Multi-Agent Oriented Programming
The JaCaMo Platform

O. Boissier\textsuperscript{1}  R.H. Bordini\textsuperscript{2}  J.F. Hübner\textsuperscript{3}  A. Ricci\textsuperscript{4}

1. Mines Saint-Etienne (ENSMSE), Saint Etienne, France
2. Pontificia Universidade Catolica do Rio Grande do Sul (PUCRS), Porto Alegre, Brazil
3. Federal University of Santa Catarina (UFSC), Florianópolis, Brazil
4. University of Bologna (UNIBO), Bologna, Italy

September 2016
In collaboration with

- **J.S. Sichman**, Universidade de São Paulo - LTI-PCS, São Paulo, Brazil (jaime.sichman@poli.usp.br)

- **G. Picard**, ENS Mines St-Etienne, France (gauthier.picard@emse.fr)

- **M. Hannoun, B. Gâteau, G. Danoy, R. Kitio, C. Persson, R. Yaich**, ENS Mines St-Etienne, France, **L. Coutinho** Brazil

- **M. Piunti, A. Santi**, Università degli studi di Bologna - DEIS, Bologna, Italy (a.ricci@unibo.it)

- **A. Ciortea, A. Sorici**, Politehnica University of Bucharest, Romania
Tutorial Organisation

- Introduction to Multi-Agent Oriented Programming
- Programming Agents within JaCaMo
- Programming Agents’ Environment within JaCaMo
- Programming Agents’ Interaction within JaCaMo
- Programming Agents’ Organisations within JaCaMo
- Conclusion & Perspectives
Acknowledgements

- **Dagstuhl Seminars:**
  - #12342 (2012), #08361 (2008), #06261 (2006)

- **Bilateral Projects:**
  - USP-COFECUB 98-04
  - CMIRA Rhône-Alpes Region 2010

- **French National Project:**
  - FORTRUST Project ANR 06-10
  - ETHICAA Project ANR 14-18
Multi-Agent Oriented Programming

Introduction
Outline

Introduction

Motivation

Multi-Agent Oriented Programming (MAOP)

MAOP Perspective in the JaCaMo Platform
Many AOSE methodology (Prometheus, Gaia, Tropos, ...) exist!

Use at least one of these methodologies for analysing and designing your MAS application.

Many agent languages have efficient and stable interpreters — used extensively in teaching.

All have some programming tools (IDE, tracing of agents’ mental attitudes, tracing of messages exchanged, etc.).

Some are integrating social aspects of MAS.

However, there are not yet proper tools to program multi-agent systems!

Some reasons and motivations follow!
Agents: abstractions for the definition of the decision/reasoning entities architectures

Environment: abstractions for structuring resources, processing entities shared among the agents

Interaction: abstractions for structuring interactions among entities

Organisation: abstractions for structuring and ruling the sets of entities within the MAS


〜 A rich set of abstractions for capturing applications complexity!
Each dimension has its own dynamics

Dynamics may be interlaced into bottom-up / top-down global cycles

Coordination of these dynamics may be programmed into one or several dimensions [Boissier, 2003]

〜 A rich palette of possible dynamics & coordination!!
MAS Programming

- **Agent Oriented Programming** [Shoham, 1993]
- **Environment Oriented Programming** [Ricci et al., 2011]
- **Interaction Oriented Programming** [Huhns, 2001]
- **Organisation Oriented Programming** [Pynadath et al., 1999]

- In these approaches, some dimensions lose their control & visibility!
- Integrating the dimensions into one programming platform is not so easy!
- Examples of Multi-Agent Oriented Programming Platforms:
  - Volcano platform [Ricordel and Demazeau, 2002], MASK platform [Occello et al., 2004], MASQ [Stratulat et al., 2009], extending AGRE and AGREEN, Situated E-Institutions [Campos et al., 2009], ...
Shifting from an A/E/I/O oriented approaches to a Multi-Agent Oriented approach

- keeping alive the concepts, dynamics and coordinations of the A, E, I and O dimensions
Outline

Introduction

Motivation

Multi-Agent Oriented Programming (MAOP)

MAOP Perspective in the JaCaMo Platform
Seamless Integration of A & E & I & O

JaCaMo Meta-model [Boissier et al., 2011], based on Cartago [Ricci et al., 2009b], Jason [Bordini et al., 2007], Moise [Hübner et al., 2009a] meta-models
Agent meta-model

Based on Jason meta-models [Bordini et al., 2007]
Agent & Agent Interaction meta-model

Agent
Belief
Goal
Plan
External Action
Internal Action
Trigger event
Action
Agent
Dimension
Content
Message
SpeechAct
Interaction
Dimension

Agent
Plan
Trigger event
Belief
Goal

Interaction
Message
Content
SpeechAct

Agent Dimension
External Action
Internal Action
Action

Interaction Dimension
Message
Content
SpeechAct
Agent’s dynamics
Environment meta-model

Based on A&A meta-model [Omicini et al., 2008]
Environment’s dynamics

<table>
<thead>
<tr>
<th>Artifact life-cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Creation/Deletion</td>
</tr>
<tr>
<td>▶ Activation/Execution/Fail or Success/Deactivation of an Operation</td>
</tr>
<tr>
<td>▶ Linking / Unlinking</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workspace life-cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Creation/Deletion of a workspace</td>
</tr>
<tr>
<td>▶ Creation/Deletion of Artifacts</td>
</tr>
<tr>
<td>▶ Creation/Deletion &amp; Entry/Exit of Agents</td>
</tr>
</tbody>
</table>
Outcomes of A & E Integration

▶ Agents with dynamic action repertoire, extended/reshaped by agents themselves
▶ Uniform implementation of any mechanisms (e.g. coordination mechanism) in terms of actions/percepts
  ▶ No need to extend agents with special purpose primitives
▶ Exploiting a new type of agent modularity, based on externalization [Ricci et al., 2009a]
Organisation meta-model

Simplified Moise meta-model [Hübner et al., 2009a]
Based on Cartago [Ricci et al., 2009b], Jason [Bordini et al., 2007], Moise [Hübner et al., 2009a] meta-models
A & O Integration

- Definition of organisational beliefs, organisational actions mediating the perception and actions on the organisation entity
- Done by instrumenting the organisation management by dedicated Organisational Artifacts
  - Mapping of the organisational state onto artifacts computational state
  - Encapsulation of organisational functionalities by suitably designed artifacts providing organisational operations
- Reification of organisation management actions/perceptions by actions/percepts on the artifacts

- Extensible set of organisational artifacts:
  - Openness Management Artifact [Kitio, 2011]
  - Reorganisation Artifact [Sorici, 2011]
  - Evaluation Artifact (kind-of reputation artifact) [Hübner et al., 2009b]
  - Communication management Artifact [Ciortea, 2011]
Exploit the uniform access to artifacts

Agents may be aware of the Organisation by the way of:

- organisational events
- organisational actions

Agents can reason on the organisation:

- to achieve organisational goals
- by developing organisational plans
Env. Artifacts provide operations on shared resources

Org. Artifacts provide organisational operations

Both artifacts bound by count-as, enact constitutive rules [Piunti et al., 2009, de Brito et al., 2012]

Org-agnostic agents may indirectly act on the organisation

Environment can act on the organisation

Organisation is embodied, situated in the environment
**Organisation’s dynamics** (triggered by Agents, Environment)

<table>
<thead>
<tr>
<th>Organisation life-cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Entrance/Exit of an agent</td>
</tr>
<tr>
<td>▶ Creation/Deletion of an Organisation entity</td>
</tr>
<tr>
<td>▶ Change of Organisation specification</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Structural Organisation life-cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Creation/Deletion of a group</td>
</tr>
<tr>
<td>▶ Adoption/Release of a role</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Functional Organisation life-cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Creation/End of a schema</td>
</tr>
<tr>
<td>▶ Commitment/Release of a mission</td>
</tr>
<tr>
<td>▶ Change of a global goal state</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Normative Organisation life-cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Activation/De-activation of obligation</td>
</tr>
<tr>
<td>▶ Fulfilment/Violation/Sanction</td>
</tr>
</tbody>
</table>
Outcomes of A & E & O Integration

- Normative deliberative agents
  - possibility to define mechanisms for agents to evolve within an organisation/several organisations
  - possibility to define proper mechanisms for deliberating on the internalisation/adoptive/violation of norms

- Reorganisation, adaptation of the organisation
  - possibility to define proper mechanisms for diagnosing/evaluating/refining/defining organisations

- “Deliberative” Organisations
  - possibility to define dedicated organisational strategies for the regulation/adaptation of the organisation behaviour (organisational agents)

- “Embodied” Organisation / Organisation Aware Environment
  - possibility to connect organisation to environment
Synthesis: MAOP meta-model

JaCaMo Meta-model [Boissier et al., 2011], based on Cartago [Ricci et al., 2009b], Jason [Bordini et al., 2007], Moise [Hübner et al., 2009a] meta-models
Outline

Introduction
Motivation
Multi-Agent Oriented Programming (MAOP)
MAOP Perspective in the JaCaMo Platform
Execution Platform

Agent execution and communication management infrastructures can be:

**Centralised**: all agents in the same machine, one thread by agent, very fast

**Centralised (pool)**: all agents in the same machine, fixed number of threads, allows thousands of agents

**Distributed (jade)**: distributed agents, FIPA-ACL communication using Jade

.... others defined by the user (e.g. AgentScape)

Environment execution can be:

**Centralised**: one centralised environment shared by the agents, is automatically included in case of no other specification

**Distributed**: multiple environments shared by the agents – specified by cartago("infrastructure")
Simple way of defining a multi-agent system within the JaCaMo Platform

Example (Building House Definition)

```plaintext
mas house_building {
    agent giacomo       // the agent that wants to build a house
    agent companyA     // builder agents (see their code for details)
    agent companyB
    agent companyC {
        instances: 5
    }
    agent companyD {
        instances: 13
    }
    agent companyE

    asl-path: src/agt, src/agt/inc
}
```
Eclipse JaCaMo plugin
Agent’s Mind inspector

Inspection of agent orgmajordomo

- Beliefs

  commitment(italian,mItaly,"JacamoPlan")
  commitment(french,mFrench,"JacamoPlan")
  commitment(brazilian1,mBrazil,"JacamoPlan")
  commitment(brazilian2,mBrazil,"JacamoPlan")
  current_wsp(cobj_1,"server","427dd8d5-408e-431a-a702-7b11ce574e09")
  formationStatus(ok)
  goalState("JacamoPlan",greetings,french,french,satisfied)
  goalState("JacamoPlan",greetings_uk_done,french,french,satisfied)
  goalState("JacamoPlan",greetings_italy_done,french,french,satisfied)
  goalState("JacamoPlan",greetings_brazil_done,brazilian1,brazilian2,brazilian1,brazilian2,satisfied)
  goalState("JacamoPlan",greetings_france_done,french,french,satisfied)
  groups(["JacamoTeam"])
  my_group("JacamoTeam")
  my_group_id(cobj_2)
  my_ssch("JacamoPlan")
  my_ssch_id(cobj_3)
Organization Structure **inspector**

**Structural Specification**

**Roles**

- `greeter` extends `soc`.
- `greetere` extends `greeter`.
- `greeterb` extends `greeter`.
- `greeteri` extends `greeter`.
- `greeteff` extends `greeter`.

**Group `jacamoGr`**

Possible roles: `greeterb`, `greeteri`, `greeteff`.

Local links:

- `greeter` has a `communication` link to `greeter` (intra-group, does not extend to subgroups)

**Constraint Formation**

- **Cardinalities**
  - Cardinality of `greeterb` is (2, 2)
  - Cardinality of `greeteri` is (1, 1)
  - Cardinality of `greeteff` is (1, 1)
jacamoTeam (group)

created from specification jacamoGr (root group) - owner is orgmajojilmo

Formation:
  ok

Players
  - brazilian1 plays greeter:b
  - brazilian2 plays greeter:b
  - french plays greeterf
  - italian plays greeterl

Responsible for the following schemes:
  - jacamoPlan
Organization Functioning **inspector**
### Organization Functioning: inspector

**jacamoPlan (scheme instance)**

created from specification `jacamoSch`

**Formation:**
- ok

**Responsible groups:** `jacamoTeam`

**Players**
- `brazilian1` committed to `mBrazil`
- `brazilian2` committed to `mBrazil`
- `french` committed to `mFrench`
- `italian` committed to `mItalia`

<table>
<thead>
<tr>
<th>goal</th>
<th>state</th>
<th>committed/achieved by</th>
<th>arguments</th>
<th>plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>greetings</td>
<td>satisfied</td>
<td>[french]/[french]</td>
<td></td>
<td><code>greetings_france</code>, <code>greetings_brazil</code>, <code>greetings_italy</code>, <code>greetings_uk_done</code></td>
</tr>
<tr>
<td>greetings_france_done</td>
<td>satisfied</td>
<td>[french]/[french]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>greetings_brazil_done</td>
<td>satisfied</td>
<td>[brazilian1,brazilian2]/[brazilian1,brazilian2]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>greetings_italy_done</td>
<td>satisfied</td>
<td>[italian]/[italian]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>greetings_uk_done</td>
<td>satisfied</td>
<td>[italian]/[italian]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**History**

- `created: obligation([french,n1,committed([french,mFrench,"jacamoPlan"],1411504910034)])`
- `created: obligation([brazilian1,n2,committed([brazilian1,mBrazil,"jacamoPlan"],14115049)])`
- `created: obligation([brazilian2,n2,committed([brazilian2,mBrazil,"jacamoPlan"],14115049)])`
- `created: obligation([italian,n3,committed([italian,mItalia,"jacamoPlan"],1411504910098)])`
- `created: obligation([french,ngoal("jacamoPlan",mFrench,greetings_france_done),achieved])`
- `created: obligation([brazilian1,ngoal("jacamoPlan",mBrazil,greetings_brazil_done),achieved])`
- `created: obligation([brazilian2,ngoal("jacamoPlan",mBrazil,greetings_brazil_done),achieved])`
- `created: obligation([italian,ngoal("jacamoPlan",mItalia,greetings_italy_done),achieved])`
- `created: obligation([italian,ngoal("jacamoPlan",mItalia,greetings_uk_done),achieved])`
Integration of Multi-Agent technologies

- **Agent**: Jason agents [Bordini et al., 2007]
- **Environment**: CArtAgO platform [Ricci et al., 2009b]
- **Organisation**: Moise framework with the extended/refactored version of the Moise OMI: ORA4MAS [Hübner et al., 2009a]
- **Interaction**: based on tight integration between Jason and KQML or ACL/FIPA

Dimensions are integrated with dedicated bridges:

- **A–E** (c4Jason, c4Jadex [Ricci et al., 2009b])
- **E–O** (count-as/enact rules [Piunti et al., 2009, de Brito et al., 2015])
- **A–O** is for free (thanks to ORA4MAS). Strategies and reasoning capabilities from J-Moise⁺ [Hübner et al., 2007] can be reused.

Open to integrate other Multi-Agent Technologies
Integration with other technologies

- **Web 2.0**
  - implementing Web 2.0 applications

- **Android Platforms**
  - implementing mobile computing applications on top of the Android platform

- **Web Services**
  - building SOA/Web Services applications

- **Arduino Platforms**
  - building “Web of Things” Applications

- **Semantic Technologies**
  - JaSA: Semantically Aware Agents
Multi-Agent Oriented Programming
The JaCaMo Platform

O. Boissier¹  R.H. Bordini²  J.F. Hübner³  A. Ricci⁴

1. Mines Saint-Etienne (ENSMSE), Saint Etienne, France
2. Pontificia Universidade Catolica do Rio Grande do Sul (PUCRS), Porto Alegre, Brazil
3. Federal University of Santa Catarina (UFSC), Florianópolis, Brazil
4. University of Bologna (UNIBO), Bologna, Italy

September 2016
Bibliography I


Modeling relationships for privacy preservation in virtual communities.

Bringing constitutive dynamics to situated artificial institutions.
In Proc. of 17th Portuguese Conference on Artificial Intelligence (EPIA 2015),
volume 9273 of LNCS, pages 624–637. Springer.

Programming institutional facts in multi-agent systems.
In COIN-12, Proceedings.

From interactions to collective behaviour in agent-based systems.
117–132.
Bibliography III

Steps towards multi-agent oriented programming.
(slides Workshop) 1st International Workshop on Multi-Agent Systems, IWMAS'97, Boston.

Instrumenting Multi-Agent Organisations with Organisational Artifacts and Agents.
*Journal of Autonomous Agents and Multi-Agent Systems.*

Developing Organised Multi-Agent Systems Using the MOISE+ Model: Programming Issues at the System and Agent Levels.

Instrumenting Multi-Agent Organisations with reputation artifacts.
Interaction-oriented programming.


MASK: An AEIO toolbox to design and build multi-agent systems.
In et al., C., editor, Knowledge Engineering and Agent Technology, IOS Series on Frontiers in AI and Applications. IOS press, Amsterdam.

Artifacts in the A&A meta-model for multi-agent systems.
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.

Toward team-oriented programming.


Environment programming in multi-agent systems: an artifact-based perspective.
Environment programming in CArtAgO.

VOLCANO: a vowels-oriented multi-agent platform.

Agent-oriented programming.

Agile governance in an ambient intelligence environment.