Multi-Agent Programming

– Conclusion & Perspectives –

O. Boissier

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

CPS2 M1 – Fall 2020

Outline

Outline

Conclusions

- MAS is not only agents
- MAS is not only organisation
- MAS is not only environment
- MAS is not only interaction

- MAS has many dimensions
- All as **first class entities**
- Multi-Agent Oriented Programming proposes a seamless integration of these dimensions

Benefits of MAOP

- **Separation of concerns** between Agent, Environment, Organisation and Interaction
  - using the best abstraction level and tools to tackle the specific dimensions,
  - avoiding design pitfalls, such as using agents to implement either non-autonomous entities (e.g., a blackboard agent) or a collection of autonomous entities (group agent)
- **Openness and heterogeneity**
  - Heterogeneous agents working in the same organisation,
  - Heterogeneous agents working in the same environment,
  - The same agent working in different and heterogeneous organisations,
  - The same agent working in different heterogeneous environments
- **Flexibility, adaptation**
  - rich coordination and regulation driven by agents, environment, interactions or organisations
- **Inclusion of Physical, Digital and Human worlds to define socio-cognitive, physical and digital systems**

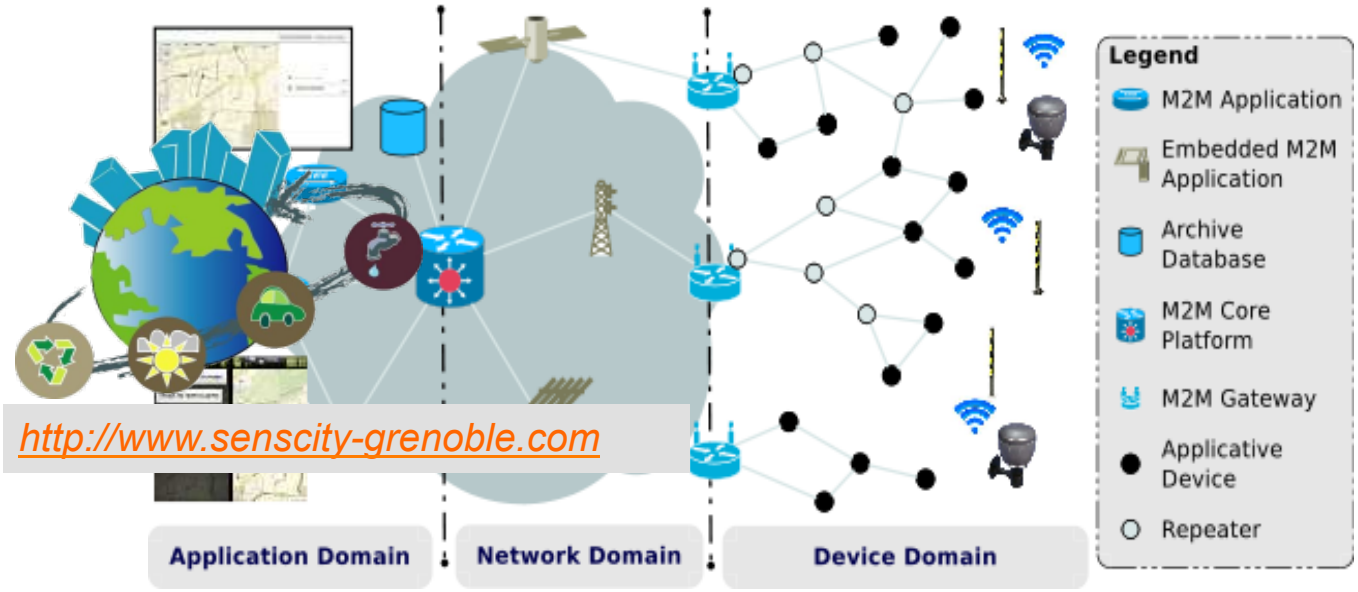
From an engineering point of view

MAOP brings **programming features** such as:

- Modularity (Agents, Artifacts, Workspace, Organization, Group, Role, ...)
- Extensibility (in terms of actions and tools provided to Agents, of agents, of organisations, ...)
- Reusability of the different constructs
- Readability, code is cleaner and more understandable ...
- Self-adaptation (Reorganization Artifact, Organization Management Artifacts, ...)
- ...

Example: Agile governance of M2M infrastructure

Conclusions



M2M Infrastructure for Smart Cities (ETSI view)

- **Device Domain**: smart devices (sensors and actuators) for collecting data and controling the environment
- **Network Domain**: shared communication infrastructure (platforms and gateways) to connect applications to devices
- **Application Domain**: applications providing ubiquitous & added value services to citizens

Objective

Example: Agile governance of M2M infrastructure

- Openness
  - Addition or deletion of stakeholders (i.e. application/sensors/actuators) during the lifetime of the system
- Adaptation
  - Reacting to the changes of environmental conditions (e.g. increase in the number of collected data, number of messages)
- Shifting from "vertical" to "horizontal" M2M infrastructures
  - Sharing of costs and resources between several applications (e.g. Parking Management, Garbage Collection, Smart Metering, etc)

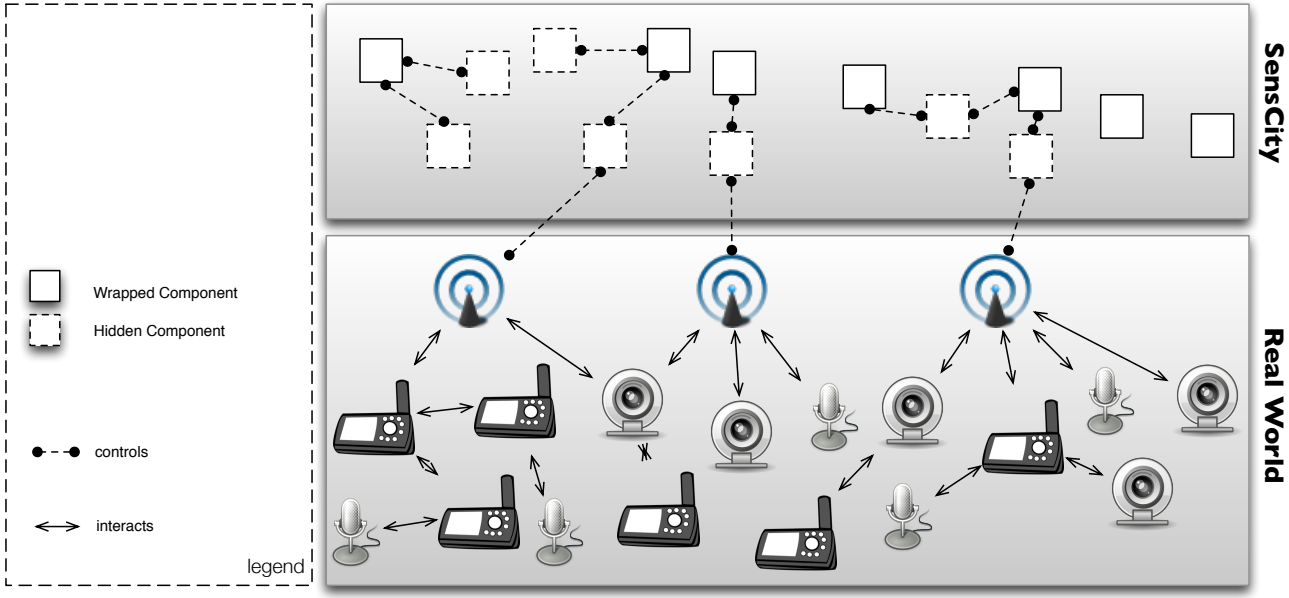
$\rightsquigarrow$ Definition of an agile and decentralized governance layer on top of the M2M Infrastructure [Persson et al., 2012]

Legacy system: SensCity for Smart Cities

Example: Agile governance of M2M infrastructure

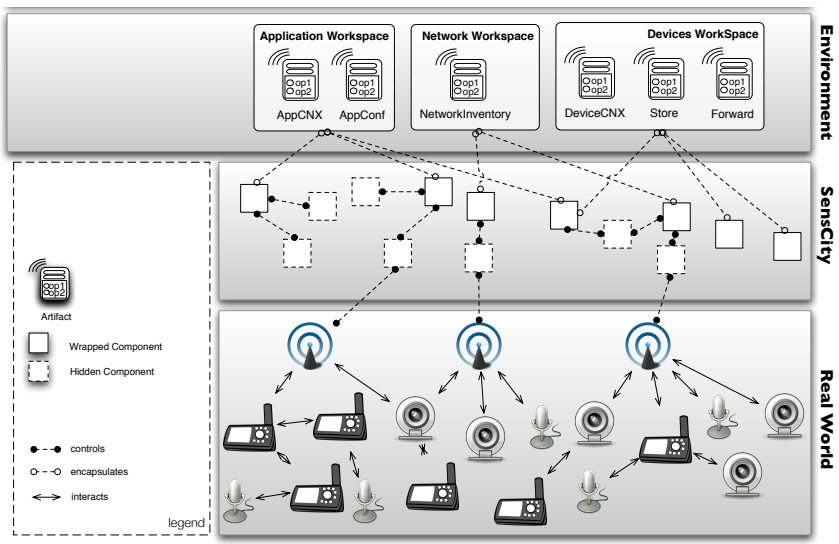
SensCity platform, a legacy system to govern,

- composed of 47 types of components
- supporting access to physical devices
- and shared by several heterogenous applications



Programmed environment

Example: Agile governance of M2M infrastructure

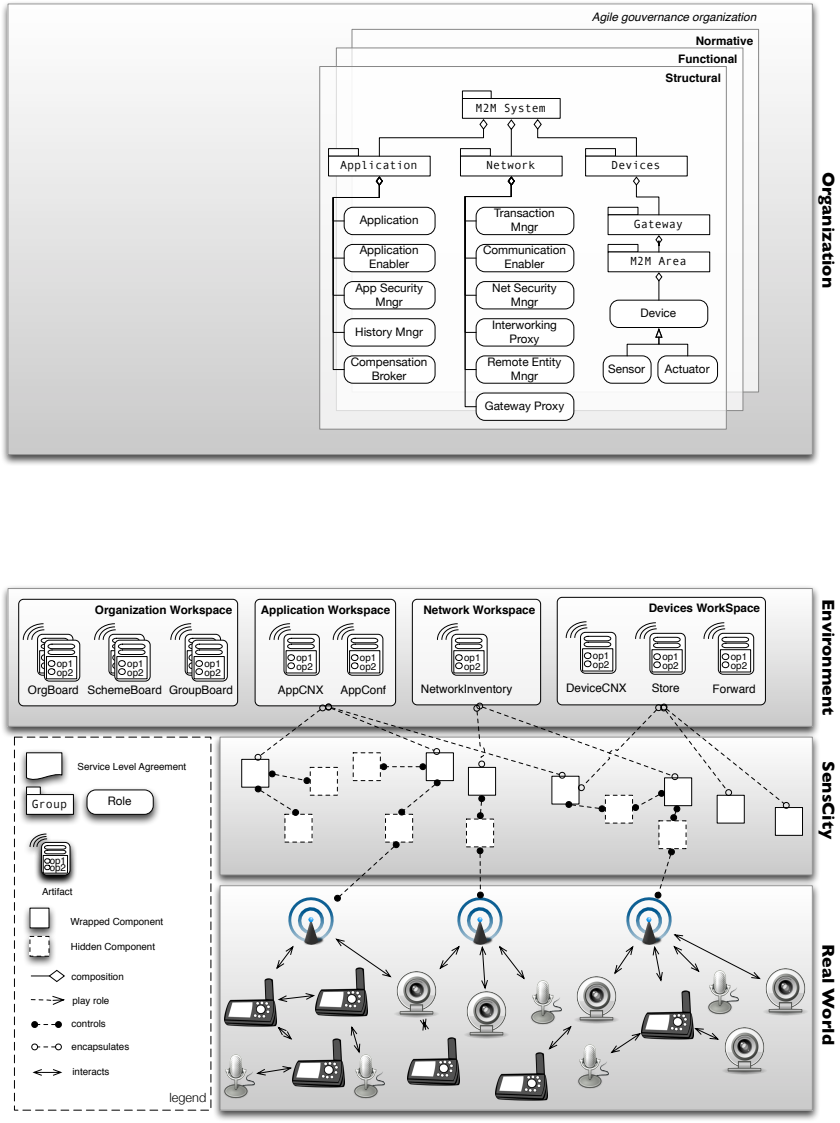


Workspaces structure the environment according to the ETSI domains Governance Artifacts as monitoring and controling tools that:

- Compute direct / interpreted / aggregated status of the SensCity Platform's components.
- Expose their public state through observable properties or signals
- May be used through a repertoire of operations (e.g. threshold definition, Load balancing, (de)activate) enabling to act on the functioning of the M2M infrastructure

Programmed organisations

Example: Agile governance of M2M infrastructure



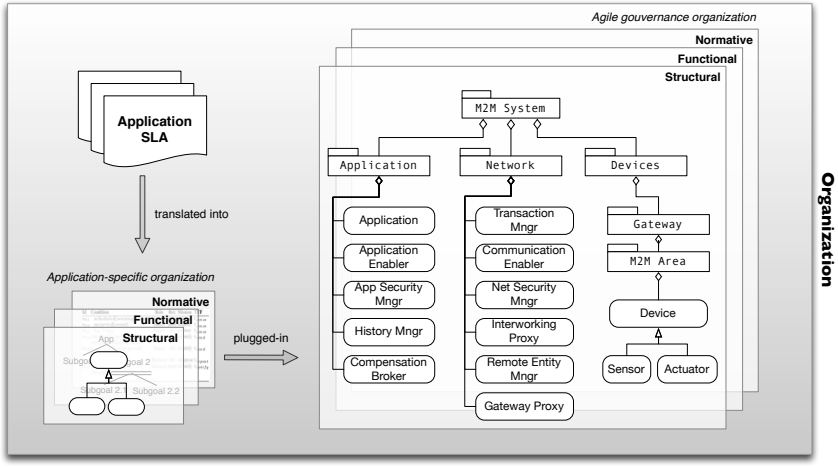
Governance organisations are:

- Horizontal organisation: defines the nominal global functioning based on ETSI standards (Domains, Service Capabilities, Capability functionalities, ...)
- Vertical organisations: based on each application SLA deployed on the Urban M2M Infrastructure

  ⇝ agents participate to multiple organisations

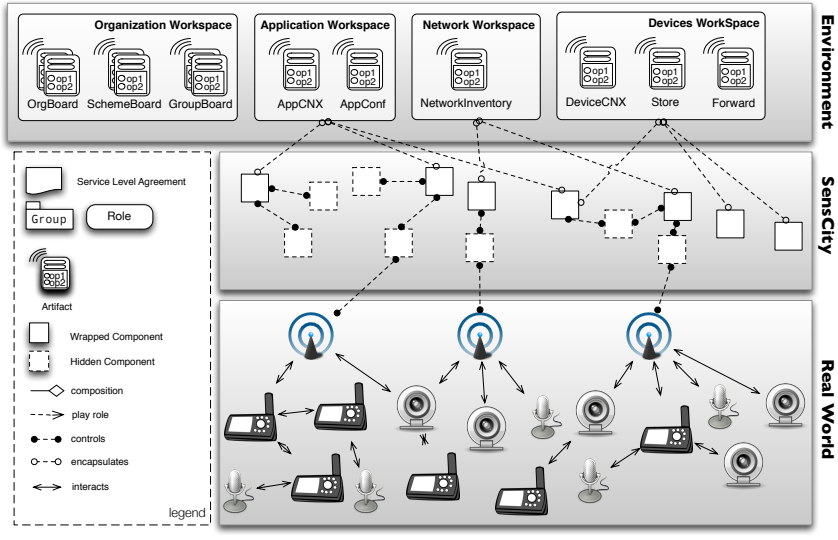
Programmed organisations

Example: Agile governance of M2M infrastructure



Governance organisations are:

- Horizontal organisation: defines the nominal global functioning based on ETSI standards (Domains, Service Capabilities, Capability functionalities, ...)

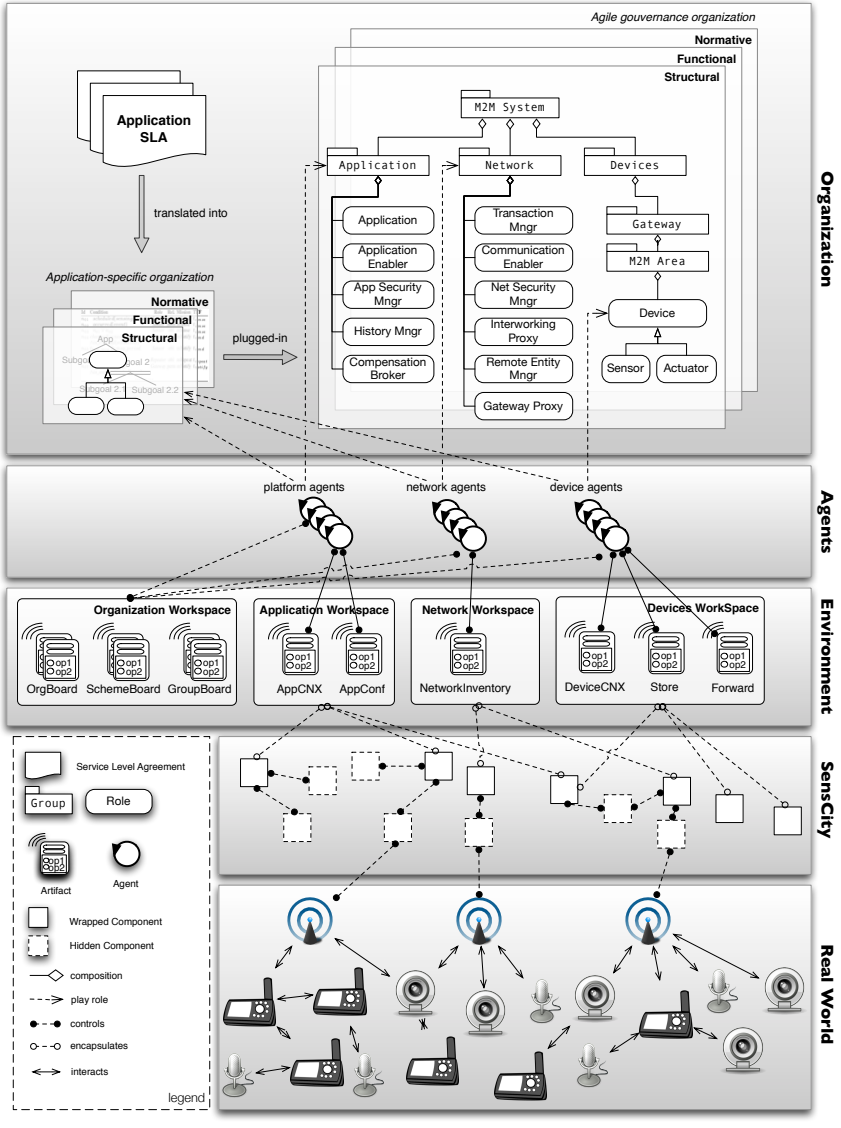


- Vertical organisations: based on each application SLA deployed on the Urban M2M Infrastructure

↝ agents participate to multiple organisations

Programmed agents

Example: Agile governance of M2M infrastructure

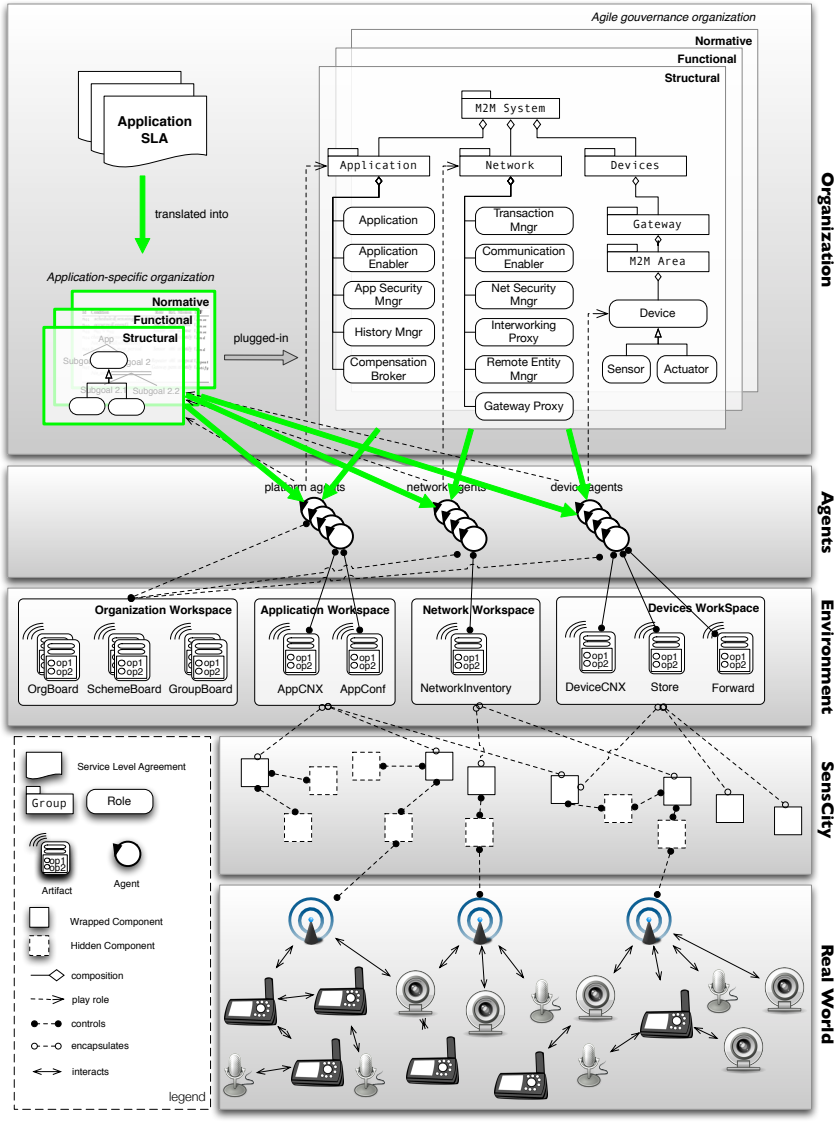


Governance agents:

- Monitor the M2M infrastructure by focusing on artifacts (e.g. Failures, Overloads) updating their beliefs, goals
- Decide locally and autonomously about execution of actions to control M2M infrastructure given their governance policies and local strategies (goals, beliefs, plans), the organisations ($\rightsquigarrow$ artifacts' operations)
- Interact with the other agents
- Adapt the global governance strategy ($\rightsquigarrow$ reorganisation)

M2M Agile Governance Execution

Example: Agile governance of M2M infrastructure

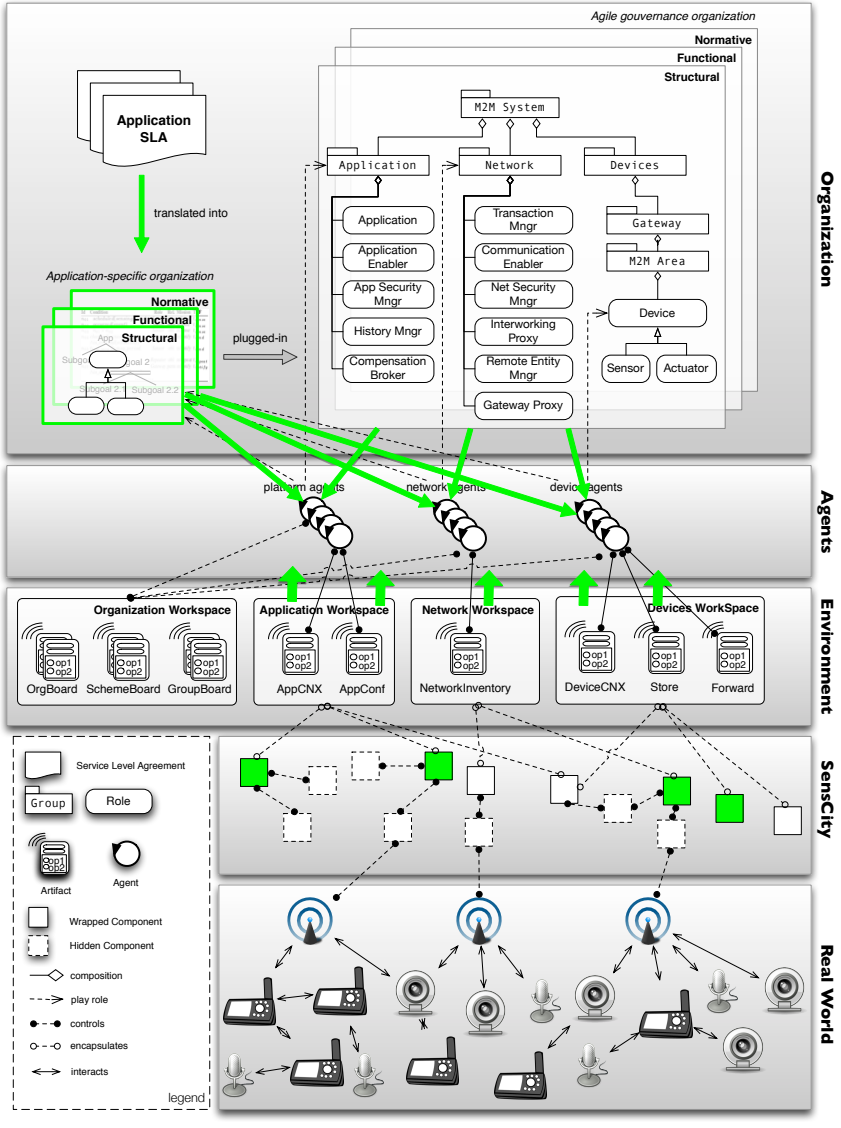


SLA definition

- Service Level Agreement: Translation of Application subscription to Devices in terms of Group, Roles, Missions, Norms
- Validation: If judged feasible, role adopted else rejected or new proposal done

M2M Agile Governance Execution

Example: Agile governance of M2M infrastructure

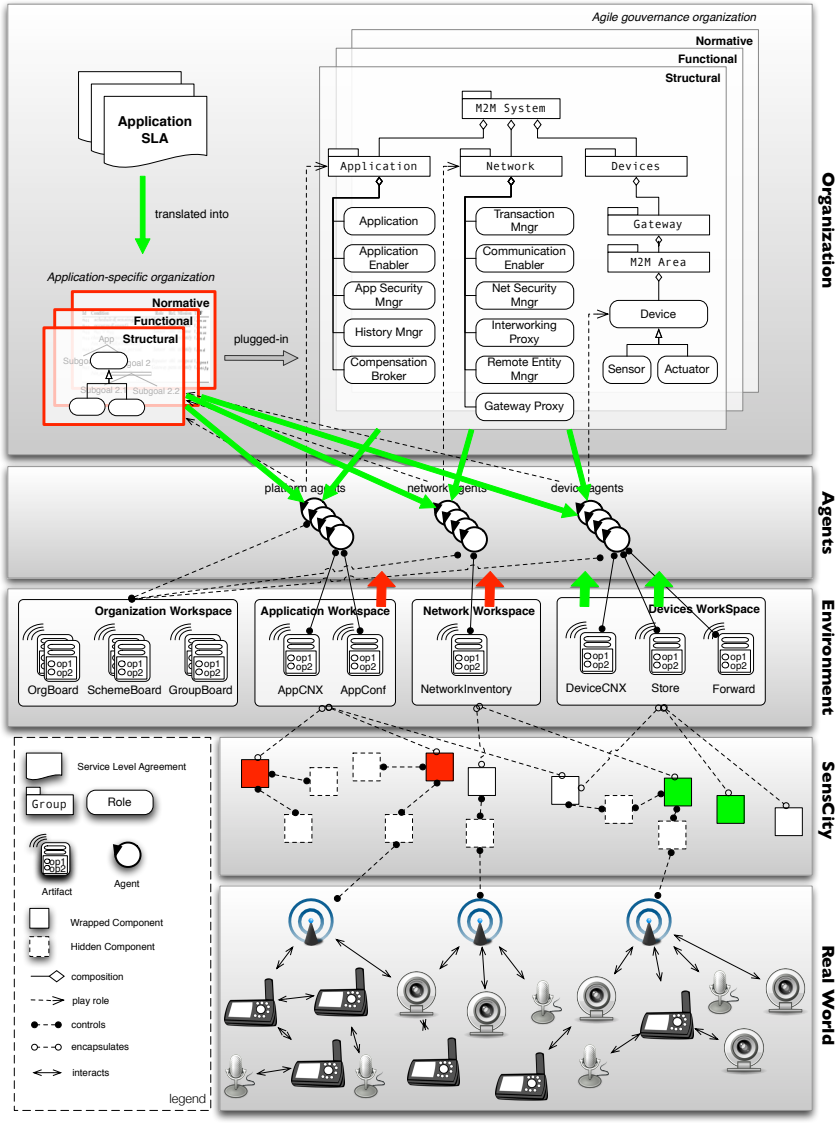


Contract Execution

- Social Scheme activation: when request is received, new scheme instance is started
- Execution Monitoring: validation of application requests, monitoring of devices' activity

M2M Agile Governance Execution

Example: Agile governance of M2M infrastructure

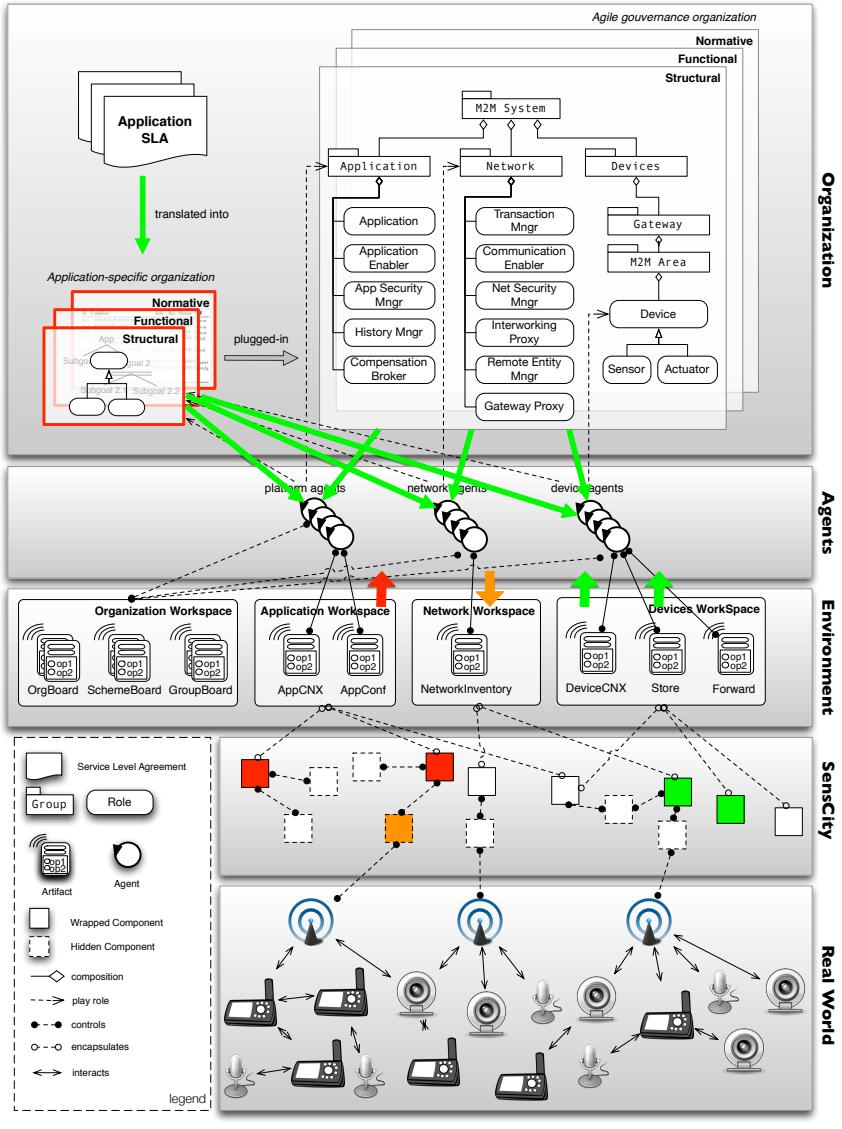


Problem Detection

- Detection of possible norm violation, i.e. failure to comply with the SLA
- Platform monitoring: find the source of the problem with the help of the artifacts

M2M Agile Governance Execution

Example: Agile governance of M2M infrastructure

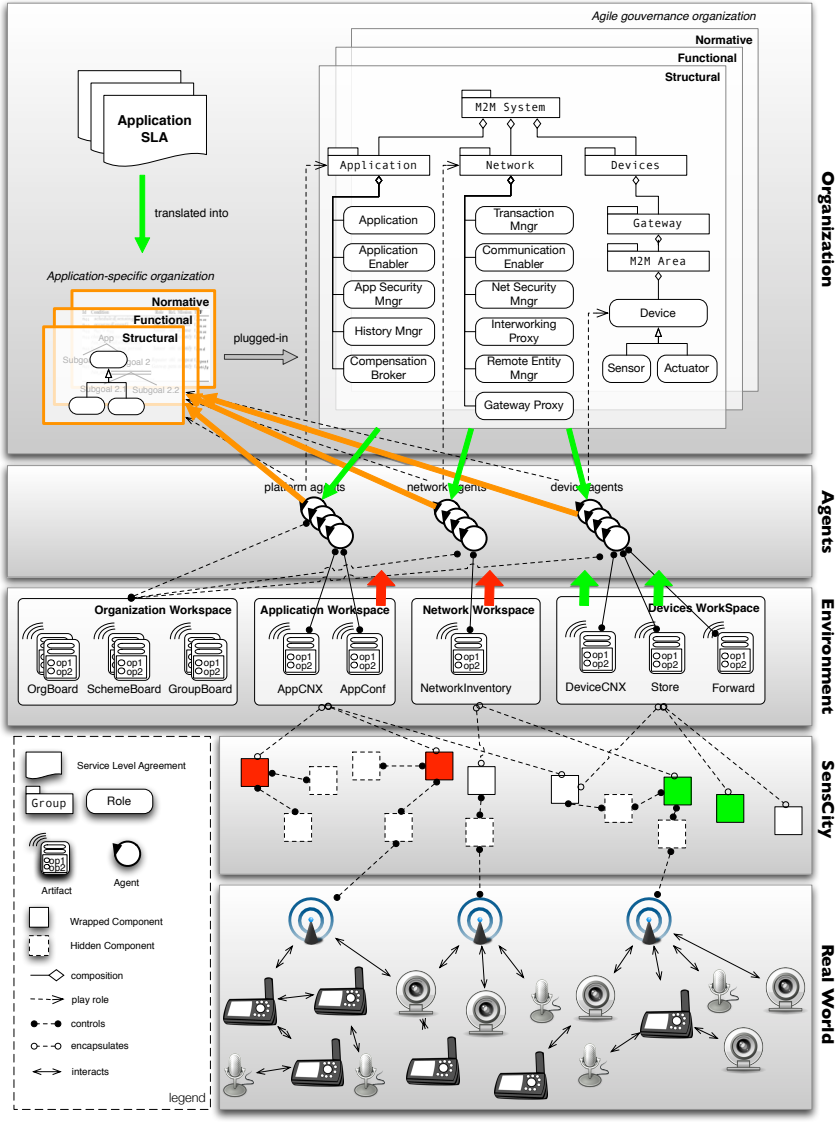


Infrastructure Adaptation

- Fix the problem by acting on the platform via the artifacts under their responsibility and their governance policies
- Coordinate with each other

M2M Agile Governance Execution

Example: Agile governance of M2M infrastructure



Governance Strategy Adaptation

- Problem with the SLA: SLA might be too greedy, infrastructure adaptation not sufficient

$\rightsquigarrow$ SLA redefinition (eg. decrease message frequency, device subscription redefined)

Synthesis

Example: Agile governance of M2M infrastructure

- Monitoring and governance of the M2M infrastructure take place at different levels embracing an increasing broader view: Artifacts $<$ Agents & Interactions $<$ Organisations

$\rightsquigarrow$ Modularity / Lisibility of the Governance Layer

- Coordination mechanisms installing Top-Down - Bottom-up loops (Synergie between Macro/Micro levels)

- Agents are in charge of the governance but also of the reorganisation process

Outline

MAOP Open Issues & Perspectives

- $\rightsquigarrow$ Coordination / Regulation
  - Integration of Bottom-up AND Top-Down functioning within MAS $\rightsquigarrow$ integration of emergence AND Normative dynamics
  - Management of Open Organisations, Multiple Organisations, Organisation of Organisations
  - Management of Situated Organisations (Interactions between E and O dimensions)
- $\rightsquigarrow$ Engineering
  - Debugging, Performance, ...
  - Life cycle of MAS (from requirement to maintenance) $\rightsquigarrow$ software engineering tools and methods
  - Shift from Agent-Oriented Sofware Engineering to **Multi**-Agent Oriented Software Engineering where all the dimensions A, E, I, O may guide each step of the process (cf. [Uez and Hübner, 2014])
  - Evaluation & Verification of MAO programmed applications,
  - Integrating with other technologies
  - Handle Scalability, Robustness

More wide and open issues & perspectives

Some questions from EMAS@AAMAS 2018

Cognitive agent/MAS programming & their application to real-world systems:

- How and is it worth to integrate reasoning and reactive behavior (in a flexible way)?
- Which architectures are most suitable for MAS of different domains?
- Deploying agents/MAS in the real world:
  - How to express the requirements for large-scale and open MAS and how to translate these requirements into agent goals?
  - How to enable agent-based systems to deal with continuous change, for example in the operating environment or user requirements?
- How to ensure/control global behavior of decentralized MAS?

More wide and open issues & perspectives (contd)

Some questions from EMAS@AAMAS 2018

- How to specify, design, implement, verify, test, validate and evolve MAS?
- How to provide guarantees for MAS systems in an efficient way?
- How to tame uncertainty when engineering MAS?
- What does convergence mean in large scale MAS that are subject to change?

More wide and open issues & perspectives (contd)

Some questions from EMAS@AAMAS 2018

MAS and other domains:

- What can MAS engineering gain from ontologies and vice versa?
- What is the synergy between Cloud and Edge computing on the one hand and MAS engineering on the other hand?
- How to seamlessly integrate MAS engineering with mainstream engineering models, languages, frameworks and tools?
- What are the implications of MAS engineering in the context of continuous development and deployment?
- What are opportunities and implications of machine learning & data science for MAS and vice versa?
- How to seamlessly integrate AI and machine learning techniques into design/programming languages and tools for agent-based systems?

Global picture (from AAMAS 2019 Tracks)

Example: Agile governance of M2M infrastructure

- Main Track
- Blue Sky Ideas (5 papers)
- Engineering Multiagent Systems (10 papers)
- Industrial Applications (4 papers)
- Robotics Track (17 papers)
- Socially Interactive Agents (17 papers)

Global picture (from AAMAS 2018, 2019 Topics of Interests)

Example: Agile governance of M2M infrastructure

- **Agent Theories and Models**: Belief-Desire-Intention theories and models, Cognitive models, Models of emotions, ...
- **Communication and Argumentation**: Commitments, Communication languages and protocols, Speech act theory, Argumentation-based dialogue and protocols, ...
- **Agent Cooperation**: Biologically-inspired approaches and methods, Collective intelligence, Distributed problem solving, Teamwork, team formation, teamwork analysis, Coalition formation, ...
- **Knowledge Representation and Reasoning**: Ontologies for agents, Reasoning in agent-based systems, Single and multi-agent planning and scheduling, Reasoning about action, plans and change, Reasoning about knowledge, beliefs, goals and norms, ...

Global picture (from AAMAS 2018, 2019 Topics of Interests) (Contd)

Example: Agile governance of M2M infrastructure

- **Agent Societies and Societal issues**: Organizations and institutions, Social networks, Socio-technical systems, Normative systems, Values in MAS (privacy, safety, security, transparency, ...), Coordination and control models for multiagent systems, Trust and reputation, Policy, regulation and legislation, Self-organization

- **Learning and Adaptation**: Reward structures for learning, Evolutionary algorithms, Co-evolutionary algorithms, Multiagent learning, Reinforcement learning, Deep learning, Adversarial machine learning, Learning agent capabilities (agent models, communication, observation), Learning agent-to-agent interactions (negotiation, trust, coordination)

- **Agents & Mainstream Computing**: Service-oriented architectures, Mobile agents, Autonomic computing, P2P, web services, grid computing, IoT, HPC

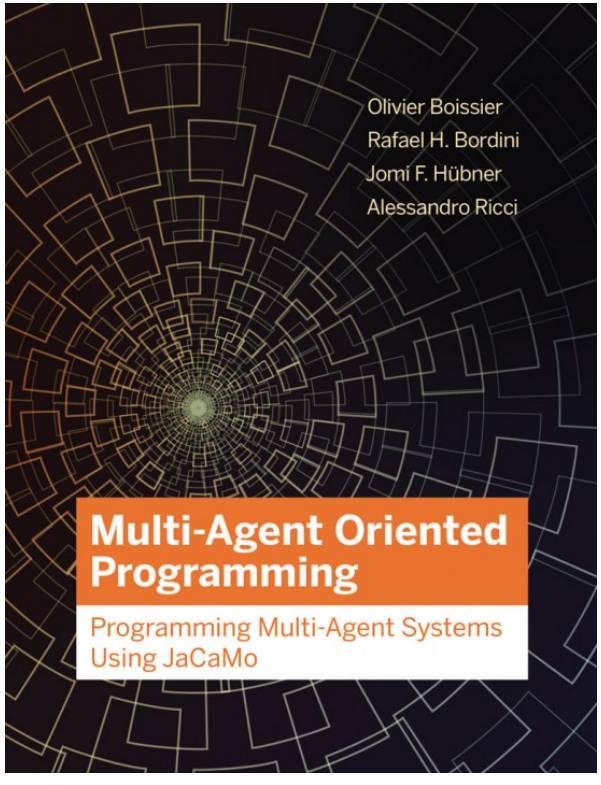
Global picture (from AAMAS 2018, 2019 Topics of Interests) (Contd)

Example: Agile governance of M2M infrastructure

- **Verification and Validation of Agent-based Systems**: Testing of agent-based systems, including model-based testing, Verification techniques for multiagent systems, including model checking, Synthesis of agent-based systems, Fault tolerance and resilience of multi-agent systems, Testing and debugging multiagent programs
- **Agent-based Simulation**: Social simulation, Simulation techniques, tools and platforms, Simulation of complex systems, Validation of simulation systems, Modelling for agent-based simulation, Interactive simulation, Emergent behaviour, Analysis of agent-based simulations
- **Economic Paradigms**: Auctions and mechanism design, Bargaining and negotiation, Behavioral game theory, Cooperative games: theory & analysis, Cooperative games: computation, Noncooperative games: theory & analysis, Noncooperative games: computation, Social choice theory, Game theory for practical applications

Further Resources

- `http://jacamo.sourceforge.net`
- O. Boissier, R.H. Bordini, J.F. Hübner, and A. Ricci
  Multi-Agent Oriented Programming: Programming Multi-Agent Systems Using JaCaMo
  MIT Press, 2020.



Bibliography I

Persson, C., Picard, G., Ramparany, F., and Boissier, O. (2012).

A jacamo-based governance of machine-to-machine systems.

In Demazeau, Y., Müller, J. P., Rodríguez, J. M. C., and Pérez, J. B., editors, *Advances on Practical Applications of Agents and Multiagent Systems, Proc. of the 10th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 12)*, volume 155 of *Advances in Soft Computing Series*, pages 161–168. Springer.

Uez, D. M. and Hübner, J. F. (2014).

Environments and organizations in multi-agent systems: From modelling to code.

In Dalpiaz, F., Dix, J., and van Riemsdijk, M. B., editors, *Engineering Multi-Agent Systems - Second International Workshop, EMAS 2014, Paris, France, May 5-6, 2014, Revised Selected Papers*, volume 8758 of *Lecture Notes in Computer Science*, pages 181–203. Springer.