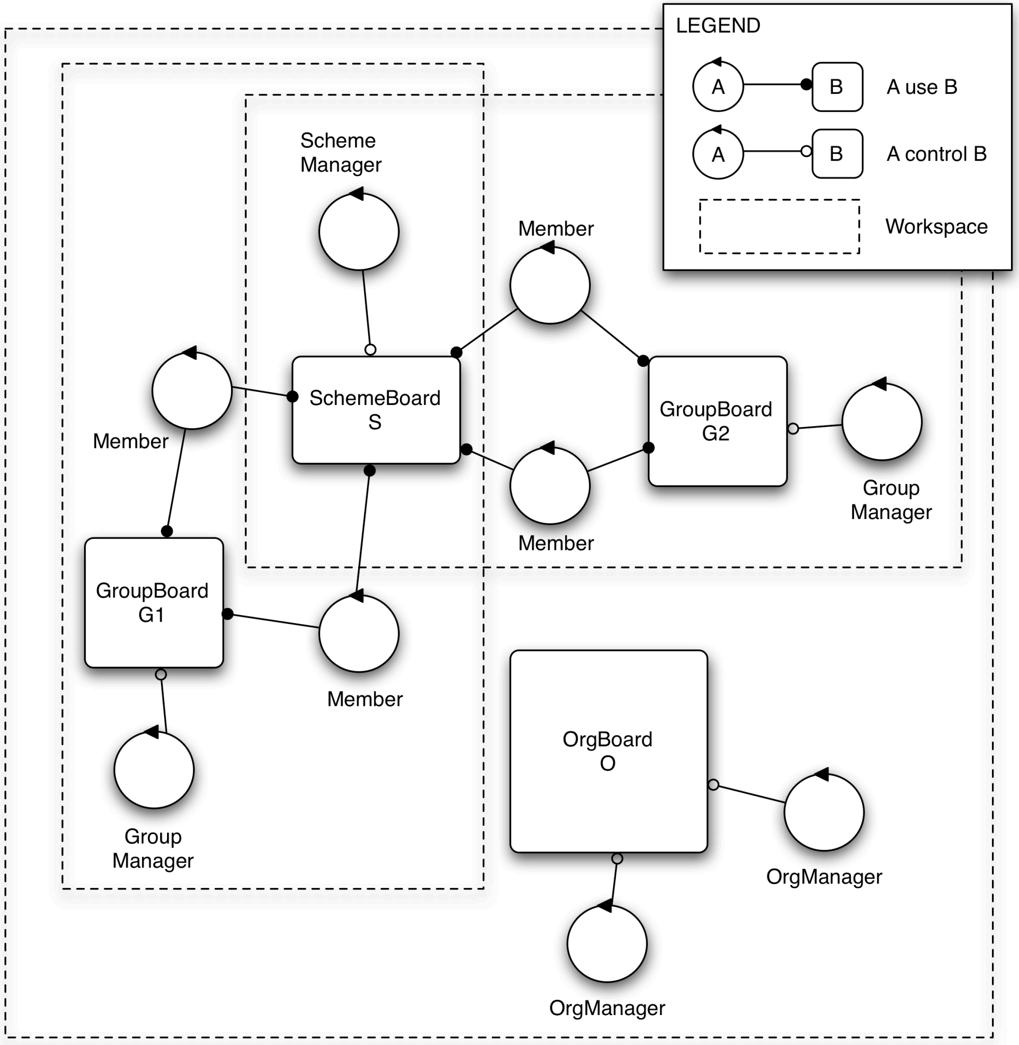


- Workspace ⬚
- Artifact 🟥

First class entities for Organisation entity

- Organisational Artifacts:
  - Encapsulate and distribute the management of the organisation
  - Are in charge of:
    - enactment of regimentations
    - evaluation of norms compliance
  - Mediate agent interaction within the organisation

- Agents:
  - Try to achieve their goals
  - Use/control the organisational artifacts: i.e observe, monitor their state, reason and decide about the organisation (sanctions, …)



ORA4MAS Global view

1. The agents' behaviours within the organisation is supported and mediated by the organisational artifacts with respect to the roles that the agents play.
   - Violation of regimentations are prevented
   - Non compliance to norms are detected

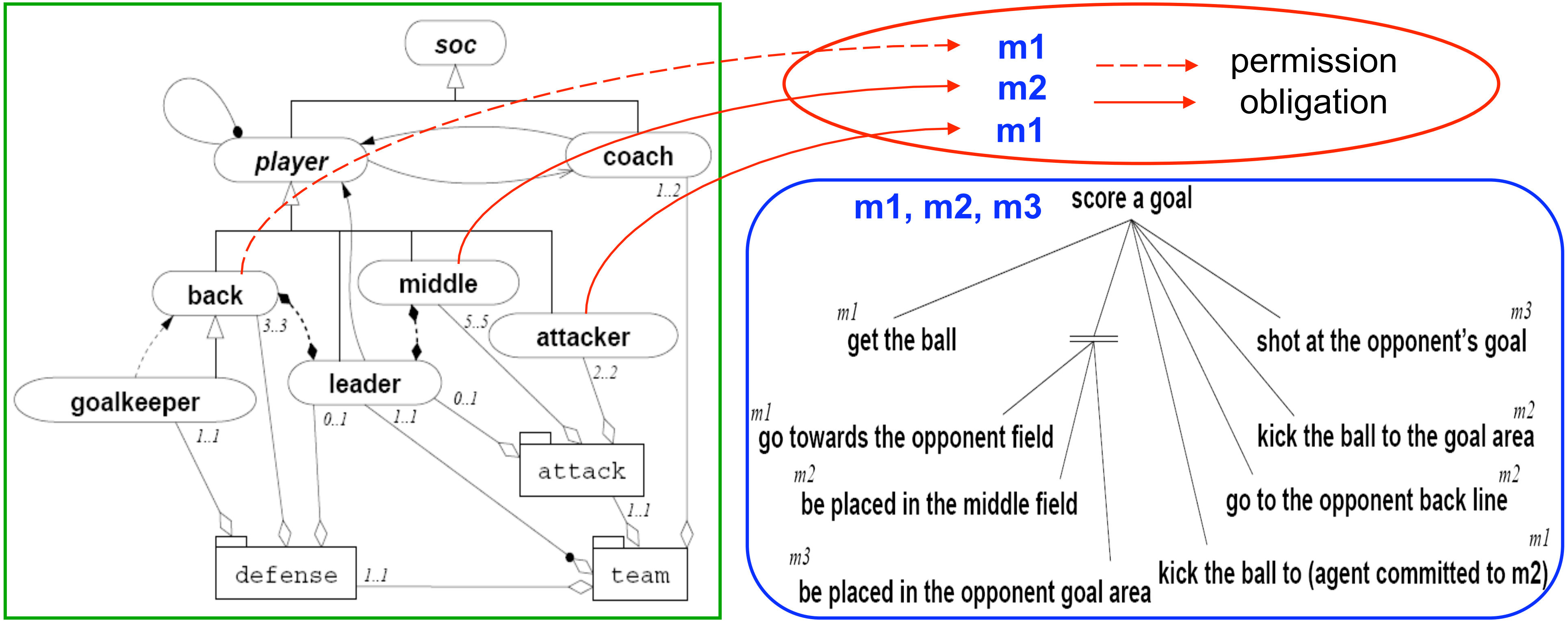
   Organisation state, status of norms is published/made accessible to the agents through the artifacts

2. Using this information,
   - agents decide about norm violation and corresponding sanctions to apply
   - agents take care/or not of the sanctions and behave accordingly or not

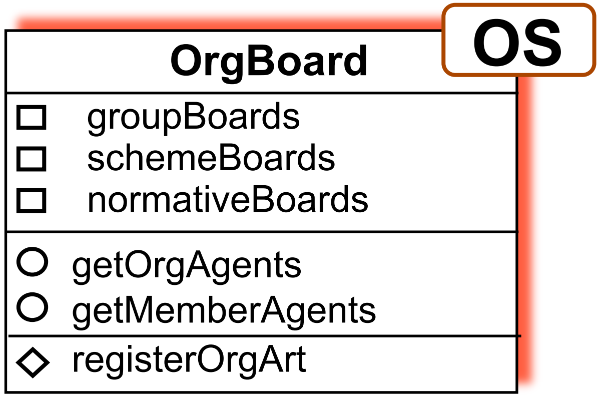
- Our current work is currently focused on point 1.

Shaping OR4MAS on MOISE+ OML

- Organisation is specified along three dimensions:
  - Structural Specification (SS) : groups, roles, …
  - Functional Specification (FS): social schemes, missions, goals, …
  - Deontic Specification (DS): obligations, permissions


soc
player
coach
1..2
back
middle
attacker
3..3
5..5
2..2
goalkeeper
leader
0..1
0..1
1..1
1..1
attack
1..1
defense
1..1
team
m1
m2
m1
permission
obligation
m1, m2, m3
score a goal
m1
get the ball
m3
shot at the opponent's goal
m1
go towards the opponent field
m2
kick the ball to the goal area
m2
be placed in the middle field
m2
go to the opponent back line
m3
be placed in the opponent goal area
m1
kick the ball to (agent committed to m2)

Organisational Artifacts for Moise+



OS

OrgBoard

□ groupBoards
□ schemeBoards
□ normativeBoards

○ getOrgAgents
○ getMemberAgents

◇ registerOrgArt

OrgBoard:

Management of the entry/exit/ access to the organisation

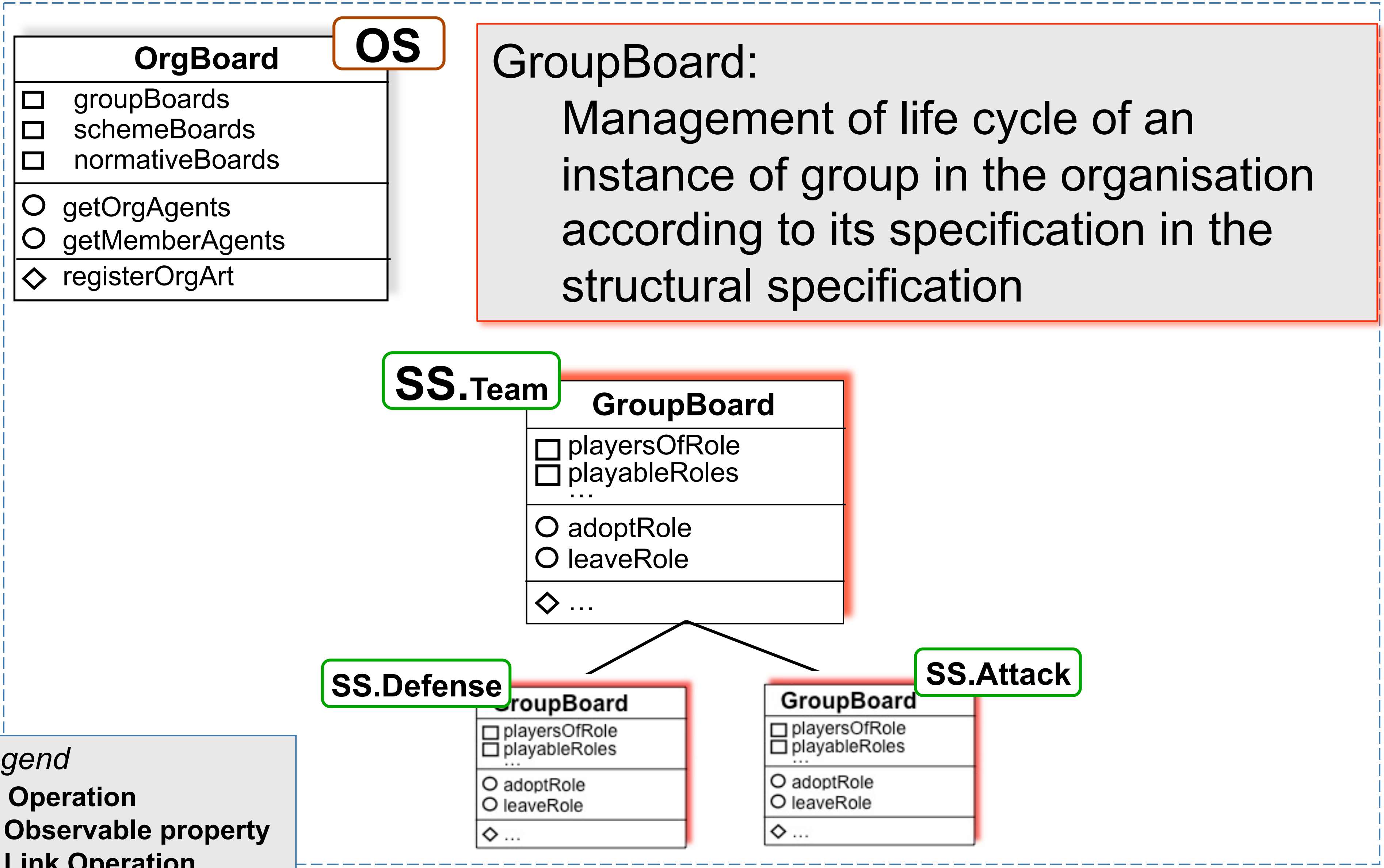
Legend

○ **Operation**
□ **Observable property**
◇ **Link Operation**

Organisational Artifacts for Moise+

GroupBoard:

Management of life cycle of an instance of group in the organisation according to its specification in the structural specification



Legend

○ **Operation**

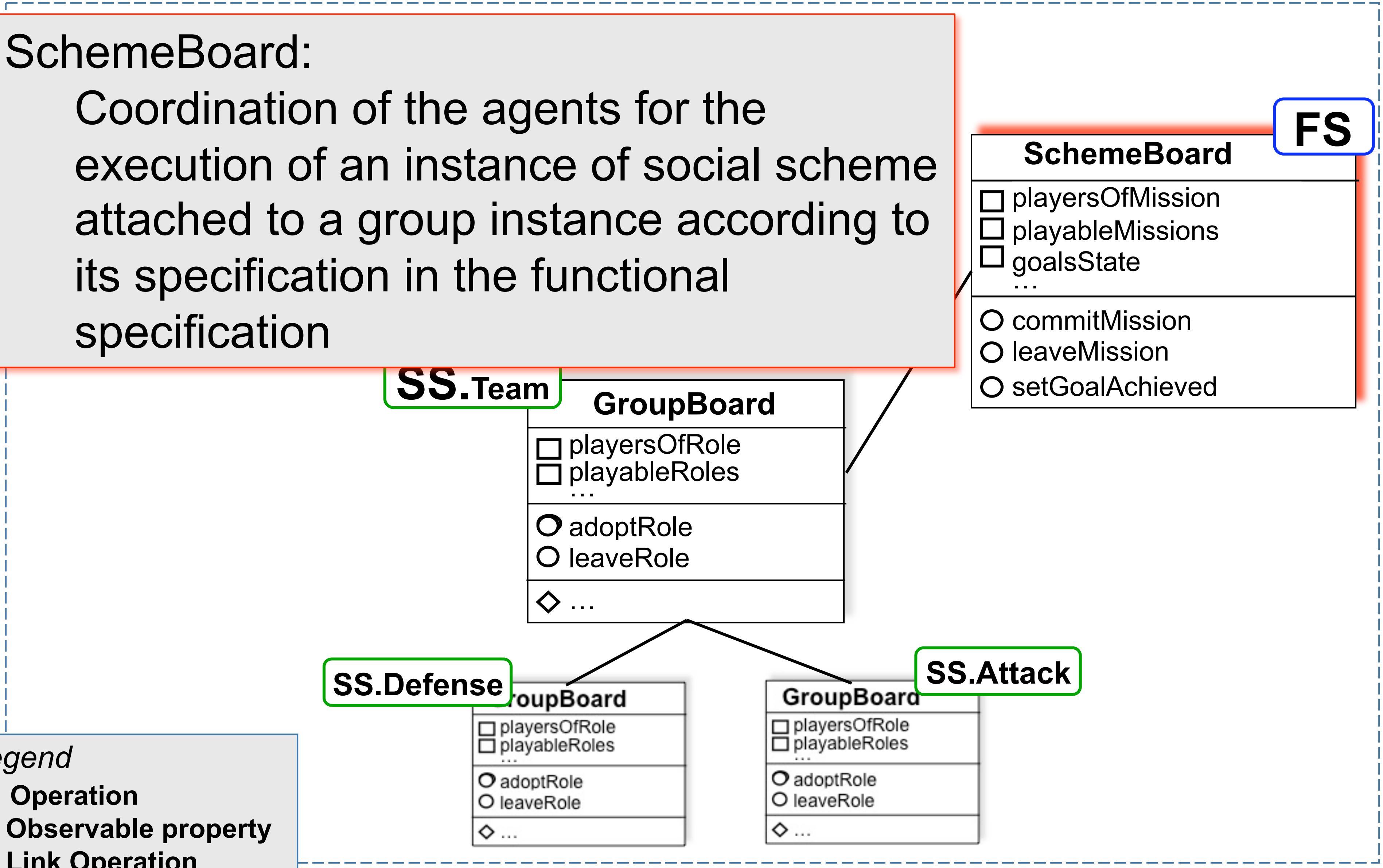
□ **Observable property**

◇ **Link Operation**

Organisational Artifacts for Moise+

SchemeBoard:

Coordination of the agents for the execution of an instance of social scheme attached to a group instance according to its specification in the functional specification



FS

SchemeBoard

□ playersOfMission
□ playableMissions
□ goalsState
...

○ commitMission
○ leaveMission
○ setGoalAchieved

SS.Team

GroupBoard

□ playersOfRole
□ playableRoles
...

○ adoptRole
○ leaveRole

◇ ...

SS.Defense

GroupBoard

□ playersOfRole
□ playableRoles
...

○ adoptRole
○ leaveRole

◇ ...

SS.Attack

GroupBoard

□ playersOfRole
□ playableRoles
...

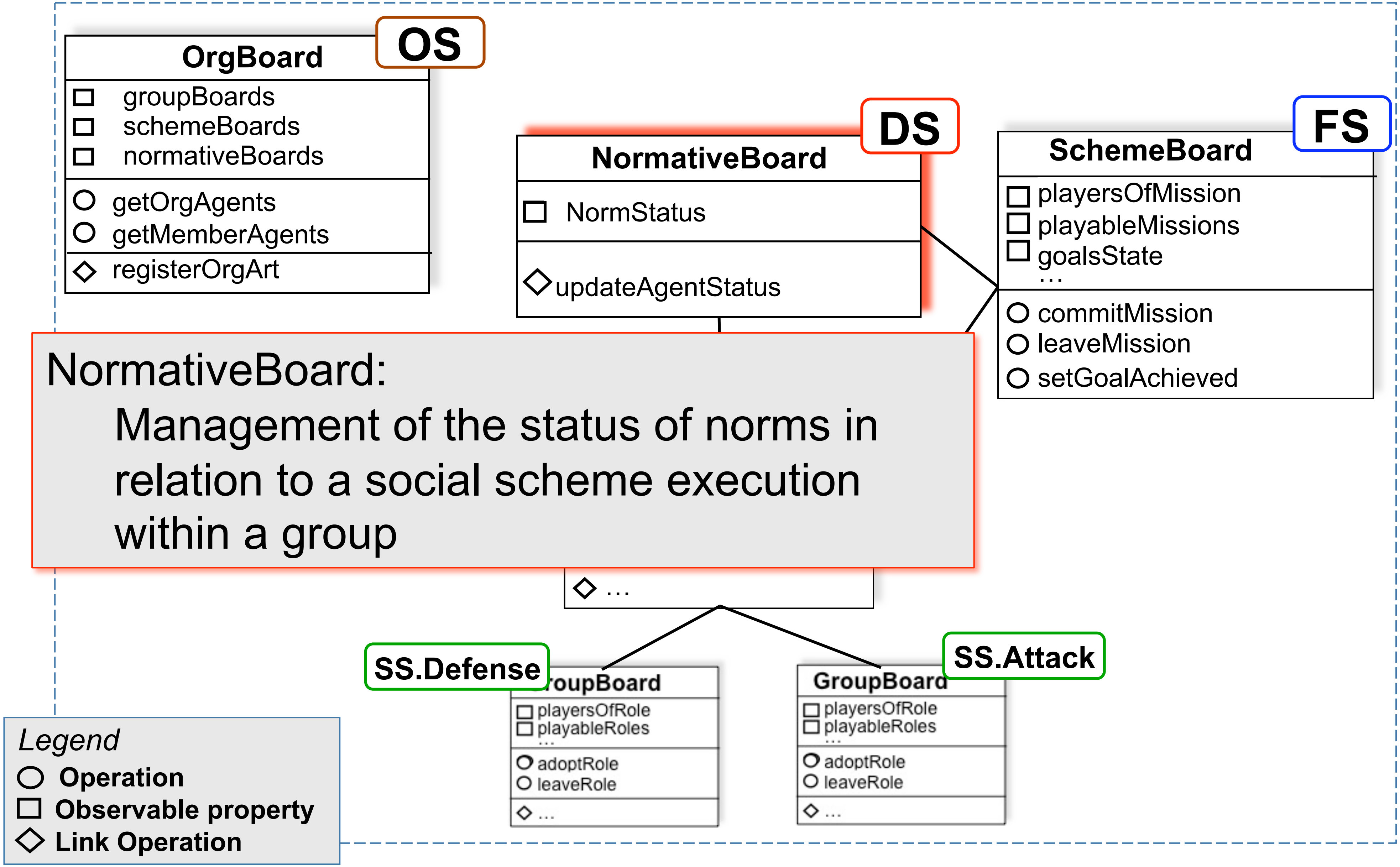
○ adoptRole
○ leaveRole

◇ ...

Legend

○ **Operation**
□ **Observable property**
◇ **Link Operation**

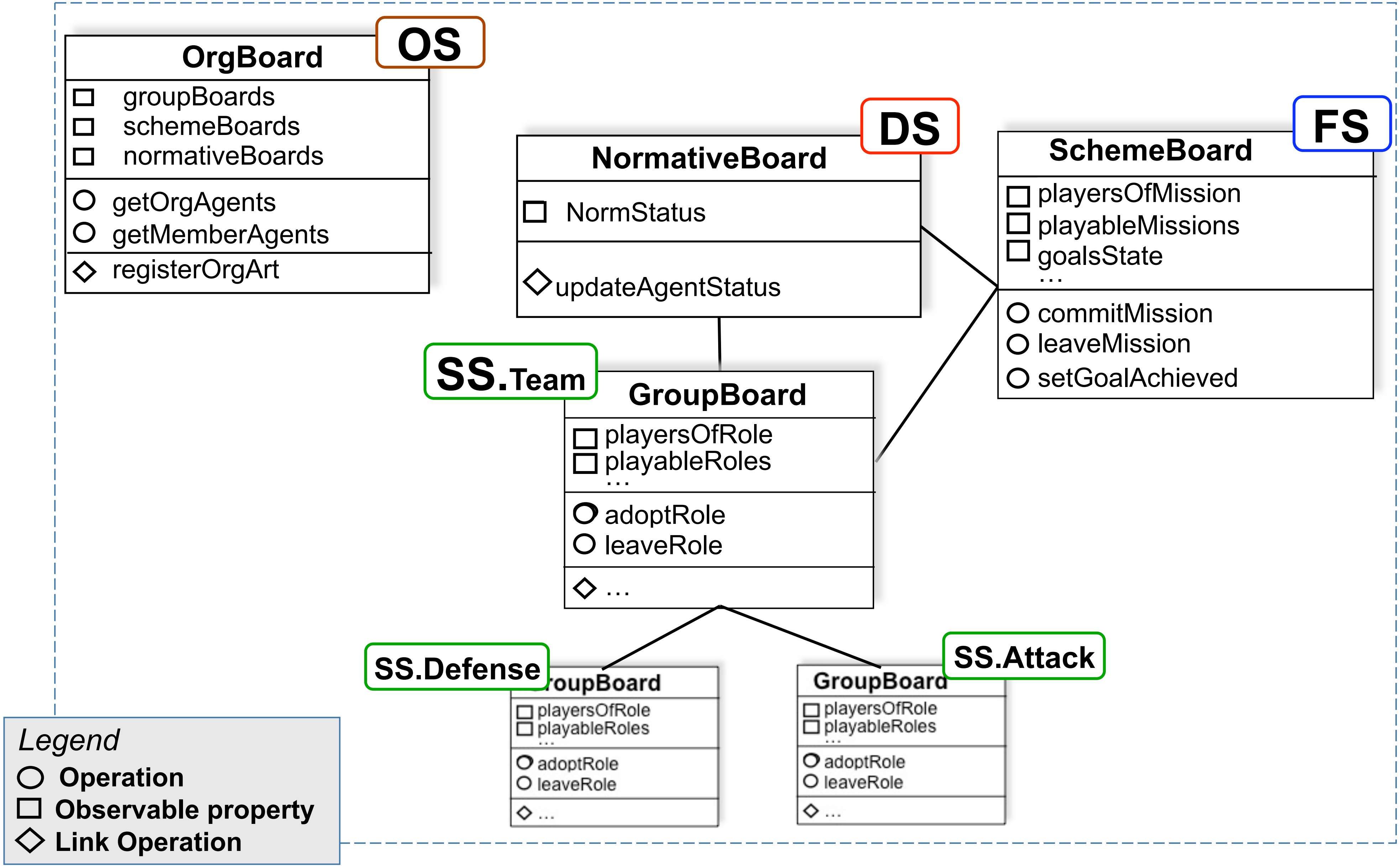
Organisational Artifacts for Moise+



NormativeBoard:

Management of the status of norms in relation to a social scheme execution within a group

Organisational Artifacts for Moise+



OS

OrgBoard
- □ groupBoards
- □ schemeBoards
- □ normativeBoards
- ○ getOrgAgents
- ○ getMemberAgents
- ◇ registerOrgArt

DS

NormativeBoard
- □ NormStatus
- ◇ updateAgentStatus

FS

SchemeBoard
- □ playersOfMission
- □ playableMissions
- □ goalsState
- …
- ○ commitMission
- ○ leaveMission
- ○ setGoalAchieved

SS.Team

GroupBoard
- □ playersOfRole
- □ playableRoles
- …
- ○ adoptRole
- ○ leaveRole
- ◇ …

SS.Defense

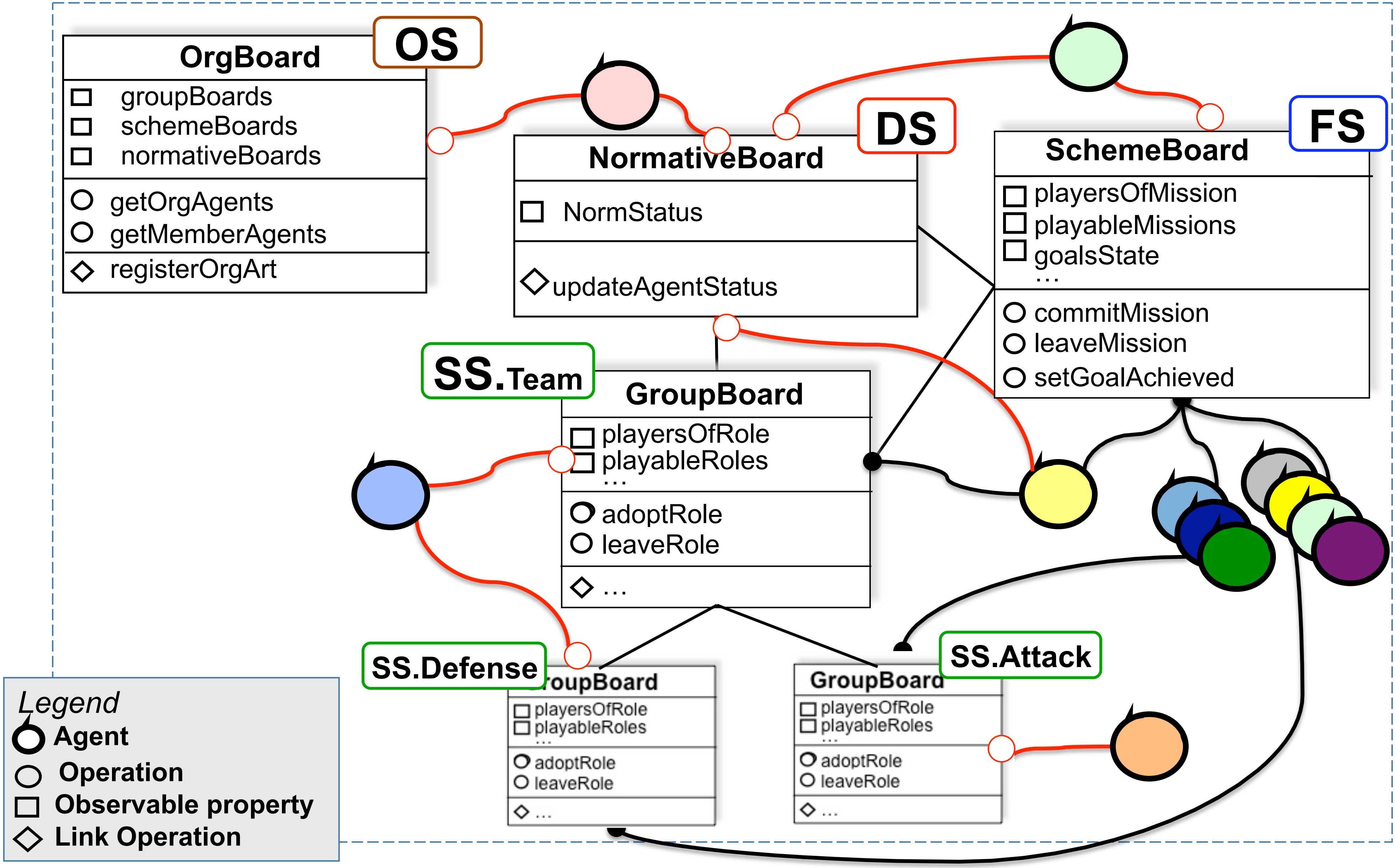
roupBoard
- □ playersOfRole
- □ playableRoles
- …
- ○ adoptRole
- ○ leaveRole
- ◇ …

SS.Attack

GroupBoard
- □ playersOfRole
- □ playableRoles
- …
- ○ adoptRole
- ○ leaveRole
- ◇ …

Legend
- ○ **Operation**
- □ **Observable property**
- ◇ **Link Operation**

Agents within ORA4MAS for MOISE+



Example scenario (1/2)

1. Using the Organisation Specification, an agent:
   a) Creates the OrgBoard for the corresponding organisation
   b) Creates the different GroupBoards corresponding to the OS (e.g. Team, Defense, Attack). GroupBoards register in the OrgBoard
2. OrgBoard and GroupBoards are perceived by the agents.
3. Agents adopt roles according to their preferences and to the OS by using the operation adoptRole of the corresponding GroupBoard
4. Maximal number of roles is reached: operation adoptRole is disabled (i.e. way of enacting regimentation)
5. Since groups are wellformed, an agent decides to create the SchemeBoard to start the process.
6. The created SchemeBoard is registered in the OrgBoard.
7. Creation of the corresponding NormativeBoard

Example scenario (2/2)

8. SchemeBoard acts as a coordination artifacts showing which goals are achieved, the ones that are still to be achieved and that are possibles, the ones that are not yet possibles, …
9. Obligations and permissions are computed and verified by the NormativeBoard
   - Use of information about members of the groups and their commitments to missions and updating of goal status in SchemeBoard
   - In case of non achievement of goals while being obliged, norm non compliance is detected
10. An agent may use this information to decide that it is a violation of the norms by the corresponding agents and may decide to apply sanctions according to its own strategies
11. …

Outline

1. Introduction
2. Organisation Oriented Programming
3. Organisation Modeling Language
4. Organisation Infrastructure
5. **Conclusion and Perspectives**

Conclusion and perspectives

- Status
  - First prototype of ORA4MAS/MOISE+ implemented on top of Cartago
  - Experiments under realisation with Jason Agents and Jadex Agents
  - Future experiments with 2APL Agents

  - Organisation is no more outside of the MAS
  - Orwell’s nightmare cannot become reality ;-)

- Where we have to go - Further works and questions
  - Other organisational artifacts? (e.g. ‘‘Reputation’’ artifact [Hubner 08])
  - How to link organisational artifacts with ‘‘cognitive artifacts’’ (i.e. tools, resources, …)?
  - Situated organisation? Situated organisational artifacts?
  - RBAC on the organisational artifacts
  - Organisational agents

References

- [Grossi 07] D. Grossi, H. Aldewered, and F. Dignum. Ubi Lex, Ibi Poena : Designing norm enforcement in e-institutions. In P. Noriega, J. Vasquez-Salceda, G. Boella, O. Boissier, V. Dignum, N. Fornara, and E. Matson, editors, Coordination, Organizations, Institutions, and Norms in Agent Systems II, volume 4386 of LNAI, pages 101–114. Springer, 2007. Revised Selected Papers.
- [Hannoun 00] M. Hannoun, O. Boissier, J. S. Sichman, and C. Sayettat, Moise : An organizational model for multi-agent systems In M.C. Monard and J.S. Sichman, editors, Advances in Artificial Intelligence, volume 1952 of LNAI, pages 156—165, Springer, 2000. International Joint Conference IBERAMIA-SBIA
- [Hubner 02] J. F. Hubner, J. S. Sichman, and O. Boissier. Using the MOISE+ for a cooperative framework of MAS reorganisation. In A. L. C. Bazzan and S. Labidi, editors, Proceedings of the 17th Brazilian Symposium on Artificial Intelligence (SBIA'04), volume 3171 of LNAI, pages 506–515, Berlin, 2004. Springer.
- [Hubner 05] J. F. Hubner, J. S. Sichman, and O. Boissier. S-MOISE+: A middleware for developing organised multi-agent systems. In O. Boissier, V. Dignum, E. Matson, and J. S. Sichman, editors, Proceedings of the International Workshop on Organizations in Multi-Agent Systems, from Organizations to Organization Oriented Programming in MAS (OOOP'2005), volume 3913 of LNAI, pages 64–78. Springer, 2006.
- [Gateau 05] Gâteau, B., Boissier, O., Khadraoui, D., Dubois, E.: MOISE-Inst : An organizational model for specifying rights and duties of autonomous agents. In van der Torre, L., Boella, G., eds.: 1st International Workshop on Coordination and Organisation (CoOrg 2005) affiliated with the 7th International Conference on Coordination Models and Languages, Namur - Belgium (2005)
- [Hubner 07] J. F. Hubner, J. S. Sichman, and O. Boissier. Developing organised multi-agent systems using the MOISE+ model: Programming issues at the system and agent levels. International Journal of Agent-Oriented Software Engineering, 1(3/4):370– 395, 2007.
- [Ricci 07] A. Ricci, M. Viroli, and A. Omicini. The A&A programming model & technology for developing agent environments in MAS. In M. Dastani, A. El Fallah Seghrouchni, A. Ricci, and M. Winikoff, editors, Post-proceedings of the 5th International Workshop "Programming Multi-Agent Systems" (PROMAS 2007), volume 4908 of LNAI, pages 91–109. Springer, 2007.
- [Kitio 07] R. Kitio, O. Boissier, J. F. Hubner, and A. Ricci. Organisational artifacts and agents for open multi-agent organisations: "giving the power back to the agents". In J. Sichman, P. Noriega, J. Padget, and S. Ossowski, editors, Coordination, Organizations, Institutions, and Norms in Agent Systems III, volume 4870 of LNAI, pages 171–186. Springer, 2008. Revised Selected Papers.
- [Hubner 08] Jomi Fred Hubner, Laurent Vercouter, and Olivier Boissier, Instrumenting Multi-Agent Organisations with Reputation Artifacts, COIN Workshop at AAAI 08.