The Moïse Framework
Organisation Oriented Programming of MAS

O. Boissier\textsuperscript{1} J. F. Hübner\textsuperscript{2} R. Kitio\textsuperscript{1} M. Piunti\textsuperscript{3} A. Ricci\textsuperscript{3} J.S. Sichman\textsuperscript{4}

\textsuperscript{1}Ecole Nationale Superieure des Mines - G2I St-Etienne - France
{boissier | kitio}@emse.fr

\textsuperscript{2}Federal University of Santa Catarina - DAS, Florianópolis - Brazil
jomi@das.ufsc.br

\textsuperscript{3}Università degli studi di Bologna - DEIS, Bologna - Italy.
{michele.piunti | a.ricci}@unibo.it

\textsuperscript{4}Universidade de São Paulo - LTI-PCS, São Paulo - Brazil
jaime.sichman@poli.usp.br
Motivations

Challenges:
▶ Seamless integration of Heterogeneity, Openness, Decentralisation, Large scale
Challenges:

- Seamless integration of Agents / Environments / Interactions / Organisations
Motivations

Challenges:

- Seamless integration of Agents / Environments / Interactions / Organisations

- Multi-Agent Oriented Programming

- Heterogeneity, Openness, Decentralisation, Large scale
Challenges: (Current focus)

▶ Seamless integration of Agents / Environments / Interactions / Organisations

→ Organisation Oriented Programming
→ Heterogeneity, Openness, Decentralisation, Large scale
Artifact meta-model for Environment

Events
- signal any relevant changes on artifacts’ state/processes
- can be perceived by agents
- can be collected and ranked at the workspace level

Coordination Artifacts
Example: Hospital Work Environment

"visitDoor" Environment Artifact

artifactName: visitDoor,
env.door
uic (operations): enter, exit
a-obs-prop: n_visitor
a-op-ev: enter,(op_exec_start)
    enter,(op_exec_complete)
    exit,(op_exec_start)
    exit,(op_exec_complete)
    a-prop-ev: n_visitor, (prop_updated)

"Terminal" Environment Artifact

artifactName: terminal,
env.terminal
uic (operations): send
a-op-ev: send, (op_exec_start)
    send, (signal, "sent")
    send, (op_exec_complete)

"billingMachine" Environment Artifact

artifactName: billingMachine,
env.billingMachine
uic (operations): pay, payFee
a-op-ev: pay, (op_exec_start)
    pay, (signal, receipt)
    pay, (op_exec_complete)
Moïse meta-model for Organisation

- Organisation modeling language (Moïse-OML)
- definition of multi-agent organisations
- Three dimensions [Hübner et al., 2007]:
  - Structural: Roles, Groups
  - Functional: Goals, Missions, Schemes
  - Normative: Norms (obligations, permissions, interdictions)
- Abstract description of the organisation for
  - the designers
  - the agents
    - J-Moïse+ [Hübner et al., 2007] Jason and Moïse+ integration
  - the organisation management
    - Organisation Management Infrastructure (OMI)
      (ORA4MAS [Hübner et al., 2009b])
Example: Hospital organisation

(a) Moise: Structural Specification

(b) Moise: Functional Specification

(c) Moise: normative description

<table>
<thead>
<tr>
<th>id</th>
<th>role</th>
<th>type</th>
<th>mission</th>
</tr>
</thead>
<tbody>
<tr>
<td>n0</td>
<td>patient</td>
<td>obl</td>
<td>m1</td>
</tr>
<tr>
<td>n1</td>
<td>staff</td>
<td>obl</td>
<td>m2</td>
</tr>
</tbody>
</table>
Organisation management infrastructure (OMI)

Responsibility:
- Managing – support, supervision – the agents’ execution in the organisation defined with Moïse OML

Normative Organisation Programming Language (NOPL)
- Automatic translation of organisation written in Moïse- OML into a NOPL - program [Hübner et al., 2009a]

Normative Organisation Program (NOP)

Example of Hospital NOP:

\[
\begin{align*}
N01 & : \quad \text{players}(\text{patient}, \text{vgroup}, V) \land (V \geq 10) \quad \rightarrow \quad \text{fail}(N01) \\
N02 & : \quad \text{committed-mission}(\text{patient}, m1) \quad \rightarrow \quad \text{obligation}(\text{patient}, \text{achieve-goal}(\text{book\_visit})) \\
N03 & : \quad \text{committed-mission}(\text{patient}, m1) \quad \rightarrow \quad \text{obligation}(\text{patient}, \text{achieve-goal}(\text{pay})) \\
N04 & : \quad \text{committed-mission}(\text{patient}, m1) \quad \rightarrow \quad \text{obligation}(\text{patient}, \text{achieve-goal}(\text{exit})) \\
N05 & : \quad \text{committed-mission}(\text{staff}, m2) \quad \rightarrow \quad \text{obligation}(\text{staff}, \text{achieve-goal}(\text{send-fee})) \\
N06 & : \quad \text{violated}(N03) \quad \rightarrow \quad \text{obligation}(\text{staff}, \text{commit-mission}(m2))
\end{align*}
\]
Organisational Artifacts (OA) [Hübner et al., 2009b]

- Instrumenting the agents’ working environment with OAs (dedicated NOPs Interpreters)
- Interactions with agents by actions and perceptions (A&A)
- "Institutional" actions: adoptRole, leaveRole, commitMission, setGoalAchieved, ...
**Agent + Environment + Organisation**

![Diagram of Moise-OML OS, Agent Platform(s), Bridge, and CArtAgO node]

**Jason + CArtAgO + Moise**

- Heterogeneous Agents (different architectures and agent platforms)
- Artefact based Environments (CArtAgO nodes)
  - Action repository: `.joinWorkspace`, `.makeArtifact`, `.lookupArtifacts`, ..., `.use`, `.observeProperty`, `.focus`
- Organisations defined with Moise-OML and managed within ORA4MAS
- Dedicated bridges: A-E (c4Jason, c4Jadex [Ricci et al., 2009]), A-O (JMoise [Hübner et al., 2007]), E-O (count-as, enact [Piunti et al., 2009])

---

Moise Framework

O. Boissier, J. F. Hübner, R. Kitio, M. Piunti, A. Ricci, J.S. Sichman

9 / 12
Example: Hospital Multi-Agent System
Future works

- Integration of Interaction models
- Enrichment of the Moïse OML to govern the Environments and Interactions models
- Multi-Agent Oriented Programming to develop large scale, open and decentralized applications
Bibliographie

Hübner, J. F., Boissier, O., and Bordini, R. H. (2009a). Normative programming for organisation management infrastructures. *In MALLOW Workshop on Coordination, Organization, Institutions and Norms in Agent Systems in Online Communities (COIN-MALLOW 2009).*


