

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

February 2017

Tutorial Organisation

- ▶ Introduction to Multi-Agent Oriented Programming
- ▶ Programming Agents
- ▶ Programming Agents' Environment
- ▶ Programming Agents' Interaction
- ▶ Programming Agents' Organisations
- ▶ Programming Applications
- ▶ Conclusion & Perspectives

Multi-Agent Oriented Programming

Introduction

Outline

Introduction

Motivation

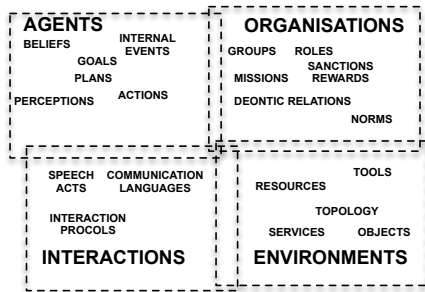
Multi-Agent Oriented Programming (MAOP)

MAOP Perspective in the JaCaMo Platform

Current Landscape of Multi-Agent Programming

- ▶ 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!

MAS Conceptual framework / Dimensions

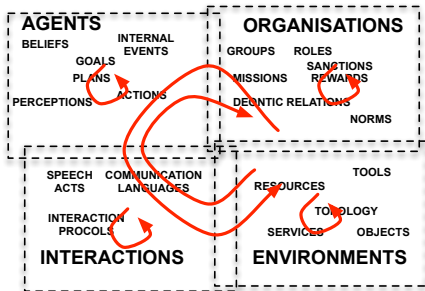


cf. VOWELS [Demazeau, 1995,
Demazeau, 1997]

- ▶ **A**gents: abstractions for the definition of the decision/reasoning entities architectures
- ▶ **E**nvironment: abstractions for structuring resources, processing entities shared among the agents
- ▶ **I**nteraction: abstractions for structuring interactions among entities
- ▶ **O**rganisation: abstractions for structuring and ruling the sets of entities within the MAS

→ A rich set of abstractions for capturing applications complexity!

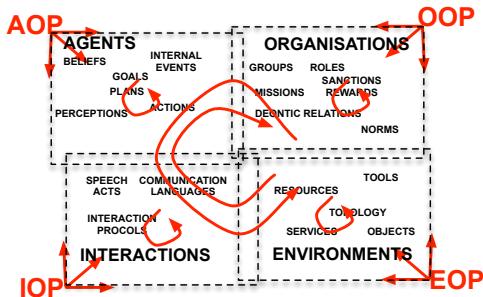
MAS Conceptual framework / Dynamics



- ▶ 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!!

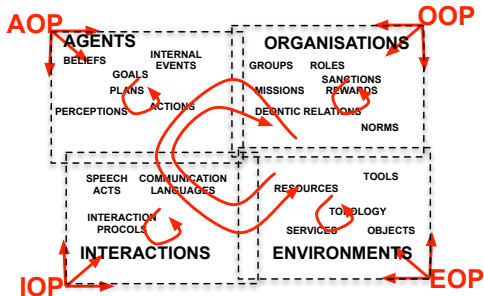
MAS Programming



- ▶ **A**gent **O**riented **P**rogramming [Shoham, 1993]
- ▶ **E**nvironment **O**riented **P**rogramming [Ricci et al., 2010b]
- ▶ **I**nteraction **O**riented **P**rogramming [Huhns, 2001]
- ▶ **O**rganisation **O**riented **P**rogramming [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], ...

MAS Programming



- ▶ Agent Oriented Programming [Shoham, 1993]
- ▶ Environment Oriented Programming [Ricci et al., 2010b]
- ▶ Interaction Oriented Programming [Huhns, 2001]
- ▶ Organisation Oriented Programming [Pynadath et al., 1999]

Challenge

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

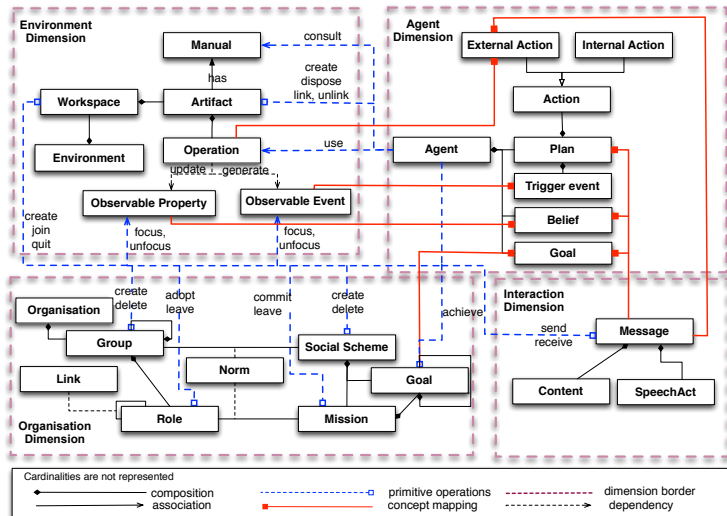
Introduction

Motivation

Multi-Agent Oriented Programming (MAOP)

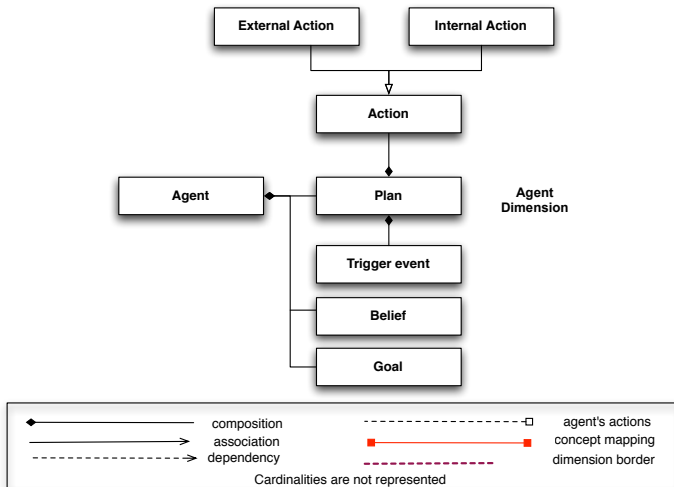
MAOP Perspective in the JaCaMo Platform

Seamless Integration of A & E & I & O



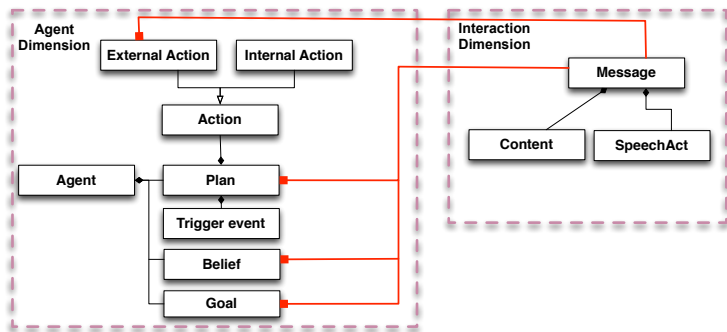
JaCaMo Meta-model [Boissier et al., 2011], based on Cartago [Ricci et al., 2009b], Jason [Bordini et al., 2007], Moise [Hübner et al., 2009a] meta-models

Agent meta-model

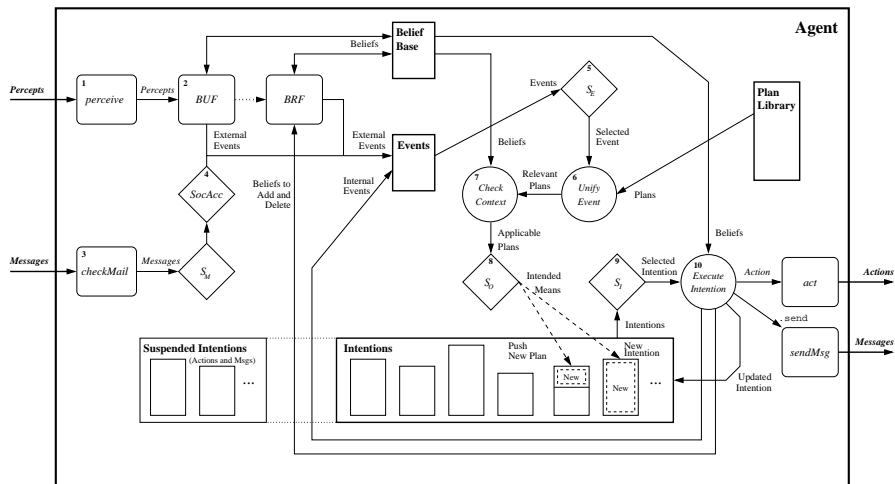


Based on Jason meta-models [Bordini et al., 2007]

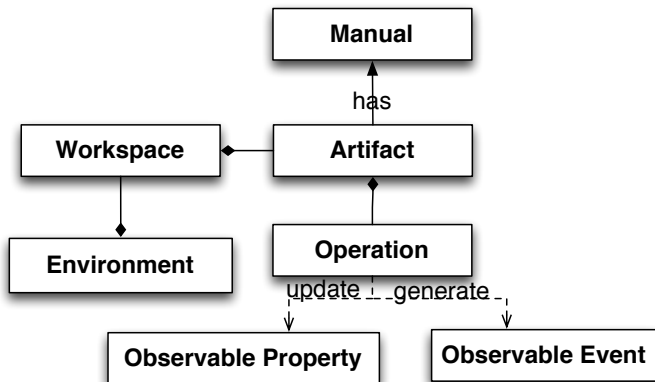
Agent & Agent Interaction meta-model



Agent's dynamics

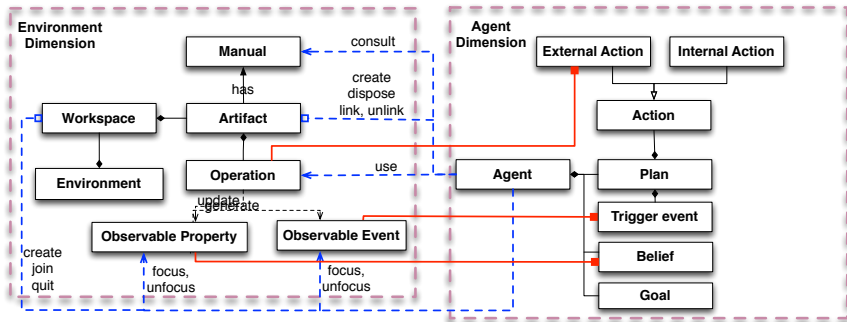


Environment meta-model



Based on A&A meta-model [Omicini et al., 2008]

A & E Interaction meta-model



Environment's dynamics

Artifact life-cycle

- ▶ Creation/Deletion
- ▶ Activation/Execution/Fail or Success/Deactivation of an Operation
- ▶ Linking / Unlinking

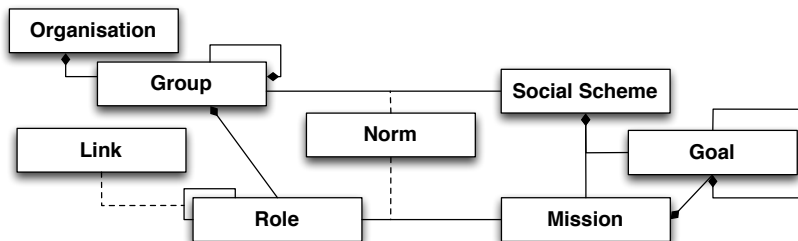
Workspace life-cycle

- ▶ Creation/Deletion of a workspace
- ▶ Creation/Deletion of Artifacts
- ▶ Creation/Deletion & Entry/Exit of Agents

Outcomes of A & E Integration

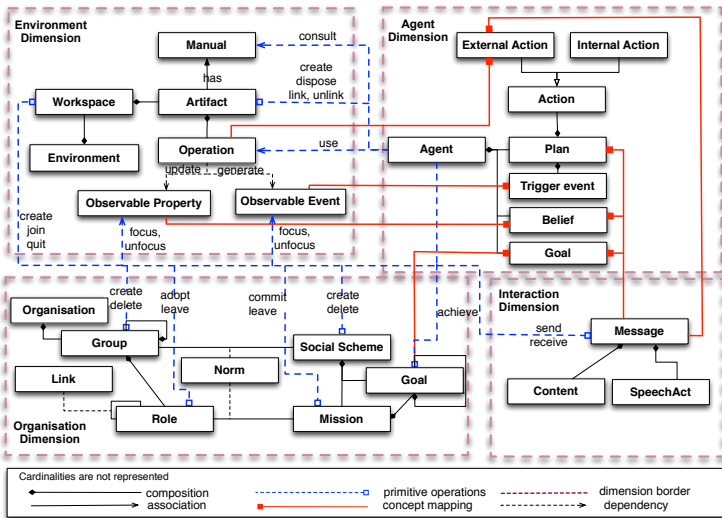
- ▶ Agents with dynamic action repertoire, extended/reshaped by agents themselves
- ▶ Uniform implementation of any mechanisms (e.g. coordination mechanism) in terms of actions/percepts
 - ▶ No need to extend agents with special purpose primitives
- ▶ Exploiting a new type of agent modularity, based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

Organisation meta-model



Simplified *Moise* meta-model [Hübner et al., 2009a]

A & E & O Interaction meta-model

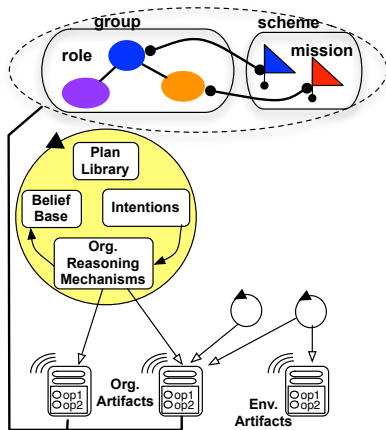


Based on Cartago [Ricci et al., 2009b], Jason [Bordini et al., 2007],
Moise [Hübner et al., 2009a] meta-models

A & O Integration

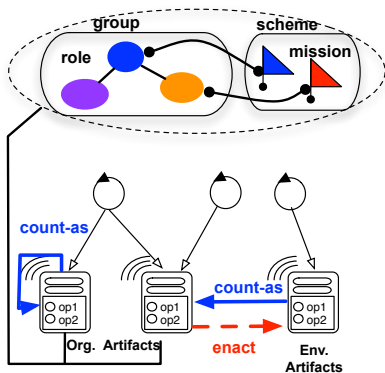
- ▶ Definition of organisational beliefs, organisational actions mediating the perception and actions on the organisation entity
- ▶ Done by instrumenting the organisation management by dedicated **Organisational Artifacts**
 - ▶ Mapping of the **organisational state** onto artifacts computational state
 - ▶ 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]

A & O Integration (2)



- ▶ 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

E & O Integration



- ▶ 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)

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

Functional Organisation life-cycle

- ▶ Creation/End of a schema
- ▶ Commitment/Release of a mission
- ▶ Change of a global goal state

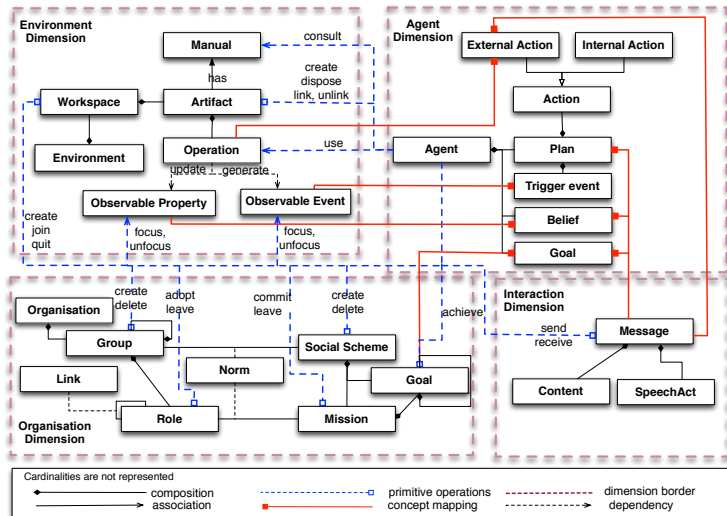
Normative Organisation life-cycle

- ▶ Activation/De-activation of obligation
- ▶ Fulfilment/Violation/Sanction

Outcomes of A & E & O Integration

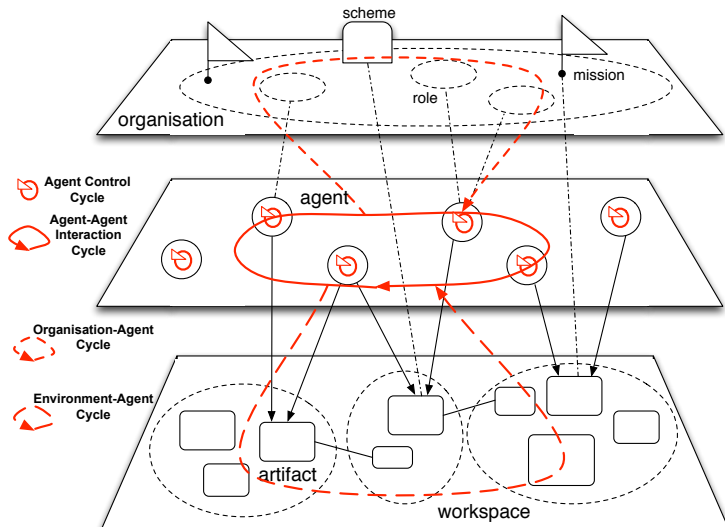
- ▶ Normative deliberative agents
 - ▶ possibility to define mechanisms for agents to evolve within an organisation/several organisations
 - ▶ possibility to define proper mechanisms for deliberating on the internalisation/adoption/violation of norms
- ▶ Reorganisation, adaptation of the organisation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

Synthesis: MAO Dynamics



Outline

Introduction

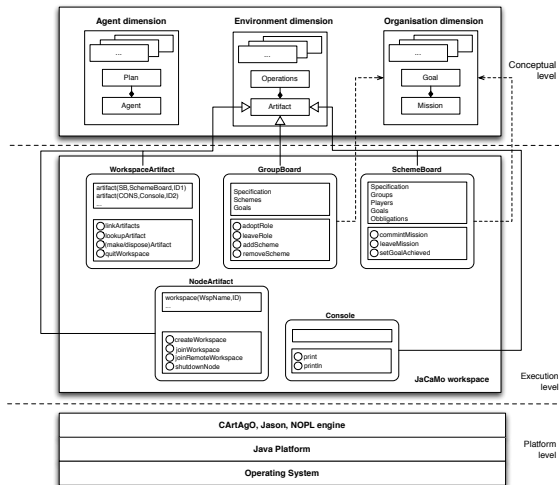
Motivation

Multi-Agent Oriented Programming (MAOP)

MAOP Perspective in the JaCaMo Platform

JaCaMo Platform

<http://jacamo.sourceforge.net>



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")`

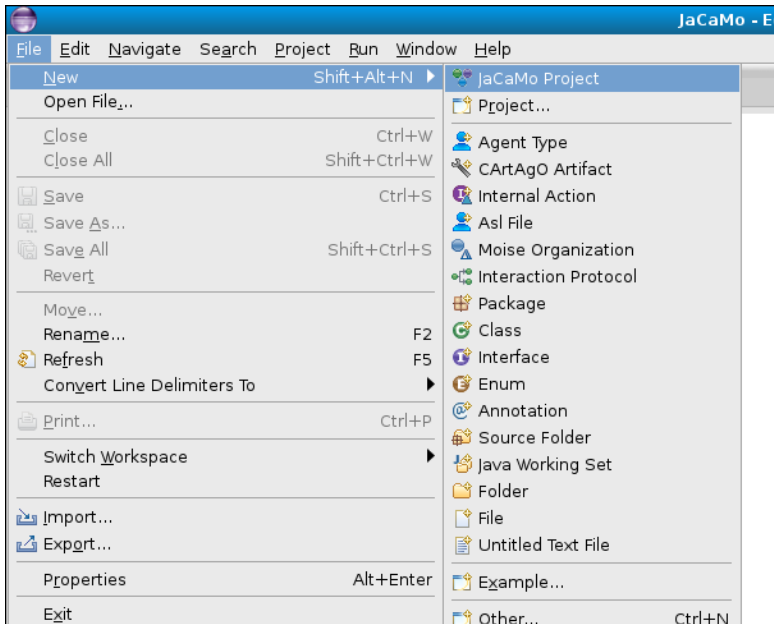
MAS Configuration & Deployment Language

- ▶ Simple way of defining a multi-agent system within the JaCaMo Platform

Example (Building House Definition)

```
mas house_building {  
  agent giacommo    // 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**

The screenshot displays the 'Mind Inspector' application window. On the left, a sidebar lists agents, with 'orgmajordomo' selected. The main panel, titled 'Agent Inspection', shows the internal state of this agent. The title 'Inspection of agent orgmajordomo' is in red. Under the 'Beliefs' section, a list of beliefs is shown in green text, each followed by a truncated representation '[_]'. The beliefs include commitments for different languages (Italian, French, Brazilian) and plans, a current workspace identifier, formation status, goal states for various greetings, group information, and identifiers for specific objects and schedules. At the bottom, an 'Agent History' section shows 'Cycle 0'. The footer contains controls for running the simulation, including a 'Run' button, a cycle count of '1', a duration of 'cycle(s) for', a dropdown for 'all agents', and a 'view as:' dropdown set to 'html'.

Agents

- majordomo
- orgmajordomo

Agent Inspection

Inspection of agent orgmajordomo

- Beliefs

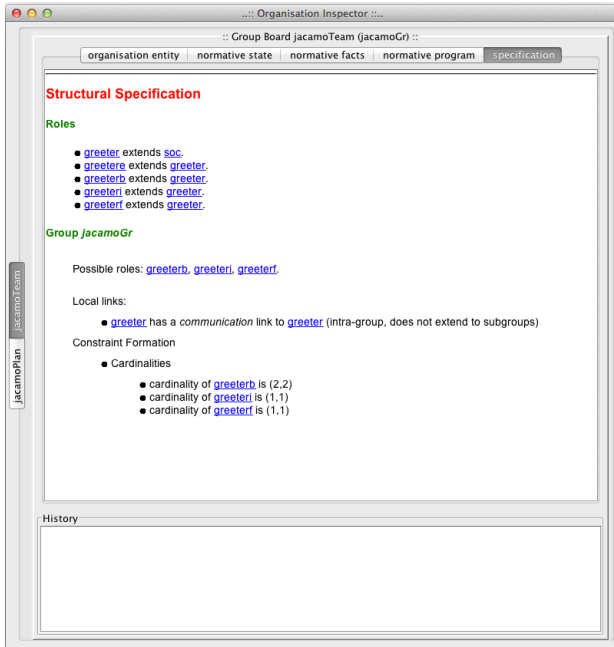
- commitment(italian,mItaly,"jacamoPlan")[_]
- commitment(french,mFrench,"jacamoPlan")[_]
- commitment(brazilian1,mBrazil,"jacamoPlan")[_]
- commitment(brazilian2,mBrazil,"jacamoPlan")[_]
- current_wsp(cobj_1,"server","427dd8d5-408e-431a-a702-7b11ce574e09")[_]
- formationStatus(ok)[_]
- goalState("jacamoPlan",greetings,[french],[french],satisfied)[_]
- goalState("jacamoPlan",greetings_uk_done,[italian],[italian],satisfied)[_]
- goalState("jacamoPlan",greetings_italy_done,[italian],[italian],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)[_]

Agent History

Cycle 0

Run 1 cycle(s) for all agents view as: html

Organization Structure **inspector**



The screenshot shows the 'Organisation Inspector' application window. The title bar reads '...: Organisation Inspector ...:'. The main window has a tabbed interface with the following tabs: 'organisation entity', 'normative state', 'normative facts', 'normative program', and 'specification'. The 'specification' tab is currently selected. The main content area is titled 'Structural Specification' in red. Below this, there is a section for 'Roles' with a list of five items: 'greeter' extends 'soc.', 'greeterc' extends 'greeter.', 'greeterb' extends 'greeter.', 'greeteri' extends 'greeter.', and 'greeterf' extends 'greeter.'. Below the roles section is a section for 'Group jacamoGr' with the following information: 'Possible roles: greeterb, greeteri, greeterf.', 'Local links:', and 'Constraint Formation' which includes 'Cardinalities' with three items: 'cardinality of greeterb is (2,2)', 'cardinality of greeteri is (1,1)', and 'cardinality of greeterf is (1,1)'. At the bottom of the window is a 'History' section which is currently empty. On the left side of the window, there is a vertical sidebar with two buttons: 'jacamoPlan' and 'jacamoTeam'.

...: Organisation Inspector ...:

:: Group Board jacamoTeam (jacamoGr) ::

organisation entity | normative state | normative facts | normative program | **specification**

Structural Specification

Roles

- [greeter](#) extends [soc.](#)
- [greeterc](#) extends [greeter.](#)
- [greeterb](#) extends [greeter.](#)
- [greeteri](#) extends [greeter.](#)
- [greeterf](#) extends [greeter.](#)

Group *jacamoGr*

Possible roles: [greeterb](#), [greeteri](#), [greeterf](#).

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 [greeterf](#) is (1,1)

History

Organization Structure **inspector**

The screenshot shows a window titled "Organisation Inspector". Inside, there's a header "Group Board jacamoTeam (jacamoGr)". Below this is a tabbed interface with five tabs: "organisation entity", "normative state", "normative facts", "normative program", and "specification". The "organisation entity" tab is selected, displaying information about the "jacamoTeam (group)".

jacamoTeam (group)

created from specification [jacamoGr](#) (root group) - owner is [orgmajordomo](#)

Formation:

ok

Players

- [brazilian1](#) plays [greeterb](#)
- [brazilian2](#) plays [greeterb](#)
- [french](#) plays [greeterf](#)
- [italian](#) plays [greeteri](#)

Responsible for the following schemes:

- [jacamoPlan](#)

On the left side of the window, there is a vertical sidebar with two buttons: "jacamoTeam" (highlighted) and "jacamoPlan".

At the bottom of the main content area, there is a section titled "History" with an empty box below it.

Organization Functioning **inspector**

...: Organisation Inspector ...

... Scheme Board jacamoPlan (jacamoSch) ...

organisation entity normative state normative facts normative program **specification**

Functional Specification

Scheme *jacamoSch*

goal	mission	type	# agents that should satisfy	tff	description	arguments	plan
greetings	mFrench	achievement	all				greetings_frar greetings_bra greetings_italy eetings_uk_dc
greetings_france_done	mFrench	achievement	all		Greetings from France		
greetings_brazil_done	mBrazil	achievement	all		Greetings from Brazil		
greetings_italy_done	mItaly	achievement	all		Greetings from France		
greetings_uk_done	mItaly	achievement	all		Greetings from UK		

History

```

created: obligation(french,n1,committed(french,mFrench,"jacamoPlan"),1411504910034){c
created: obligation(brazilian1,n2,committed(brazilian1,mBrazil,"jacamoPlan"),14115049
created: obligation(brazilian2,n2,committed(brazilian2,mBrazil,"jacamoPlan"),14115049
created: obligation(italian,n3,committed(italian,mItaly,"jacamoPlan"),1411504910098){
created: obligation(french,ngoal("jacamoPlan",mFrench,greetings_france_done),achieved
created: obligation(brazilian1,ngoal("jacamoPlan",mBrazil,greetings_brazil_done),achi
created: obligation(brazilian2,ngoal("jacamoPlan",mBrazil,greetings_brazil_done),achi
created: obligation(italian,ngoal("jacamoPlan",mItaly,greetings_italy_done),achieved(
created: obligation(italian,ngoal("jacamoPlan",mItaly,greetings_uk_done),achieved("ja
    
```

Organization Functioning **inspector**

The screenshot shows the 'Organisation Inspector' application window. The title bar reads '...: Organisation Inspector ...'. The main window has a tabbed interface with the following tabs: 'organisation.entity' (selected), 'normative state', 'normative facts', 'normative program', and 'specification'. The selected tab displays the 'jacamoPlan (scheme instance)' details.

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 [mItaly](#)

goal	state	committed/achieved by	arguments	plan
greetings	satisfied	[french]/[french]		= greetings_france greetings_brazil reetings_italy_done etings_uk_done
greetings_france_done	satisfied	[french]/[french]		
greetings_brazil_done	satisfied	[brazilian1,brazilian2]/[brazilian1,brazilian2]		
greetings_italy_done	satisfied	[italian]/[italian]		
greetings_uk_done	satisfied	[italian]/[italian]		

History

```
created: obligation(french,n1,committed(french,mFrench,"jacamoPlan"),1411504910034){c
created: obligation(brazilian1,n2,committed(brazilian1,mBrazil,"jacamoPlan"),14115049
created: obligation(brazilian2,n2,committed(brazilian2,mBrazil,"jacamoPlan"),14115049
created: obligation(italian,n3,committed(italian,mItaly,"jacamoPlan"),1411504910098){
created: obligation(french,ngoal("jacamoPlan",mFrench,greetings_france_done),achieved
created: obligation(brazilian1,ngoal("jacamoPlan",mBrazil,greetings_brazil_done),achi
created: obligation(brazilian2,ngoal("jacamoPlan",mBrazil,greetings_brazil_done),achi
created: obligation(italian,ngoal("jacamoPlan",mItaly,greetings_italy_done),achieved(
created: obligation(italian,ngoal("jacamoPlan",mItaly,greetings_uk_done),achieved("ja
```

Integration of Multi-Agent technologies

- ▶ **A**gent: *Jason* agents [Bordini et al., 2007]
- ▶ **E**nvironment: CArtAgO platform [Ricci et al., 2009b]
- ▶ **O**rganisation: *Moise* framework with the extended/refactored version of the *Moise* OMI: ORA4MAS [Hübner et al., 2009a]
- ▶ **I**nteraction: 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 $\mathcal{J}\text{-Moise}^+$ [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
 - ▶ <http://jaca-web.sourceforge.net>
- ▶ Android Platforms (JaCa-Android)
 - ▶ implementing mobile computing applications on top of the Android platform [Santi et al., 2011]
 - ▶ <http://jaca-android.sourceforge.net>
- ▶ Web Services
 - ▶ building SOA/Web Services applications [Ricci et al., 2010a]
 - ▶ <http://cartagows.sourceforge.net>
- ▶ Arduino Platforms
 - ▶ building “Web of Things” Applications
 - ▶ <http://jacamo.sourceforge.net>
- ▶ Semantic Technologies
 - ▶ JaSA: Semantically Aware Agents
 - ▶ <http://cartago.sourceforge.net>

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

February 2017

Bibliography I



Boissier, O. (2003).

Contrôle et coordination orientés multi-agents.

Habilitation à diriger des recherches, ENS Mines Saint-Etienne et Université Jean Monnet.



Boissier, O., Bordini, R. H., Hübner, J. F., Ricci, A., and Santi, A. (2011).

Multi-agent oriented programming with jacamo.

Science of Computer Programming, pages –.



Bordini, R. H., Hübner, J. F., and Wooldrige, M. (2007).

Programming Multi-Agent Systems in AgentSpeak using Jason.

Wiley Series in Agent Technology. John Wiley & Sons.



Campos, J., López-Sánchez, M., Rodriguez-Aguilar, J. A., and Esteva, M. (2009).

Formalising situatedness and adaptation in electronic institutions.

In *Coordination, Organizations, Institutions and Norms in Agent Systems IV*, volume 5428/2009 of *LNCS*. Springer Berlin / Heidelberg.

Bibliography II



Ciortea, A. (2011).

Modeling relationships for privacy preservation in virtual communities.
Master's thesis, University Politehnica of Bucharest.



de Brito, M., Hübner, J. F., and Boissier, O. (2015).

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.



de Brito, M., Hübner, J. F., and Bordini, R. H. (2012).

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



Demazeau, Y. (1995).

From interactions to collective behaviour in agent-based systems.
In *Proc. of the 1st European Conf. on Cognitive Science. Saint-Malo*, pages
117–132.

Bibliography III



Demazeau, Y. (1997).

Steps towards multi-agent oriented programming.

(slides Workshop) 1st International Workshop on Multi-Agent Systems, IWMAS'97, Boston.



Hübner, J. F., Boissier, O., Kitio, R., and Ricci, A. (2009a).

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. (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.



Hübner, J. F., Vercouter, L., and Boissier, O. (2009b).

Instrumenting Multi-Agent Organisations with reputation artifacts.

In Hubner, J. F., Matson, E., Boissier, O., and Dignum, V., editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems IV*, volume 5428 of *LNAI*, pages 96–110. Springer.

Bibliography IV



Huhns, M. N. (2001).

Interaction-oriented programming.

In *First international workshop, AOSE 2000 on Agent-oriented software engineering*, pages 29–44, Secaucus, NJ, USA. Springer-Verlag New York, Inc.



Kitio, R. (2011).

Gestion de l'ouverture au sein d'organisations multi-agents. Une approche basée sur des artefacts organisationnels.

PhD thesis, ENS Mines Saint-Etienne.



Occello, M., Baeijs, C., Demazeau, Y., and Koning, J.-L. (2004).

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.



Omicini, A., Ricci, A., and Viroli, M. (2008).

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

Autonomous Agents and Multi-Agent Systems, 17(3):432–456.

Bibliography V



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., Tambe, M., Chauvat, N., and Cavedon, L. (1999).

Toward team-oriented programming.

In Jennings, N. R. and Lespérance, Y., editors, ATAL, volume 1757 of LNCS, pages 233–247. Springer.



Ricci, A., Denti, E., and Piunti, M. (2010a).

A platform for developing SOA/WS applications as open and heterogeneous multi-agent systems.

Multiagent and Grid Systems International Journal (MAGS), Special Issue about “Agents, Web Services and Ontologies: Integrated Methodologies”.

To Appear.

Bibliography VI



Ricci, A., Piunti, M., and Viroli, M. (2009a).

Externalisation and internalization: A new perspective on agent modularisation in multi-agent system programming.

In Dastani, M., Fallah-Seghrouchni, A. E., Leite, J., and Torroni, P., editors, *LADS*, volume 6039 of *Lecture Notes in Computer Science*, pages 35–54. Springer.



Ricci, A., Piunti, M., and Viroli, M. (2010b).

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

Autonomous Agents and Multi-Agent Systems.

Published Online with ISSN 1573-7454 (will appear with ISSN 1387-2532).



Ricci, A., Piunti, M., Viroli, M., and Omicini, A. (2009b).

Environment programming in CArtAgO.

In *Multi-Agent Programming: Languages, Platforms and Applications, Vol.2*. Springer.

Bibliography VII



Ricordel, P. and Demazeau, Y. (2002).

VOLCANO: a vowels-oriented multi-agent platform.

In Dunin-Keplicz and Nawarecki, editors, *Proceedings of the International Conference of Central Eastern Europe on Multi-Agent Systems (CEEMAS'01)*, volume 2296 of *LNAI*, pages 252–262. Springer Verlag.



Santi, A., Guidi, M., and Ricci, A. (2011).

Jaca-android: An agent-based platform for building smart mobile applications.

In Dastani, M., Fallah-Seghrouchni, A. E., Hübner, J., and Leite, J., editors, *Languages, Methodologies and Development Tools for Multi-agent systems*, volume 6822 of *LNAI*. Springer Verlag.



Shoham, Y. (1993).

Agent-oriented programming.

Artif. Intell., 60(1):51–92.



Sorici, A. (2011).

Agile governance in an ambient intelligence environment.

Master's thesis, University Politehnica of Bucharest.

Bibliography VIII



Stratulat, T., Ferber, J., and Tranier, J. (2009).

MASQ: towards an integral approach to interaction.

In *AAMAS* (2), pages 813–820.