
S-MOISE+:

A Middleware for developing Organised Multi-Agent Systems

Jomi F. Hübner, Jaime S. Sichman, O.Boissier

University of Blumenau, University of São Paulo,
École Mines Saint-Etienne

Context

- **Organisation for MAS**
 - Constraints for **open systems**
 - Constraints for agents' autonomy
- **Main approaches**
 - Agent centered
 - Organisation centered
- **We already have many organisational models**
 - TAEMS, STEAM
 - AGR, TOVE
 - ISLANDER, OPERA
 - MOISE, MOISE+
 - ...

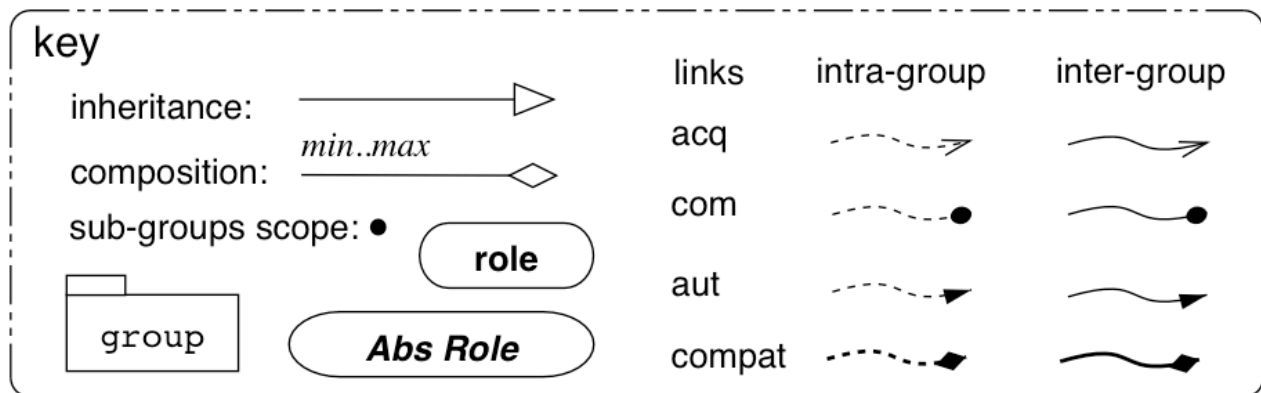
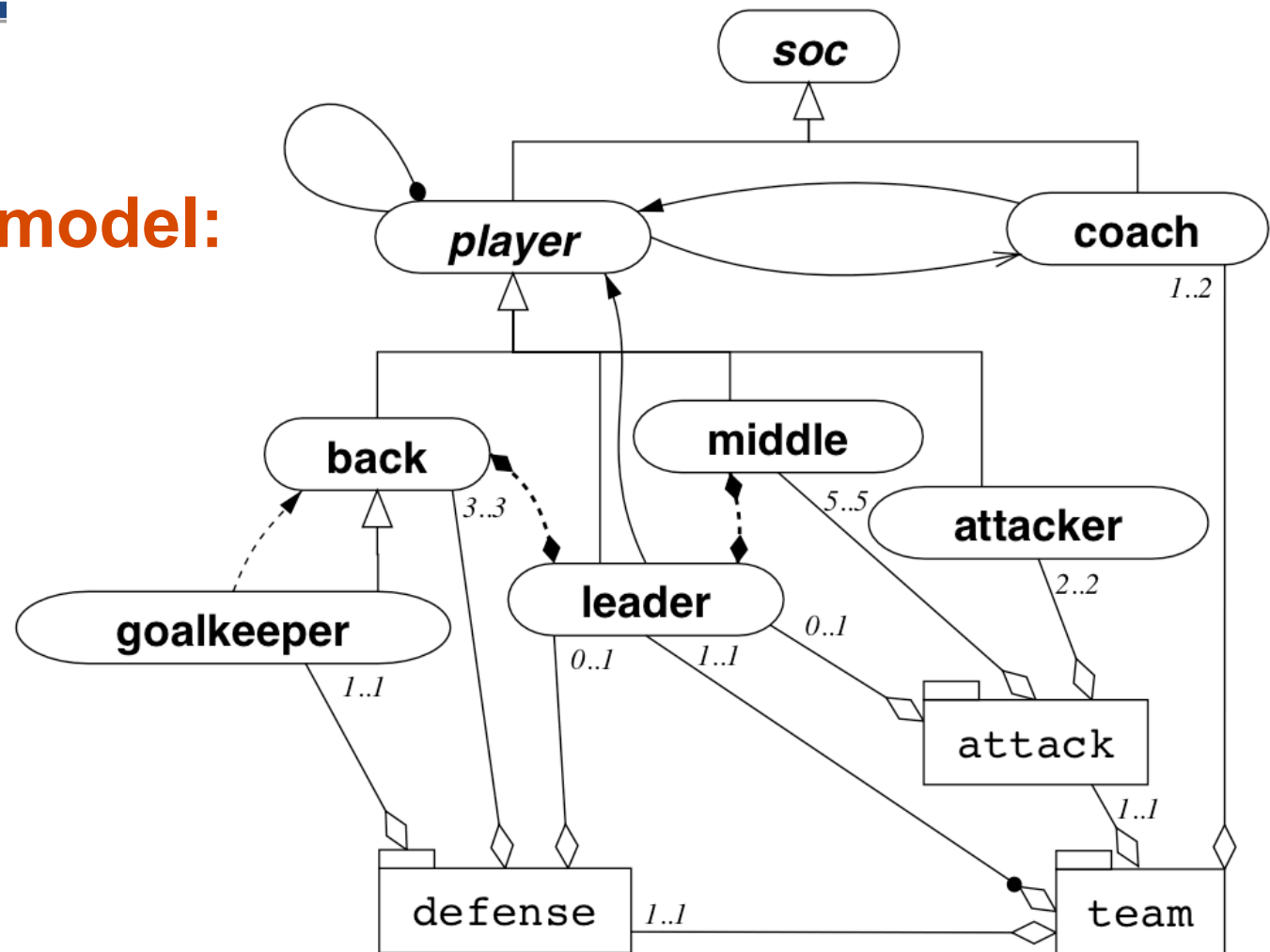
Problem

- **How to implement MAS that follow an organisation?**
- **Agent Centered approach**
 - How to develop agent reasoning mechanisms that are aware of the organisation?
 - Not suitable for open systems
- **Organisational approach (our focus)**
 - How to develop an agent infrastructure that ensures that the organisational constraints will be followed?
 - The agents have to respect the organisation despite their architecture.
- **Available tools**
 - Ameli (ISLANDER), MadKit (AGR), Karma (STEAMS)

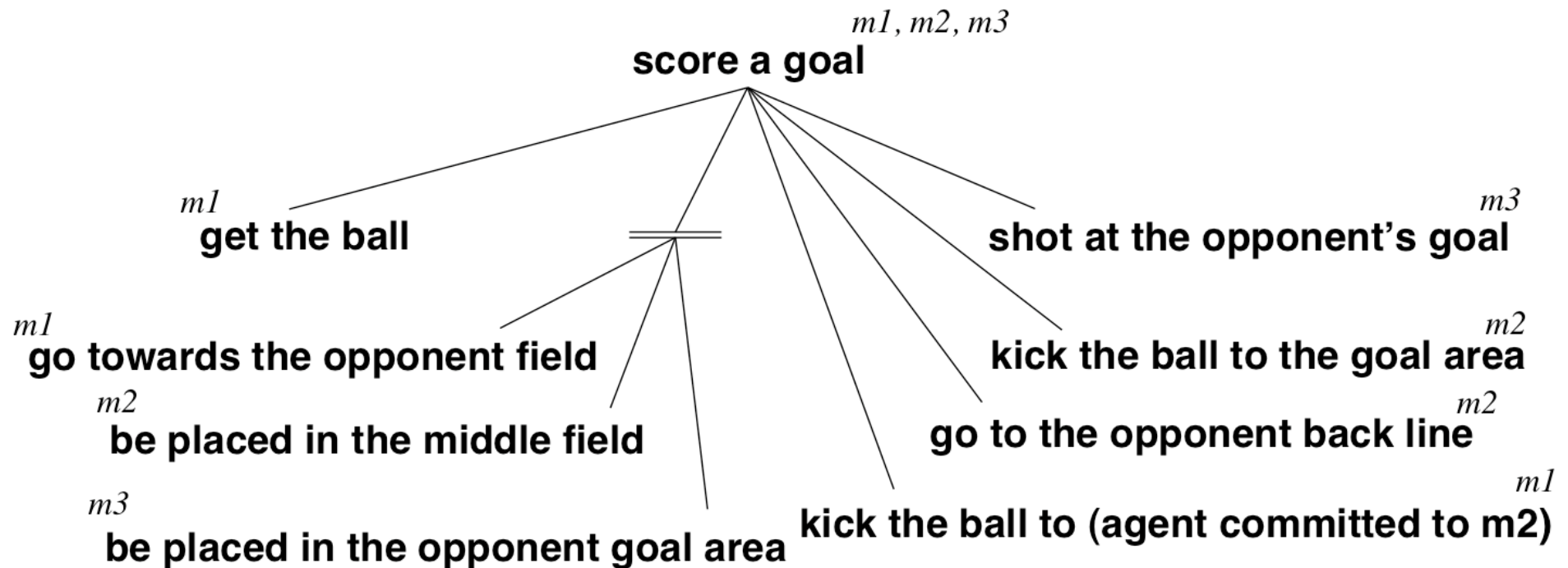
Objective

- **Develop an MAS infrastructure that**
 - Ensures that agents will follow the organisational constraints
 - Is suitable for open systems (so organisational approach)
 - Supports for reorganisation (so based on MOISE+ model)
- **This infrastructure is called S-MOISE+**
 - Based on MOISE+ model
 - Implemented using SACI for communication and distribution

The MOISE+ model: Structure (by example)



The MOISE+ mode: Functioning



Key

Scheme

missions
goal
success rate

sequence

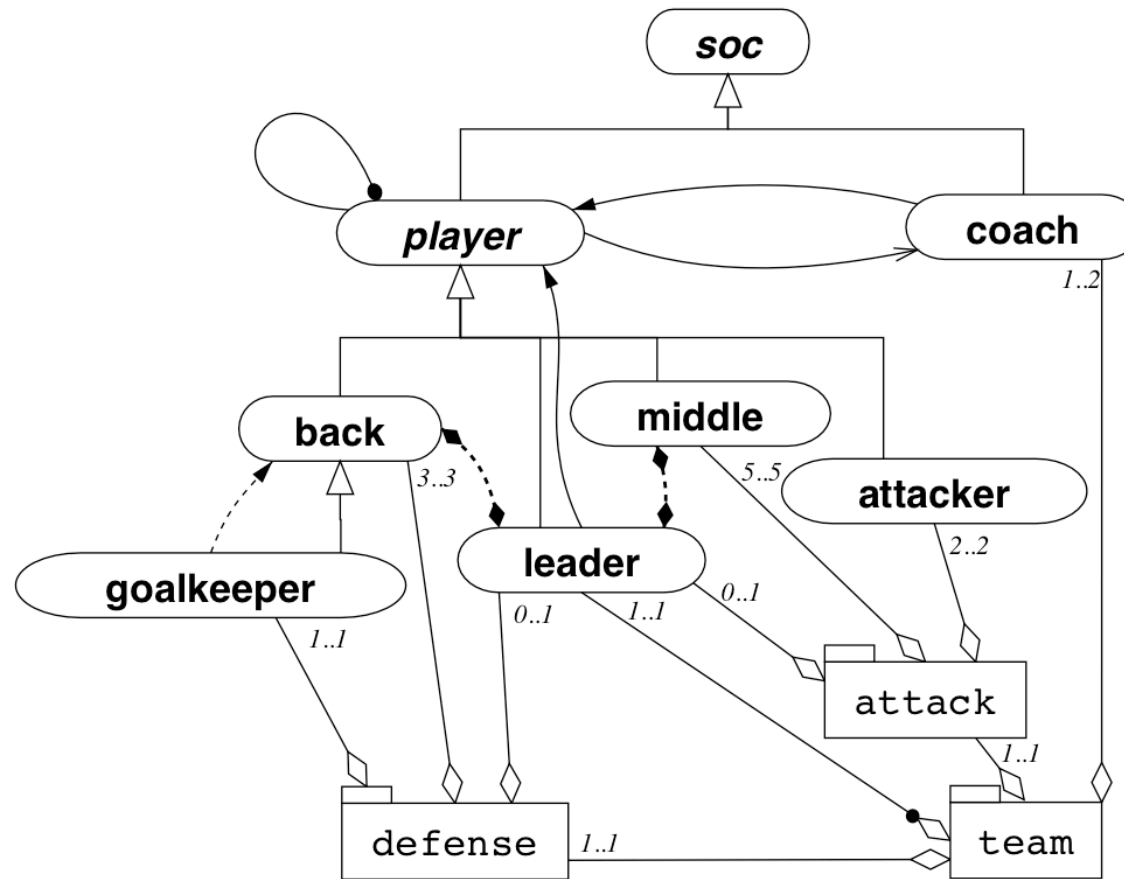
choice

parallelism

The MOISE+ model: Deontic

- **Link roles from Structure to Missions from Functioning**
 - Back role has **permission** for mission m1 (get the ball, go towards, ...)
 - Middle role has **obligation** for mission m2 (be placed in the middle field, kick the ball to the goal area, ...)
 - Attacker role has **obligation** for mission m3 (be placed in the opponent goal area, shot at the opponent's goal)

The MOISE+ model: Organisational Entity



Organizational Entity (structure 3-5-2)

Marcos ----- goalkeeper
 Lucio -----
 Edmilson ----- back
 Roque Jr. -----
 Cafu ----- leader
 Gilberto Silva ----- middle
 Juninho -----
 Ronaldinho -----
 Roberto Carlos -----
 Ronaldo ----- attacker
 Rivaldo -----

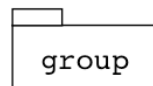
Key

Structure

inheritance:

composition: *min..max*

sub-groups scope:



role

Abs Role

links

intra-group

inter-group

acq



com



aut

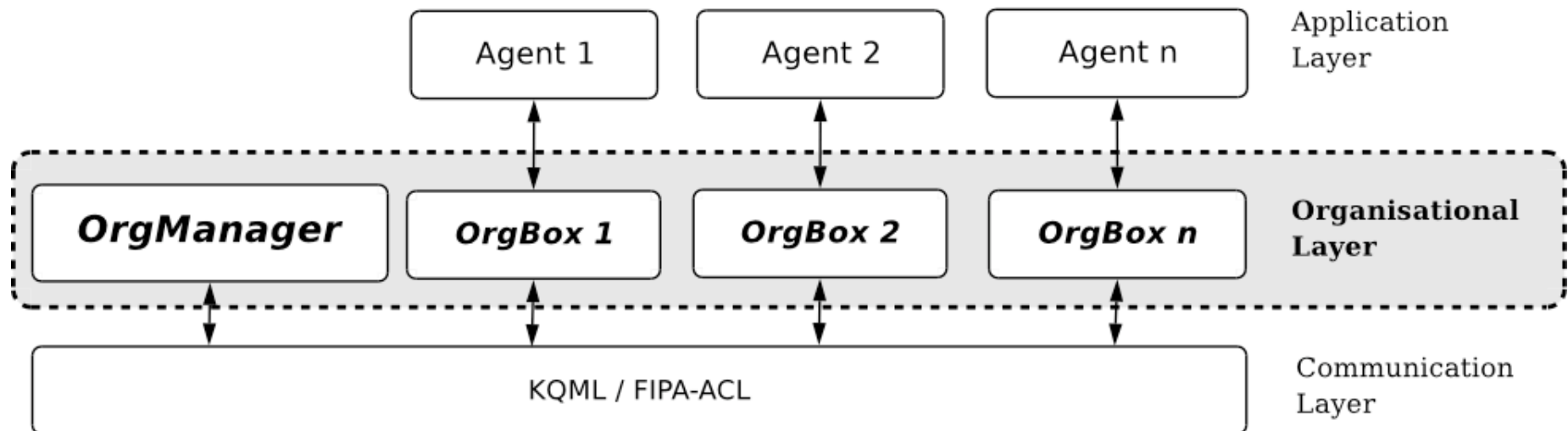


compat



Organisational Middleware: S-MOISE+

- Two main components: **OrgBox** and **OrgManager**



OrgManager Agent

- **Maintains the current state of the organisational entity**
 - Created groups and schemes
 - Scheme goals state (satisfied or not)
 - Roles assignments (Agents to Roles)
 - Missions assignments (Agents to Missions)
- **Receive messages from the agents' OrgBox asking for changes in the organisational entity**
 - Ensure that the request is organisationally permitted for the agent

Organisational Entity (OE) Dynamics

- The OE is changed by requests coming from agents' OrgBox
- Example:
 - createGroup('team'): a group ($g1$) is created from the team group specification
 - createSubGroup('defense', $g1$): a group ($g1-1$) is created from the defense specification as a $g1$ sub-group
 - createSubGroup('attack', $g1$): a group ($g1-2$) is created from the attack specification as a $g1$ sub-group
 - createScheme('side_attack', $g1$): an instance of the side attack scheme specification (schSA) is created, the agents of the group $g1$ are responsible for this scheme missions

Entity after the events

The
defense
group is
not **well
formed**
since there
is no agent
playing its
roles



The role adoption event

- **The adoption of a role R by an agent A in the group G has the following preconditions:**
 - The role R must belong to G specification
 - The number of R players in G must be lesser or equals than the maximum number of R players defined in the G compositional specification
 - For all roles R_i that A already plays, the roles R and R_i must be intra-group compatible in the G specification
 - For all roles R_i that A already plays in groups other than G , the roles R and R_i must be inter-group compatible.

Permitted goals and agent coordination for Scheme execution

- When an agent is committed to a mission, it is responsible for some goals. But only some may be permitted: those that its pre-goals are already satisfied.



OrgBox

- **The OrgBox is the interface the agents use to access the organisational layer and thus the communication layer**
- **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 to respect the acquaintance relation
- **OrgManager notifies OrgBox when the state of a scheme related to the OrbBox's agent changes**
- **No particular agent architecture is required**

Contribution

- **S-MOISE+ ensure that the agents follow some of the constraints specified for the organisation**
 - Cardinality of groups
 - Communication and acquaintance links
 - Role and mission adoption
 - Goals satisfaction
- **Follows an organisational centred approach**
- **The organisation is interpreted at runtime, not hardwired in the agents code**
- **Has a synchronisation mechanism for scheme execution**
- **Suitable for open systems since no specific agent architecture is required**
- **Based on AMELI, MadKit, and KARMA, but uses MOISE+ as organisational model (developed to enable reorganisation)**

Future work

- **Implement a sanction system to deal with agents that do not achieve its organisational goals**
- **Complement out work with an agent point of view.**
 - Organisational reasoning.
 - Implement an agent architecture to ensure, for instance, the authority link.