Multi-Agent Oriented Programming

– Organisation –

O. Boissier

Univ. Lyon, IMT Mines Saint-Etienne, LaHC UMR CNRS 5516, France

CPS2 M1 – Winter 2020

Tutorial Organisation

- Introduction
  - Multi-Agent Systems
  - Multi-Agent Oriented Programming (MAOP)
- Agent
  - Concepts and approaches
  - Programming agents in a shared working environment
- Agent working environment
  - Concepts and approaches
  - Programming agents and their working environment
- Agent organisation
  - Concepts and approaches
  - Programming organisations of agents
- Integrating Intelligent Systems with MAS
- Conclusion & Perspectives

Multi-Agent Oriented Programming

Agent Organization: concepts and approaches

Outline

Fundamentals

Existing approaches

Intuitive notions of organisation

- Organisations are structured, patterned systems of activity, knowledge, culture, memory, history, and capabilities that are distinct from any single agent [Gasser, 2001]
  ↝ Organisations are **supra-individual** phenomena
- A decision and communication schema which is applied to a set of actors that together fulfill a set of tasks in order to satisfy goals while guarantying a global coherent state [Malone, 1999]
  ↝ definition by the designer, or by actors, to achieve a **purpose**
- An organisation is characterized by : a division of tasks, a distribution of roles, authority systems, communication systems, contribution-retribution systems [Bernoux, 1985]
  ↝ **pattern of predefined cooperation**
- An arrangement of relationships between components, which results into an entity, a system, that has unknown skills at the level of the individuals [Morin, 1977]
  ↝ **pattern of emergent cooperation**

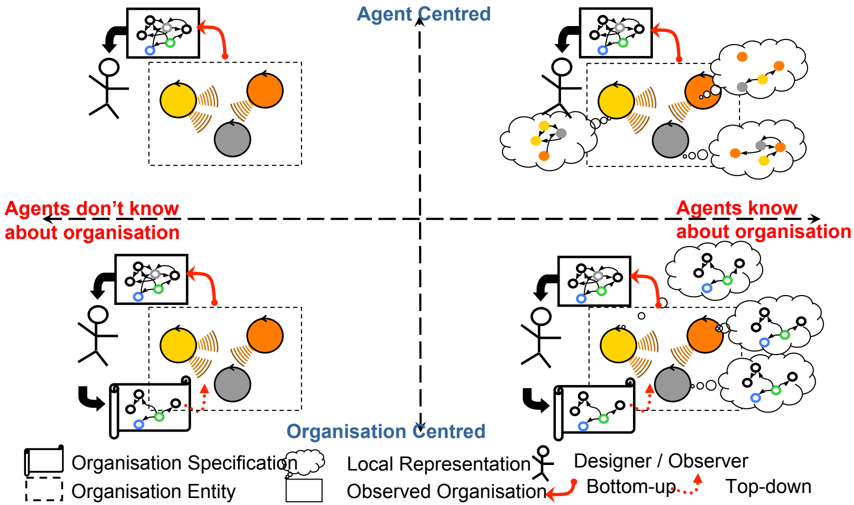
Organisation in MAS

Definition

Purposive **supra-agent** pattern of emergent or (pre)defined agents cooperation, that could be defined by the designer or by the agents themselves.

- Pattern of emergent/potential cooperation
  - called **organisation entity**, institution, social relations, commitments
- Pattern of (pre)defined cooperation
  - called **organisation specification**, structure, norms, ...

Perspective on organisations from EASSS'05 Tutorial (Sichman, Boissier)



Agent Centred

Agents don't know about organisation

Agents know about organisation

Organisation Centred

Organisation Specification

Organisation Entity

Local Representation

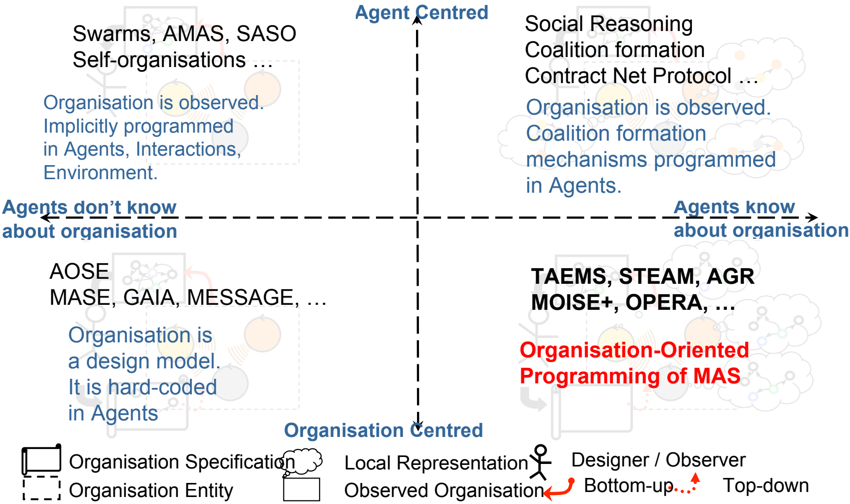
Observed Organisation

Designer / Observer

Bottom-up

Top-down

Perspective on organisations from EASSS'05 Tutorial (Sichman, Boissier)



Agent Centred

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.

Agents don't know about organisation

Agents know about organisation

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

Organisation Centred

Organisation Specification
Organisation Entity
Local Representation
Observed Organisation
Designer / Observer
Bottom-up
Top-down

Perspective on Org.-Oriented Programming of MAS

From organisations as

- an explicit description of the structure of the agents in the MAS in order to help them to interact

To organisations as

- the declarative and explicit definition of the coordination scheme aiming at “controlling/coordinating” the global reasoning of the MAS

⇝ Normative Organisations

Norms

Norm

Norms are **rules** that a society has in order to influence the behaviour of agents.

Norm mechanisms

- **Regimentation**: norm violation by the agents is prevented
  - e.g. the access to computers requires an user name
  - e.g. messages that do not follow the protocol are discarded
- **Enforcement**: norm violation by the agents is made possible but it is monitored and subject to incentives
  - e.g. a master thesis should be written in two years

  $\rightsquigarrow$ Detection of violations, decision about ways of enforcing the norms (e.g. sanctions)

Normative Multi-Agent Organisation

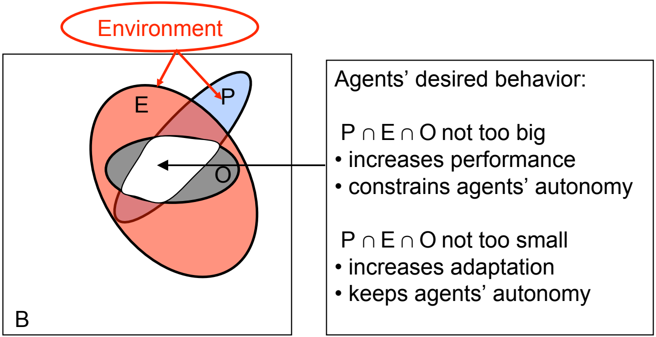
Normative Multi-Agent System [Boella et al., 2008]

A MAS composed of mechanisms to represent, communicate, distribute, detect, create, modify, and enforce norms, and mechanisms to deliberate about norms and detect norm violation and fulfillment.

Normative Multi-Agent Organisation [**?**]

- Norms are expressed in the organisation specification to clearly define the coordination of the MAS:
  - anchored/situated in the organisation
  - i.e. norms refer to organisational concepts (roles, groups, etc.)
- Norms are interpreted and considered in the context of the organisation entity
- Organisation management mechanisms are complemented with norms management mechanisms (enforcement, regimentation, ...)

Challenges: Normative Organisation vs 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

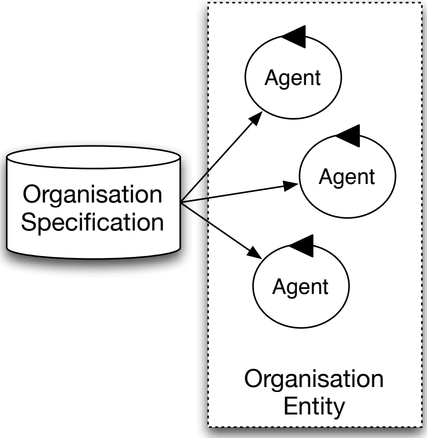
- B: agents' possible behaviors
- P: agents' behaviors that lead to global purpose
- E: agents' possible behaviors constrained by the environment
- O: agents' possible/permitted/obliged behaviors constrained by the normative organisation

Organisation Oriented Programming (OOP)

Organisation as a **first class entity** in the multi-agent eco-system

- Clear distinction between description of the organisation **wrt** agents, **wrt** environment
- Different representations of the organisation:
  - **Organisation specification**
    - partially/totally accessible to the agents, to the environment, to the organisation
  - **Organisation entity**
    - Local representation in the mental state of the agents
      ⤳ possibly inconsistant with the other agents' representations
    - Global/local representation in the MAS
      ⤳ difficulty to manage and build such a representation in a distributed and decentralized setting
- Different sources of actions on (resp. of) the organisation by (resp. on) agents / environment / organisation

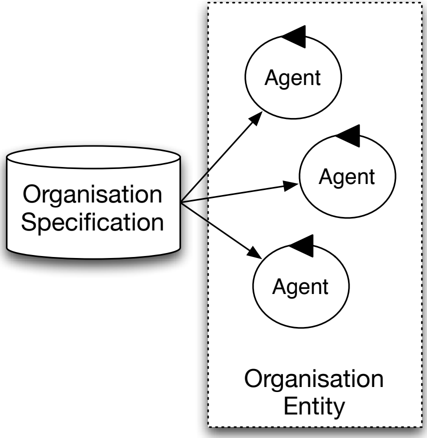
Organisation Oriented Programming (OOP)



- Using organisational concepts
- To define a cooperative pattern
- Programmed outside of the agents and outside of the environment

- Program = Specification
- By changing the organisation, we can change the MAS overall behaviour

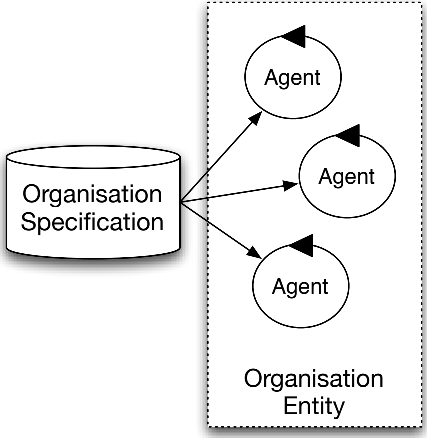
Organisation Oriented Programming (OOP)



First approach

- Agents read the program and follow it

Organisation Oriented Programming (OOP)



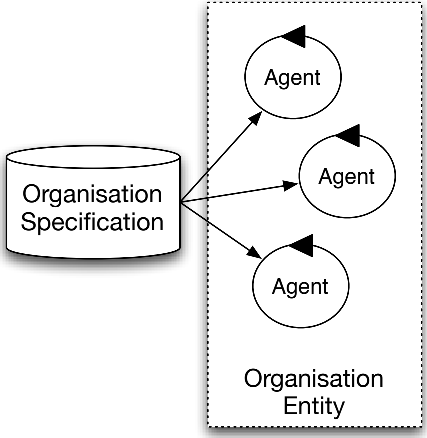
First approach

- Agents read the program and follow it

Second approach

- **regimentation**
  - Agents **are forced** to follow the program
- **enforcement**
  - Agents **are rewarded** if they follow the program
  - Agents **are sanctioned** in the other case

Organisation Oriented Programming (OOP)



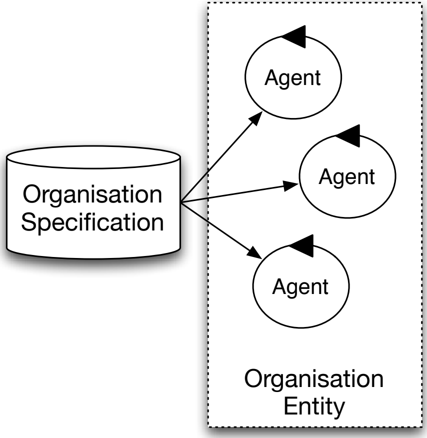
First approach

- Agents read the program and follow it

Second approach

- **regimentation**
  - Agents **are forced** to follow the program
- **enforcement**
  - Agents **are rewarded** if they follow the program
  - Agents **are sanctioned** in the other case

Organisation Oriented Programming (OOP)



Components

- Programming Language (Org. Modeling Lang. – OML)
- Management Infrastructure (Org. Mngt Inf. – OMI)
- Integration to Agent architectures and to Environment

Components of OOP: Organisation Modelling Language (OML)

- Declarative specification of the organisation(s)
- Specific constraints, norms and cooperation patterns imposed on the agents

  e.g. AGR [Ferber and Gutknecht, 1998],
  TeamCore [Tambe, 1997],
  Islander [Esteva et al., 2001],
  $\mathcal{M}$oise^{+} [Hübner et al., 2002], ...
- Specific anchors for situating organisations within the environment

  e.g. embodied organisations [Piunti et al., 2009]

Components of OOP:
Organisation Management Infrastructure (OMI)

- **Coordination mechanisms**, i.e. support infrastructure

  e.g. MadKit [Gutknecht and Ferber, 2000],
  karma [Pynadath and Tambe, 2003],
  ...

- **Regulation mechanisms**, i.e. governance infrastructure

  e.g. Ameli [Esteva et al., 2004],
  $\mathcal{S}$-$\mathcal{M}$oise^{+} [Hübner et al., 2006],
  ORA4MAS [Hübner et al., 2009],
  ...

- **Adaptation mechanisms**, i.e. reorganisation infrastructure

Components of OOP: Integration mechanisms

- **Agent** integration mechanisms allow agents to be aware of and to deliberate on:
  - entering/exiting the organisation
  - modification of the organisation
  - obedience/violation of norms
  - sanctioning/rewarding other agents

  e.g. $\mathcal{J}$-$\mathcal{M}$oise^{+} [Hübner et al., 2007], Autonomy based reasoning [Carabelea, 2007], *ProsA*$_2$ Agent-based reasoning on norms [Ossowski, 1999], ...

- **Environment** integration mechanisms transform organisation into embodied organisation so that:
  - organisation may act on the environment (e.g. enact rules, regimentation)
  - environment may act on the organisation (e.g. count-as rules)

  e.g [de Brito et al., 2012], [**?**], [Okuyama et al., 2008]

Motivations for OOP: **Applications** point of view

- Current applications show an increase in
  - Number of agents
  - Duration and repetitiveness of agent activities
  - Heterogeneity of the agents, Number of designers of agents
  - Agent ability to act, to decide,
  - Action domains of agents, ...
  - Openness, scalability, dynamicity, ...

- More and more applications require the integration of human communities and technological communities (ubiquitous and pervasive computing), building connected communities (ICities) in which agents act on behalf of users
  - Trust, security, ..., flexibility, adaptation

Motivations for OOP: **Constitutive** point of view

- Organisation **helps** the agents to cooperate with the other agents by defining **common** cooperation schemes
  - global tasks
  - protocols
  - groups, responsibilities

e.g. ‘to bid’ for a product on eBay is an **institutional action** only possible because eBay defines the rules for that very action

- the bid protocol is a constraint but it also **creates** the action

e.g. when a soccer team plays a match, the organisation helps the members of the team to synchronise actions, to share information, etc

Motivations for OOP: **Normative** point of view

- MAS have two properties which seem contradictory:
  - a **global** purpose
  - **autonomous** agents
  - ⤳ While the autonomy of the agents is essential, it may cause loss in the global coherence of the system and achievement of the global purpose
- Embedding **norms** within the **organisation** of a MAS is a way to constrain the agents' behaviour towards the global purposes of the organisation, while explicitly addressing the autonomy of the agents within the organisation
  - ⤳ Normative organisation
  - e.g. when an agent adopts a role, it adopts a set of behavioural constraints that support the global purpose of the organisation. It may decide to obey or disobey these constraints

Motivations for OOP: **Agents** point of view

An organisational specification is required to enable agents to “reason” about the organisation:

- to decide to enter into/leave from the organisation during execution
  - $\rightsquigarrow$ Organisation is no more closed
- to change/adapt the current organisation
  - $\rightsquigarrow$ Organisation is no more static
- to obey/disobey the organisation
  - $\rightsquigarrow$ Organisation is no more a regimentation

Motivations for OOP: **Organisation** point of view

An organisational specification is required to enable the organisation to "reason" about itself and about the agents in order to ensure the achievement of its global purpose:

- to decide to let agents enter into/leave from the organisation during execution
  - ⇝ Organisation is no more closed
- to decide to let agents change/adapt the current organisation
  - ⇝ Organisation is no more static and blind
- to govern agents behaviour in the organisation (i.e. monitor, enforce, regiment)
  - ⇝ Organisation is no more a regimentation

Outline

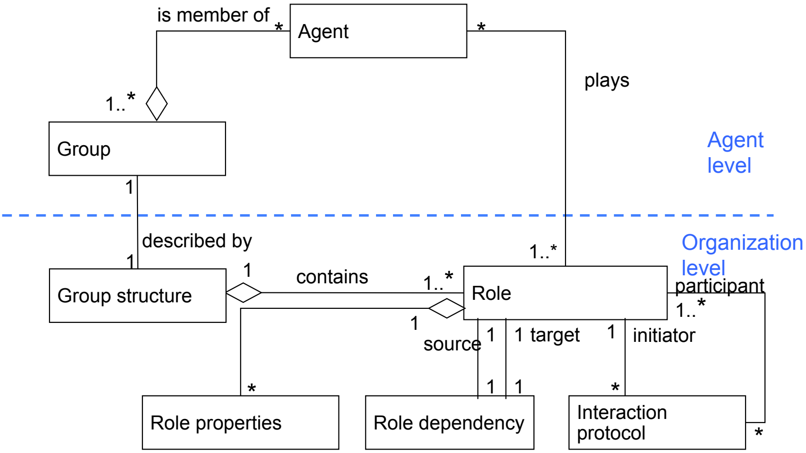
Fundamentals

Existing approaches

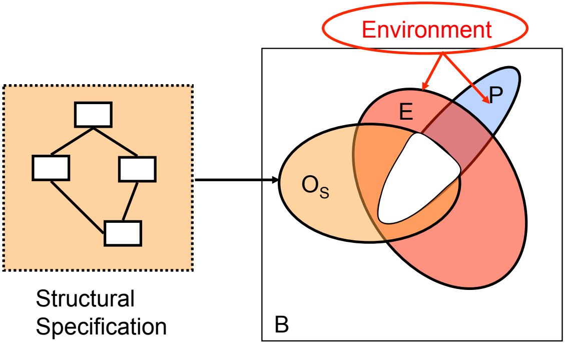
AGR [Ferber and Gutknecht, 1998]

- Agent Group Role, previously known as AALAADIN
    - Agent: Active entity that plays roles within groups. An agent may have several roles and may belong to several groups.
    - Group: set of agents sharing common characteristics, i.e. context for a set of activities. Two agents can't communicate with each other if they don't belong to the same group.
    - Role: Abstract representation of the status, position, function of an agent within a group.
- OMI: the Madkit platform

AGR OML

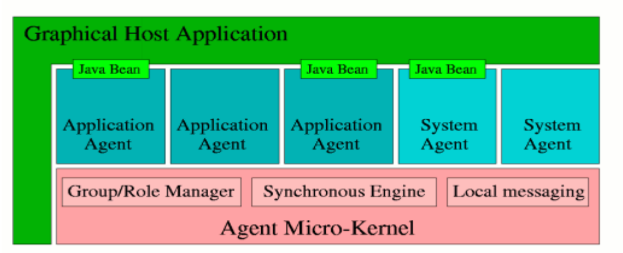

is member of
*
Agent
*
plays
1..*
Group
1
Agent
level
described by
1
1..*
Organization
level
1
contains
1..*
Group structure
Role
participant
1..*
1
source
1
1 target
1
initiator
*
1
1
*
Role properties
Role dependency
Interaction
protocol
*

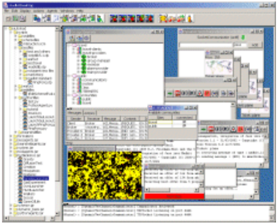
AGR OML Modelling Dimensions



B: agents' possible behaviors
P: agents' behaviors that lead to global purpose
E: agents' possible behaviors constrained by the environment
O_S: agents' possible behaviors structurally constrained by the organization

AGR OMI: Madkit





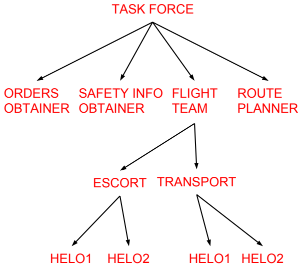
Multi-Agent Development Kit

www.madkit.org

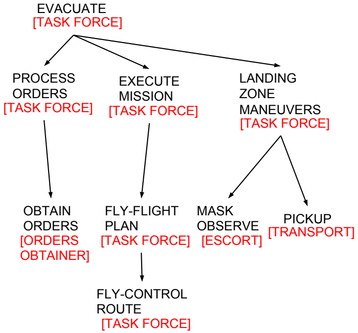
STEAM [Tambe, 1997]

- Shell for TEAMwork is a general framework to enable agents to participate in teamwork.
    - Different applications: Attack, Transport, Robocup soccer
    - Based on an enhanced SOAR architecture and 300 domain independent SOAR rules
- Principles:
    - Team synchronization: Establish joint intentions, Monitor team progress and repair, Individual may fail or succeed in own role
    - Reorganise if there is a critical role failure
    - Reassign critical roles based on joint intentions
    - Decision theoretic communication
- Supported by the TEAMCORE OMI.

STEAM OML [Tambe, 1997]



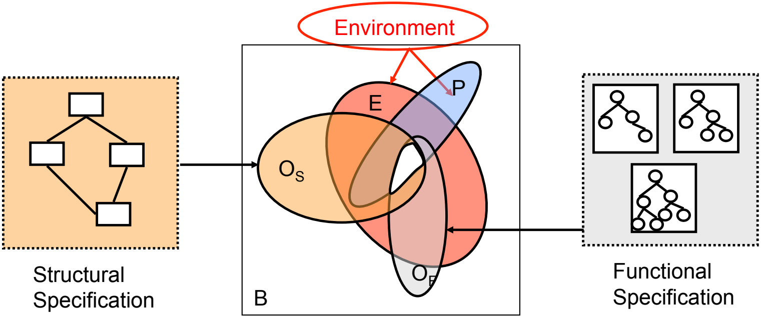
Organization: hierarchy of roles that may be filled by agents or groups of agents.


EVACUATE [TASK FORCE]
PROCESS ORDERS [TASK FORCE]
EXECUTE MISSION [TASK FORCE]
LANDING ZONE MANEUVERS [TASK FORCE]
OBTAIN ORDERS [ORDERS OBTAINER]
FLY-FLIGHT PLAN [TASK FORCE]
MASK OBSERVE [ESCORT]
PICKUP [TRANSPORT]
FLY-CONTROL ROUTE [TASK FORCE]

Team Plan:

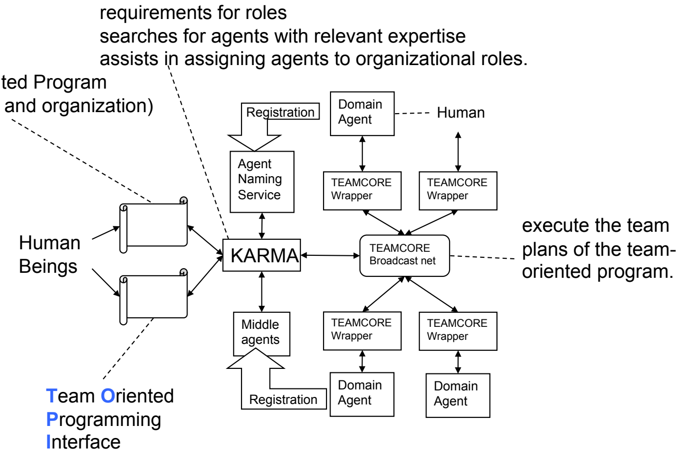
- initial conditions,
- term. cond. : achievability, irrelevance, unachievability
- team-level actions.

STEAM OML Modelling Dimensions



B: agents' possible behaviors
P: agents' behaviors that lead to global purpose
E: agents' possible behaviors constrained by the environment
O_S: agents' possible behaviors structurally constrained by the organization
O_F: agents' possible behaviors functionally constrained by the organization

STEAM OMI: TEAMCORE [Pynadath and Tambe, 2003]



requirements for roles
searches for agents with relevant expertise
assists in assigning agents to organizational roles.

Team-Oriented Program
(team plans and organization)

Human Beings

execute the team plans of the team-oriented program.

Team **O**riented **P**rogramming **I**nterface

ISLANDER

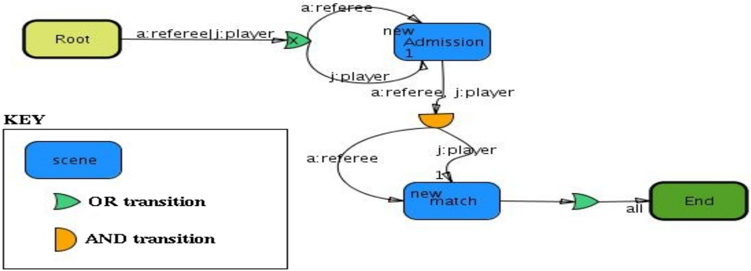
- Based on different influences: economics, norms, dialogues, coordination

⇝ electronic institutions

- Combining different alternative views: dialogical, normative, coordination
- Institution Description Language:
  - Performative structure (Network of protocols),
  - Scene (multi-agent protocol),
  - Roles,
  - Norms
- Ameli as OMI

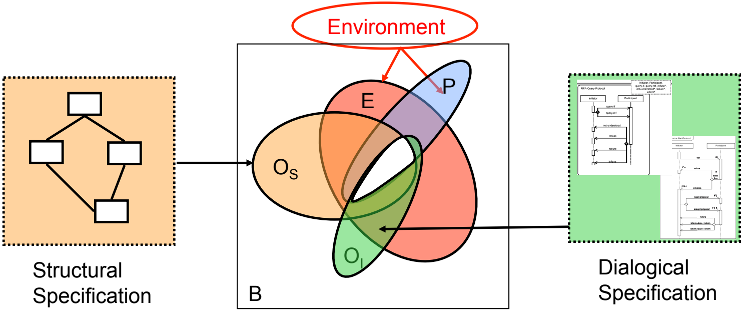
ISLANDER OML: IDL [Esteva et al., 2001]

```
(define-institution
          soccer-server as
          dialogic-framework = soccer-df
          performative-structure = soccer-pf
          norms =  ( free-kick  coach-messages … )
)
```



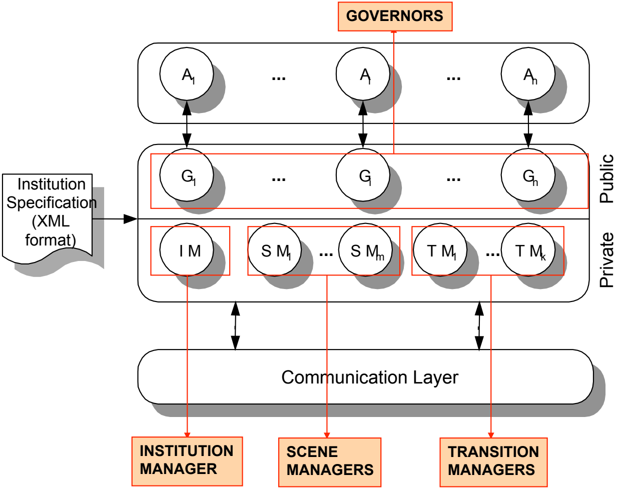
Performative Structure

ISLANDER OML Modelling Dimensions



B: agents' possible behaviors
P: agents' behaviors that lead to global purpose
E: agents' possible behaviors constrained by the environment
O_S: agents' possible/permitted/obliged behaviors structurally constrained by the organisation
O_I: agents' possible/permitted/obliged behaviors interactionally constrained by the organisation

ISLANDER OMI: AMELI [Esteva et al., 2004]



Agents Layer

AMELI

From [Noriega 04]

2OPL

slides from Dastani

The aim is to design and develop a programming language to support the implementation of coordination mechanisms in terms of **normative** concepts.

An organisation

- determines effect of external actions
- normatively assesses effect of agents' actions (monitoring)
- sanctions agents' wrongdoings (enforcement)
- prevents ending up in really bad states (regimentation)

Programming Language for Organisations

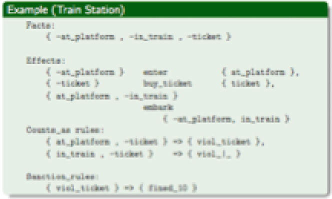
Example (Train Station)

```
Facts:
    { -at_platform , -in_train , -ticket }

Effects:
    { -at_platform }    enter           { at_platform },
    { -ticket }         buy_ticket      { ticket },
    { at_platform , -in_train }
                        embark
                            { -at_platform, in_train }
Counts_as rules:
    { at_platform , -ticket } => { viol_ticket },
    { in_train , -ticket }    => { viol_|_ }

Sanction_rules:
    { viol_ticket } => { fined_10 }
```

2OPL Modelling Dimension



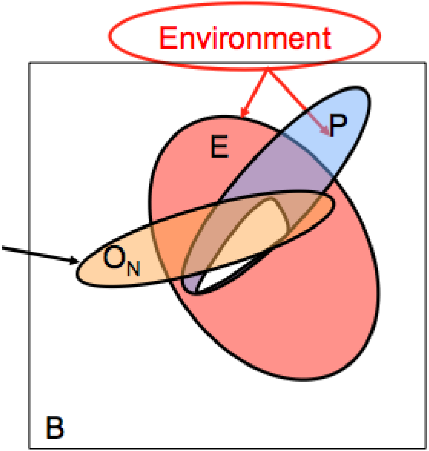
Example (Train Station)

```
Facts:
    { -at_platform , -in_train , -ticket }

Effects:
    { -at_platform }    enter          { at_platform },
    { -ticket }         buy_ticket     { ticket },
    { at_platform , -in_train }
                        embark
                            { -at_platform, in_train }
Counts_as rules:
    { at_platform , -ticket } => { viol_ticket },
    { in_train , -ticket }    => { viol_|_ }

Sanction_rules:
    { viol_ticket } => { fined_10 }
```

Normative Specification



Summary

- Several models
- Several dimensions on modelling organisation
  - Structural (roles, groups, ...)
  - Functional (global plans, ...)
  - Dialogical (scenes, protocols, ...)
  - Normative (norms)
- Several ways of managing organization within the MAS
- Several ways of addressing the autonomy of the agents

Bibliography I

Bernoux, P. (1985).
La sociologie des organisations.
Seuil, 3ème edition.

Boella, G., Torre, L., and Verhagen, H. (2008).
Introduction to the special issue on normative multiagent systems.
Autonomous Agents and Multi-Agent Systems, 17(1):1–10.

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.

de Brito, M., Hübner, J. F., and Bordini, R. H. (2012).
Programming institutional facts in multi-agent systems.
In *COIN-12, Proceedings*.

Bibliography II

Esteva, M., Rodriguez-Aguiar, 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.

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.

Bibliography III

Gasser, L. (2001).
Organizations in multi-agent systems.
In *Pre-Proceeding of the 10th European Worshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW'2001)*, Annecy.

Gutknecht, O. and Ferber, J. (2000).
The MadKit agent platform architecture.
In *Agents Workshop on Infrastructure for Multi-Agent Systems*, pages 48–55.

Hübner, J. F., Boissier, O., Kitio, R., and Ricci, A. (2009).
Instrumenting Multi-Agent Organisations with Organisational Artifacts and Agents.
Journal of Autonomous Agents and Multi-Agent Systems.

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.

Bibliography IV

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.

Malone, T. W. (1999).
Tools for inventing organizations: Toward a handbook of organizational process.
Management Science, 45(3):425–443.

Morin, E. (1977).
La méthode (1) : la nature de la nature.
Points Seuil.

Bibliography V

Okuyama, F. Y., Bordini, R. H., and da Rocha Costa, A. C. (2008).
A distributed normative infrastructure for situated multi-agent organisations.
In Baldoni, M., Son, T. C., van Riemsdijk, M. B., and Winikoff, M., editors, *DALT*, volume 5397 of *Lecture Notes in Computer Science*, pages 29–46. Springer.

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., 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 VI

Tambe, M. (1997).
Towards flexible teamwork.
Journal of Artificial Intelligence Research, 7:83–124.