Exploiting Interaction Affordances: On Engineering Autonomous Systems for the Web of Things

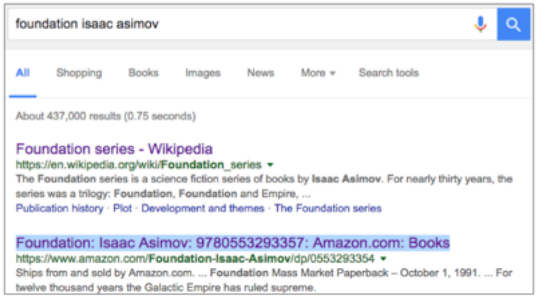
Position Paper for the Second W3C Workshop on the Web of Things

Andrei Ciortea[1,2] Simon Mayer[1,3] Olivier Boissier[4] Fabien Gandon[2]

[1] University of St. Gallen, Switzerland
[2] Wimmics, Inria Sophia Antipolis, Univ. Côte D'Azur, CNRS, France
[3] ETH Zurich, Switzerland
[4] MINES Saint-Étienne, Univ. de Lyon, Laboratoire Hubert Curien, CNRS, France

The World Wide Web -

An Internet-scale hypermedia environment for people



The World Wide Web -

An Internet-scale hypermedia environment for people

Local guidance: hypermedia

The World Wide Web -

An Internet-scale hypermedia environment for people

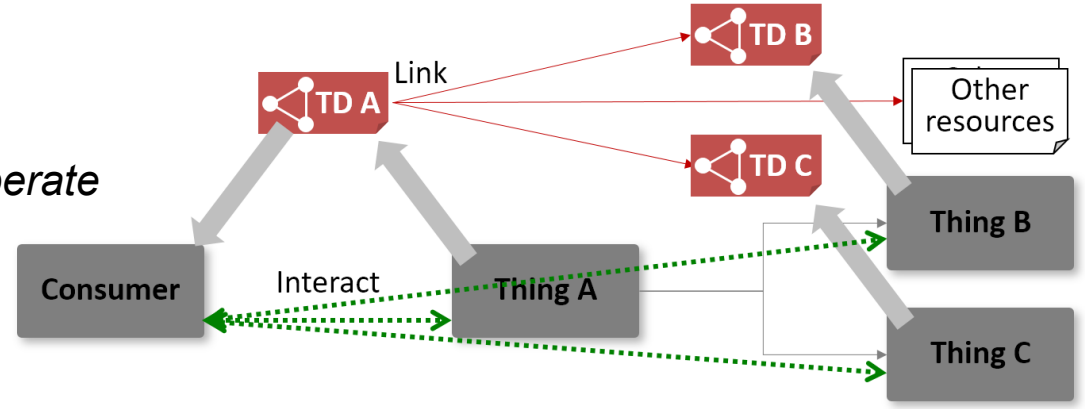
Local guidance: hypermedia

Global guidance: the buyer's goal

The W3C WoT -

An Internet-scale hypermedia environment for consumers

Ideally:
arrive-and-operate

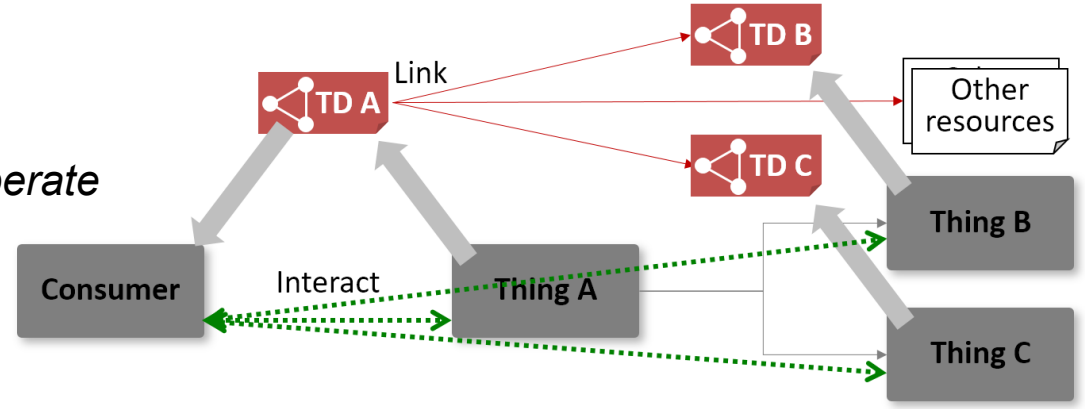


Matthias Kovatsch et al. (eds.), *Web of Things (WoT) Architecture*, W3C Candidate Recommendation

The W3C WoT -

An Internet-scale hypermedia environment for consumers

Ideally:
arrive-and-operate



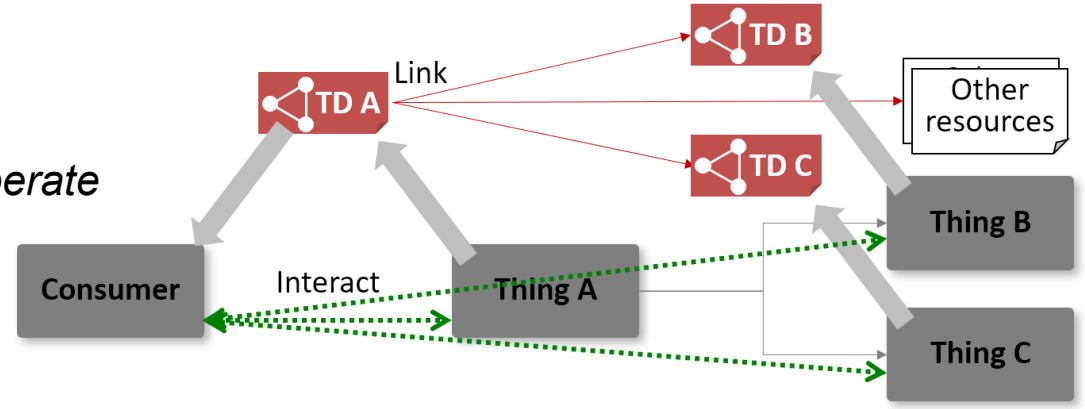
Matthias Kovatsch et al. (eds.), *Web of Things (WoT) Architecture*, W3C Candidate Recommendation

Local guidance: hypermedia

The W3C WoT -

An Internet-scale hypermedia environment for consumers

Ideally:
arrive-and-operate



Matthias Kovatsch et al. (eds.), *Web of Things (WoT) Architecture*, W3C Candidate Recommendation

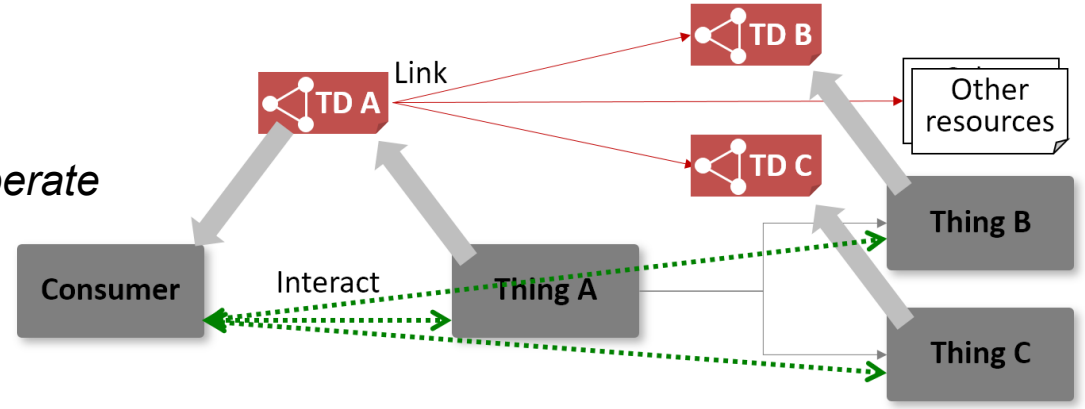
Local guidance: hypermedia

Global guidance: how can consumers achieve their tasks by *navigating* the hypermedia and *deciding autonomously* among the various options presented to them at run time?

The W3C WoT -

An Internet-scale hypermedia environment for consumers

Ideally:
arrive-and-operate



Matthias Kovatsch et al. (eds.), *Web of Things (WoT) Architecture*, W3C Candidate Recommendation

Local guidance: hypermedia

Global guidance: how can consumers achieve their tasks by *navigating* the hypermedia and *deciding autonomously* among the various options presented to them at run time?

- research on **multi-agent systems** already provides solutions to *design*, *program*, *debug*, *monitor*, *regulate*, and *coordinate* autonomous, goal-directed agents

(Multi-)Agent Oriented Programming – (M)AOP

Paradigms & metaphors:

imperative	=>	machines
functional	=>	math
object-oriented	=>	world of objects
(multi-)agent oriented	=>	

Alessandro Ricci, *Levels of Abstraction in Designing and Programming Systems of Cognitive Agents*, First Workshop on Hypermedia Multi-Agent Systems (HyperAgents 2019), co-located with TheWebConf 2019.

(Multi-)Agent Oriented Programming – (M)AOP

Paradigms & metaphors:

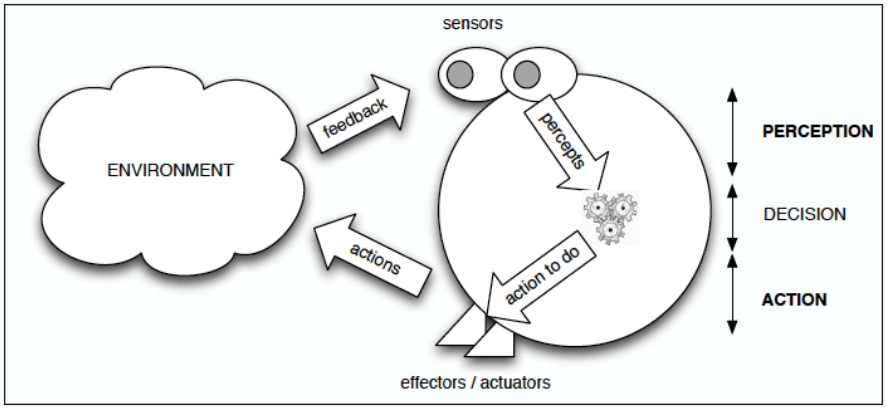
imperative	=>	machines
functional	=>	math
object-oriented	=>	world of objects
(multi-)agent oriented	=>	world of humans

Alessandro Ricci, *Levels of Abstraction in Designing and Programming Systems of Cognitive Agents*, First Workshop on Hypermedia Multi-Agent Systems (HyperAgents 2019), co-located with TheWebConf 2019.

(Multi-)Agent Oriented Programming – (M)AOP

Paradigms & metaphors:

imperative	=>	machines
functional	=>	math
object-oriented	=>	world of objects
(multi-)agent oriented	=>	world of humans



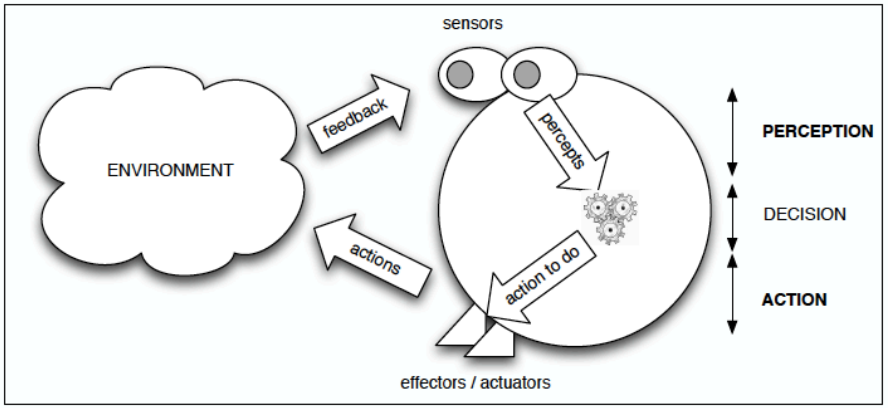
- task/goal-oriented
- pro-active + reactive
- decision making

Alessandro Ricci, *Levels of Abstraction in Designing and Programming Systems of Cognitive Agents*, First Workshop on Hypermedia Multi-Agent Systems (HyperAgents 2019), co-located with TheWebConf 2019.

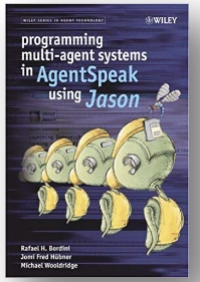
(Multi-)Agent Oriented Programming – (M)AOP

Paradigms & metaphors:

imperative	=>	machines
functional	=>	math
object-oriented	=>	world of objects
(multi-)agent oriented	=>	world of humans



- task/goal-oriented
- pro-active + reactive
- decision making



[Bordini et al., 2007]

Alessandro Ricci, *Levels of Abstraction in Designing and Programming Systems of Cognitive Agents*, First Workshop on Hypermedia Multi-Agent Systems (HyperAgents 2019), co-located with TheWebConf 2019.

(Multi-)Agent Oriented Programming – (M)AOP

Belief-Desire-Intention (BDI) model/architecture (‘80s):

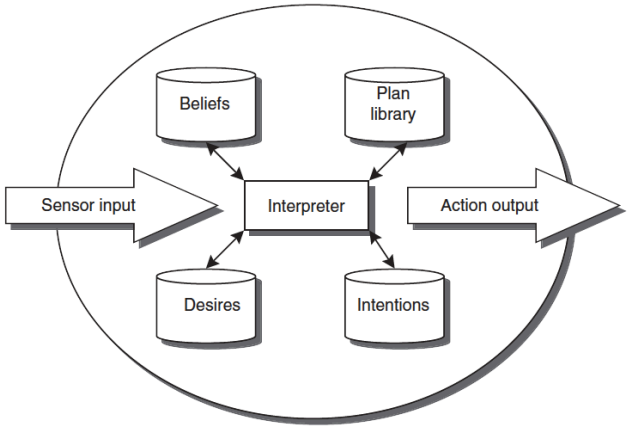
- **beliefs**: information the agent holds about the world (can be out of date or inaccurate);
- **desires**: states of affairs the agent wishes to bring to the world (i.e., the agent’s *goals*)
- **intentions**: desires / goals the agent is committed to achieve

Alessandro Ricci, *Levels of Abstraction in Designing and Programming Systems of Cognitive Agents*, HyperAgents 2019 workshop, co-located with TheWebConf 2019.

(Multi-)Agent Oriented Programming – (M)AOP

Belief-Desire-Intention (BDI) model/architecture (‘80s):

- **beliefs**: information the agent holds about the world (can be out of date or inaccurate);
- **desires**: states of affairs the agent wishes to bring to the world (i.e., the agent’s *goals*)
- **intentions**: desires / goals the agent is committed to achieve



Procedural Reasoning System (PRS) [Georgeff et al., 1987]

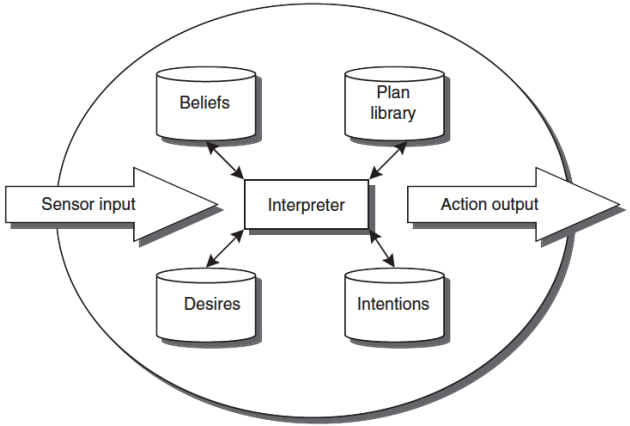
Alessandro Ricci, *Levels of Abstraction in Designing and Programming Systems of Cognitive Agents*, HyperAgents 2019 workshop, co-located with TheWebConf 2019.

```
/* Initial beliefs and rules */

environment_iri("http://yggdrasill.andreiciortea.ro/environments/env1").

positive_color(0.409, 0.518).
negative_color(0.167, 0.04).

/* Initial goals */

!start.


/* Plans for loading the environment */

+!start : environment_iri(EnvIRI) <-
  .print("hello world, today I'll explore the environment: ", EnvIRI);
  .wait(1000);
  .send(node_manager, achieve, environment_loaded(EnvIRI)).

+environment_loaded(EnvIRI, WorkspaceNames) : true <-
  .print("Environment loaded: ", EnvIRI).


/* Plans for discovering and using artifacts */

+artifact_available("emas.EventGeneratorArtifact", ArtifactName, WorkspaceName) : true <-
  .print("An event generator artifact is available in workspace: ", WorkspaceName);
  joinWorkspace(WorkspaceName, WorkspaceArtId);
  focusWhenAvailable(ArtifactName).

+artifact_available("emas.HueArtifact", ArtifactName, WorkspaceName) : true <-
  .print("A Philis Hue light bulb artifact is available in workspace: ", WorkspaceName);
  joinWorkspace(WorkspaceName, WorkspaceArtId);
  focusWhenAvailable(ArtifactName).

+thing_artifact_available(ArtifactIRI, ArtifactName, WorkspaceName) : true <-
  .print("A thing artifact is available: " , ArtifactIRI);
  joinWorkspace(WorkspaceName, WorkspaceArtId);
  focusWhenAvailable(ArtifactName).

/* Plans for handling positive and negative events */

+event("positive")
  : thing_artifact_available(_, ArtifactName, WorkspaceName) &
    hasAction(_,"http://iotschema.org/SwitchOn")[artifact_name(_, ArtifactName)]
    & hasAction(_,"http://iotschema.org/SwitchOff")[artifact_name(_, ArtifactName)]
    & hasAction(_,"http://iotschema.org/SetColor")[artifact_name(_, ArtifactName)]
  <-
  .print("There is a positive event and I can turn on a green light via a thing: ", ArtifactName);
  joinWorkspace(WorkspaceName, WorkspaceArtId);
  ?positive_color(CIEx, CIEy);
  !thing_colored_light_notification(ArtifactName, CIEx, CIEy).
```

(Multi-)Agent Oriented Programming – (M)AOP

Belief-Desire-Intention (BDI) model/architecture (‘80s):

- **beliefs**: information the agent holds about the world (can be out of date or inaccurate);
- **desires**: states of affairs the agent wishes to bring to the world (i.e., the agent’s *goals*)
- **intentions**: desires / goals the agent is committed to achieve



Procedural Reasoning System (PRS) [Georgeff et al., 1987]

Developers can then program agents to **deliberate** about their *mental states* (and to modify them as needed)

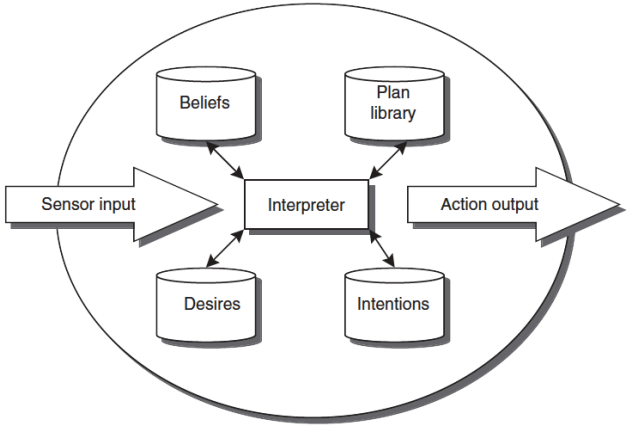
- e.g., suspend an intention to react to an event, drop an intention if it becomes unachievable, etc.

Alessandro Ricci, *Levels of Abstraction in Designing and Programming Systems of Cognitive Agents*, HyperAgents 2019 workshop, co-located with TheWebConf 2019.

(Multi-)Agent Oriented Programming – (M)AOP

Belief-Desire-Intention (BDI) model/architecture (‘80s):

- **beliefs**: information the agent holds about the world (can be out of date or inaccurate);
- **desires**: states of affairs the agent wishes to bring to the world (i.e., the agent’s *goals*)
- **intentions**: desires / goals the agent is committed to achieve



Procedural Reasoning System (PRS) [Georgeff et al., 1987]

Developers can then program agents to **deliberate** about their *mental states* (and to modify them as needed)

- e.g., suspend an intention to react to an event, drop an intention if it becomes unachievable, etc.

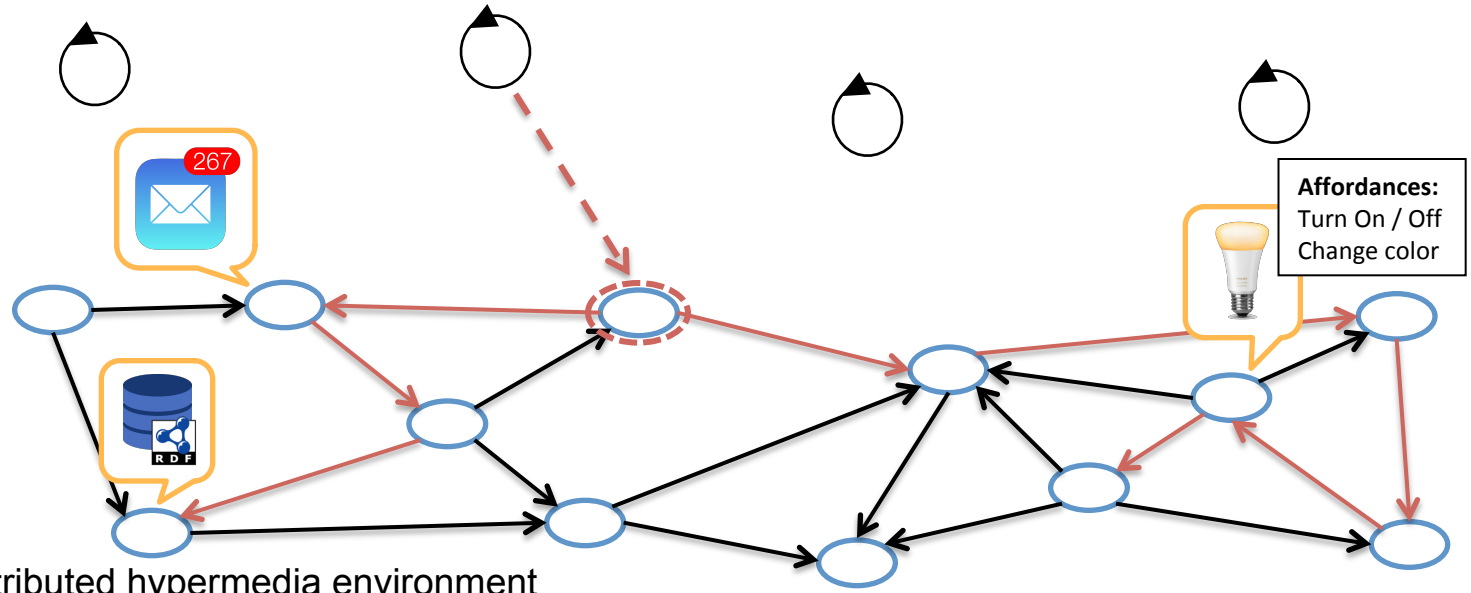
MAOP: integrates design & programming dimensions and abstractions in addition to agents [Boissier et al., 2013]

- **environment** and **organization** dimensions
- separation of concerns (away from *everything-is-an-agent*)

Alessandro Ricci, *Levels of Abstraction in Designing and Programming Systems of Cognitive Agents*, HyperAgents 2019 workshop, co-located with TheWebConf 2019.

Hypermedia Multi-Agent Systems

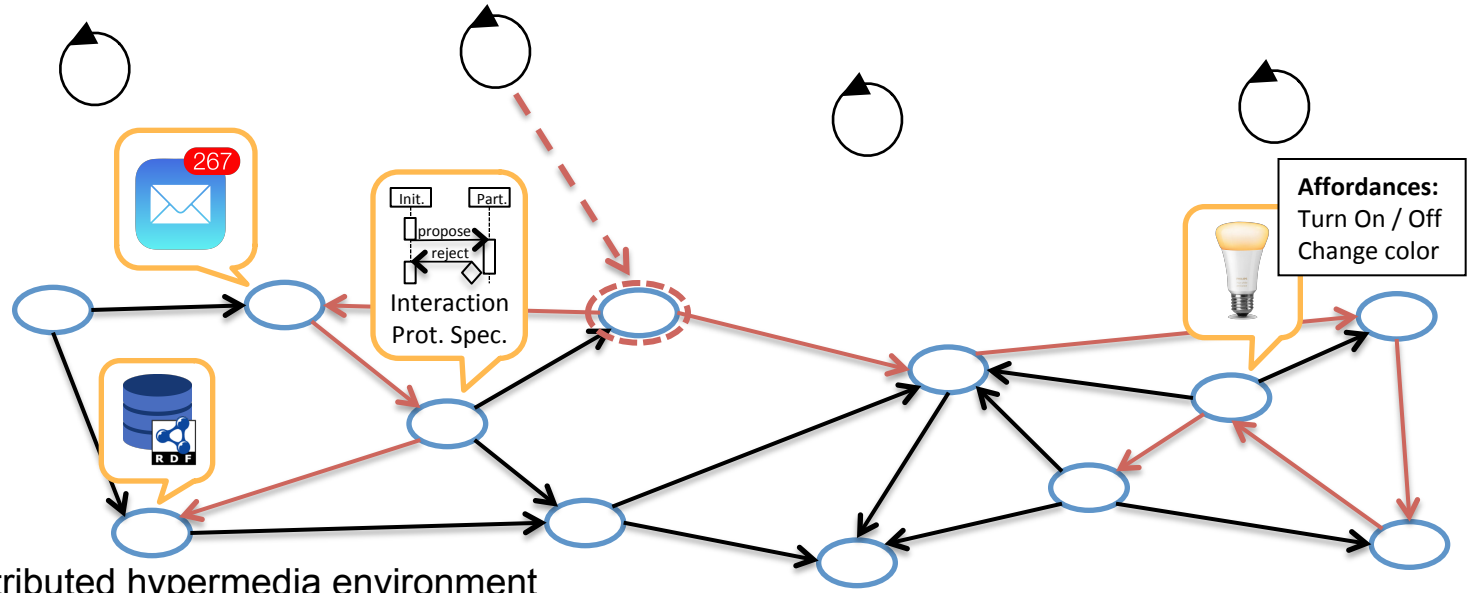
Socio-technical systems composed of people and autonomous agents situated and interacting in a shared hypermedia environment that is distributed across the open Web.



Distributed hypermedia environment

Hypermedia Multi-Agent Systems

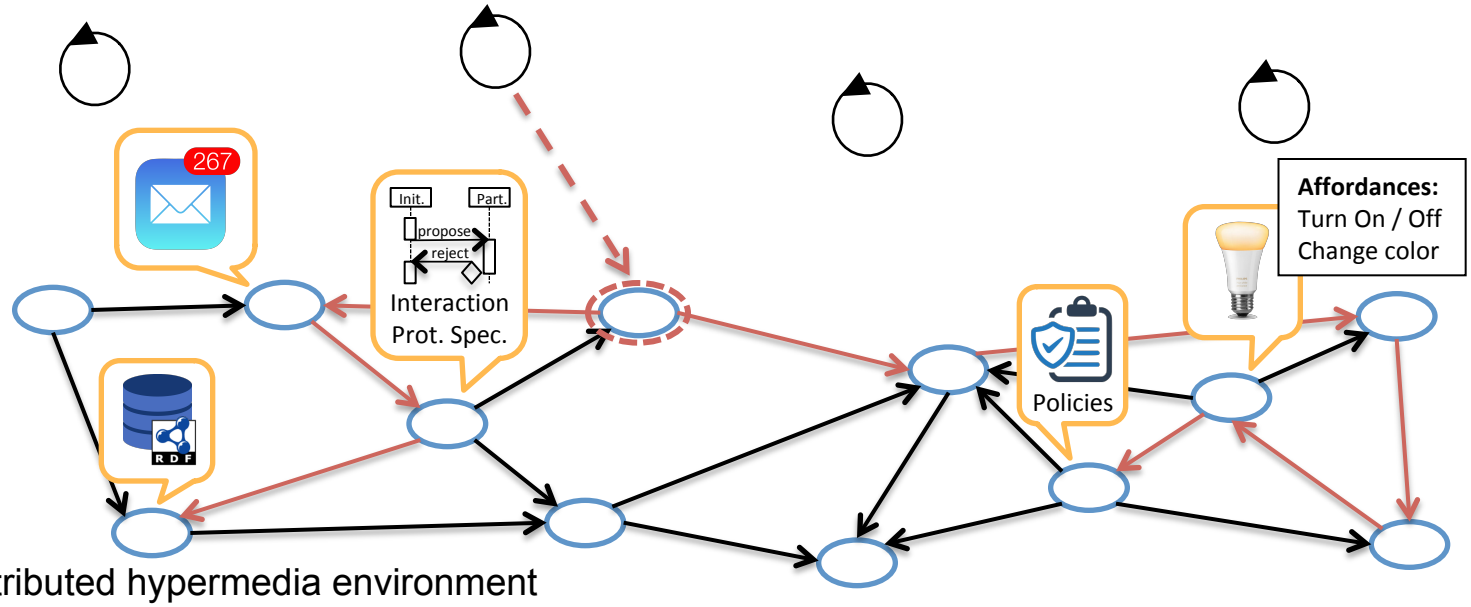
Socio-technical systems composed of people and autonomous agents situated and interacting in a shared hypermedia environment that is distributed across the open Web.



Distributed hypermedia environment

Hypermedia Multi-Agent Systems

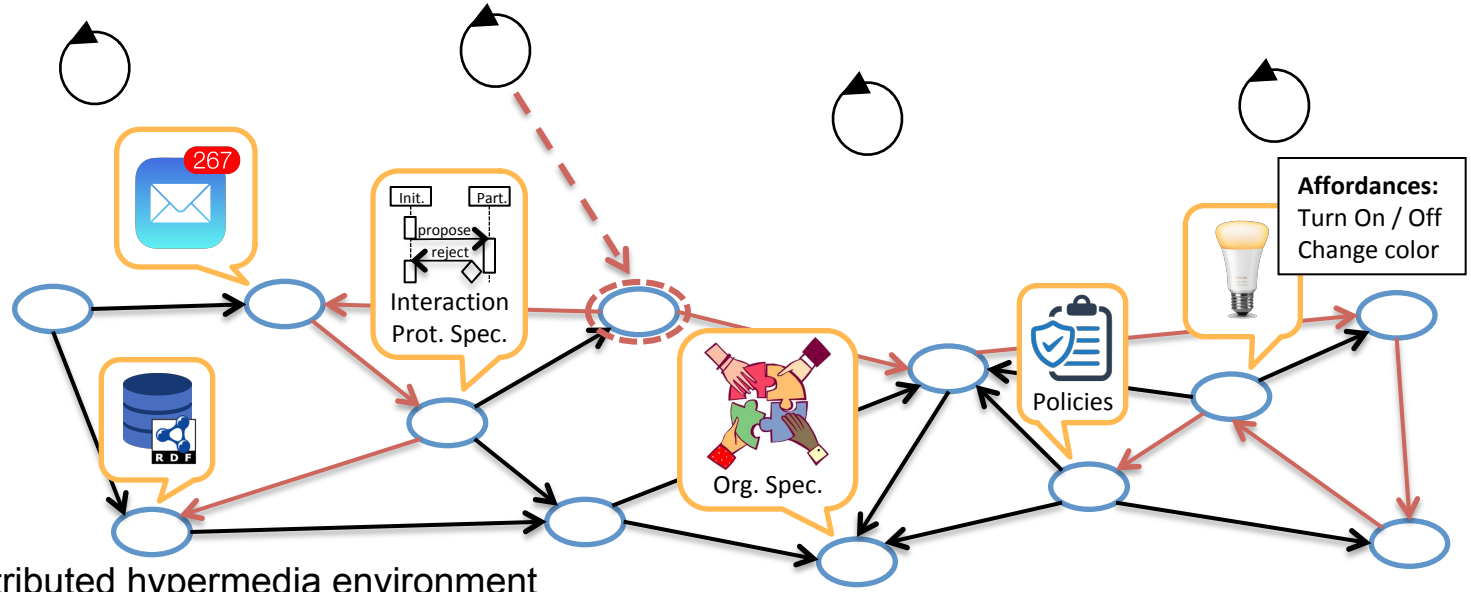
Socio-technical systems composed of people and autonomous agents situated and interacting in a shared hypermedia environment that is distributed across the open Web.



Distributed hypermedia environment

Hypermedia Multi-Agent Systems

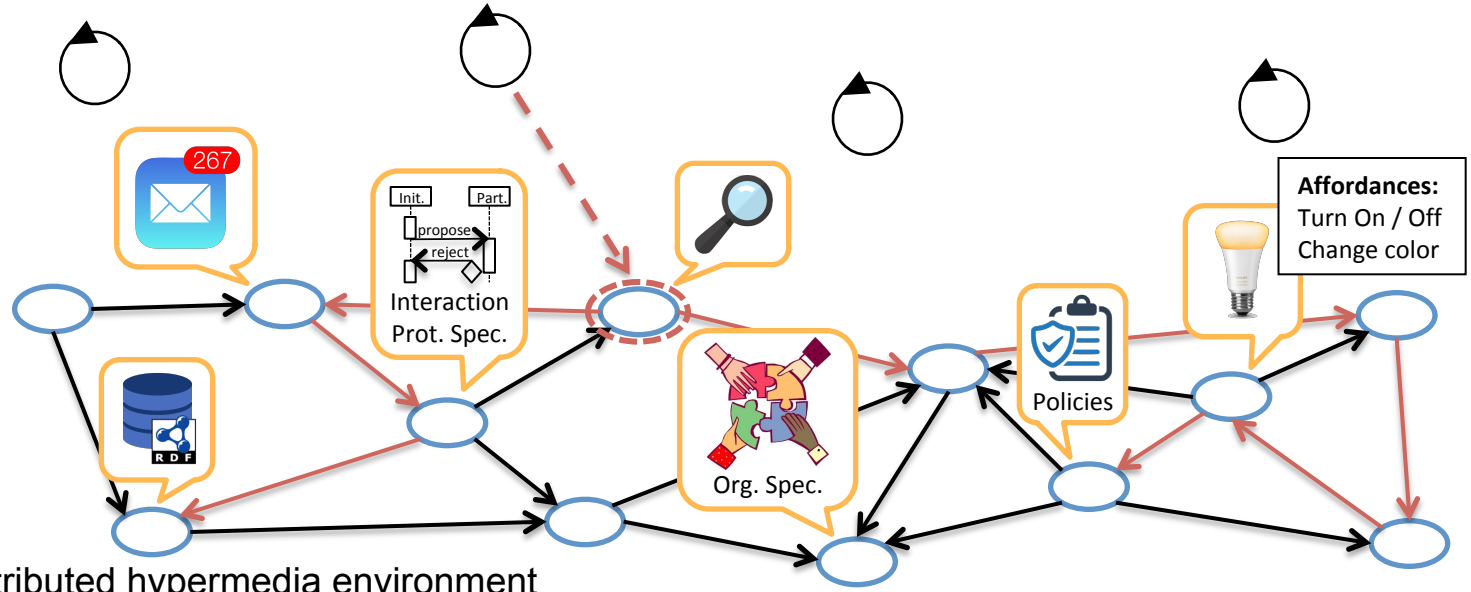
Socio-technical systems composed of people and autonomous agents situated and interacting in a shared hypermedia environment that is distributed across the open Web.



Distributed hypermedia environment

Hypermedia Multi-Agent Systems

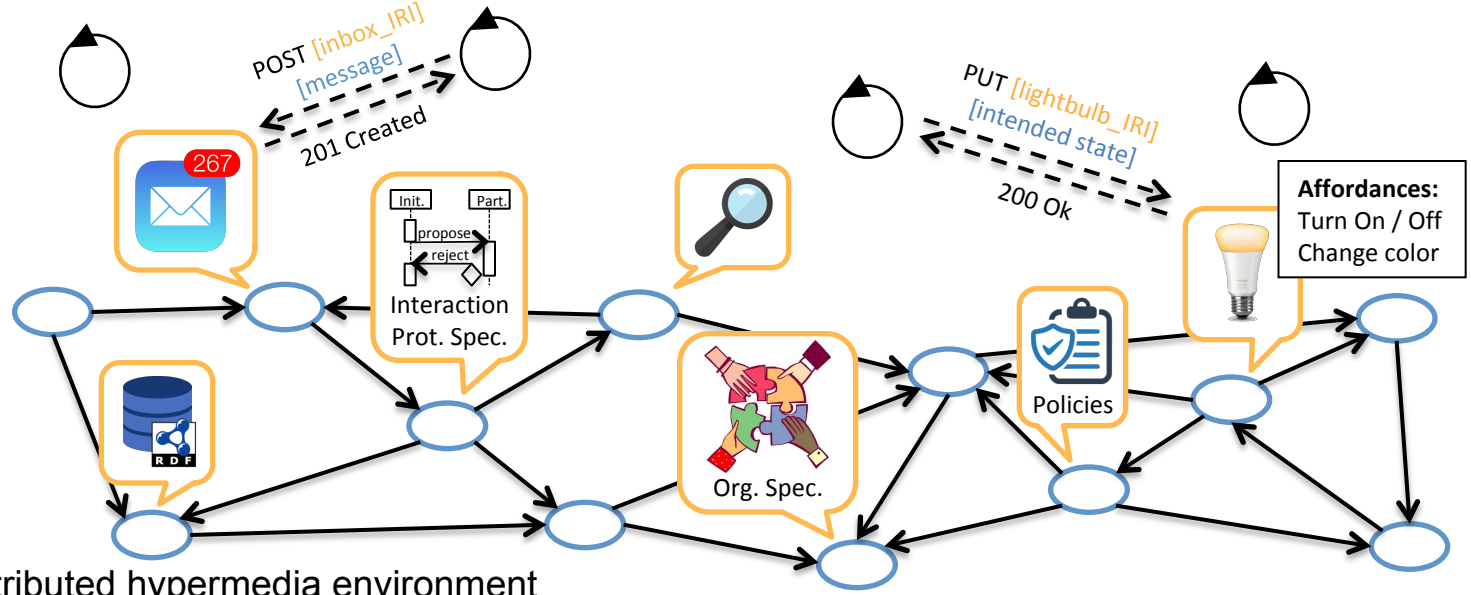
Socio-technical systems composed of people and autonomous agents situated and interacting in a shared hypermedia environment that is distributed across the open Web.



Distributed hypermedia environment

Hypermedia Multi-Agent Systems

Socio-technical systems composed of people and autonomous agents situated and interacting in a shared hypermedia environment that is distributed across the open Web.

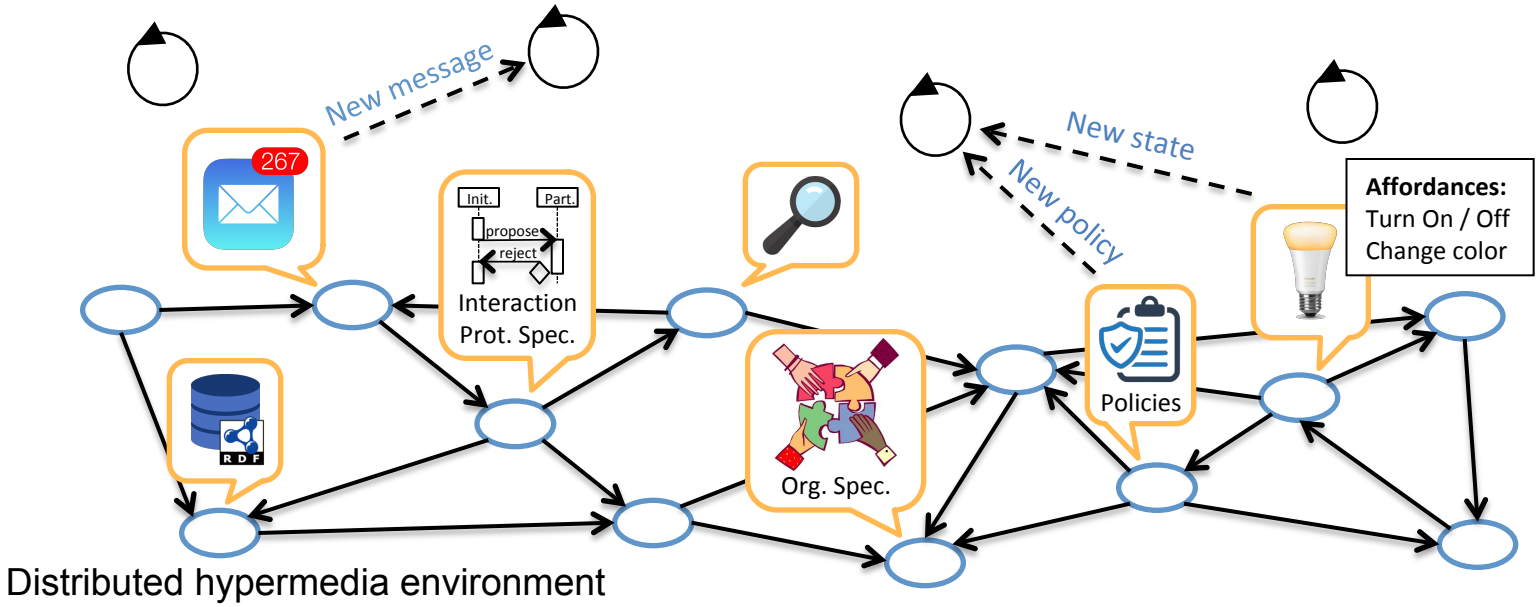


Distributed hypermedia environment

Hypermedia Multi-Agent Systems

Socio-technical systems composed of people and autonomous agents situated and interacting in a shared hypermedia environment that is distributed across the open Web.

- **Key point**: design & program hypermedia environments for autonomous agents



Distributed hypermedia environment

Agent-based Manufacturing for the WoT [Ciortea et al., 2018]

Manufacturing lines are **expensive** to design and build

… and **hard to reuse or reconfigure** for new (generations of) products!

Agent-based Manufacturing for the WoT [Ciortea et al., 2018]

Manufacturing lines are **expensive** to design and build

… and **hard to reuse or reconfigure** for new (generations of) products!

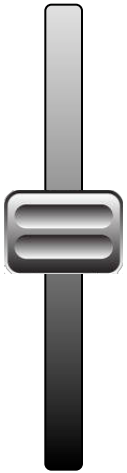
How to design manufacturing systems that can be **repurposed on the fly**?

Agent-based Manufacturing for the WoT [Ciortea et al., 2018]

Manufacturing lines are **expensive** to design and build

… and **hard to reuse or reconfigure** for new (generations of) products!

How to design manufacturing systems that can be **repurposed on the fly**?



Autonomy Lever

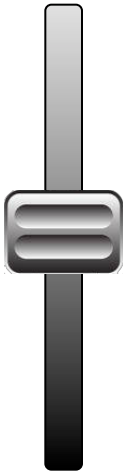
0% Production engineers configure and program the system.

Agent-based Manufacturing for the WoT [Ciortea et al., 2018]

Manufacturing lines are **expensive** to design and build

… and **hard to reuse or reconfigure** for new (generations of) products!

How to design manufacturing systems that can be **repurposed on the fly**?



Autonomy Lever

100% Production engineers provide a specification of the desired product, the system infers a production plan (if possible).

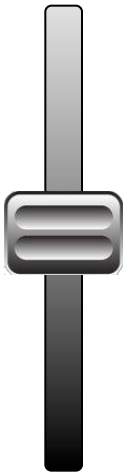
0% Production engineers configure and program the system.

Agent-based Manufacturing for the WoT [Ciortea et al., 2018]

Manufacturing lines are **expensive** to design and build

… and **hard to reuse or reconfigure** for new (generations of) products!

How to design manufacturing systems that can be **repurposed on the fly**?



Autonomy Lever

100% Production engineers provide a specification of the desired product, the system infers a production plan (if possible).

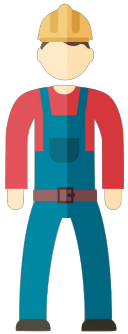
Production engineers provide parts of the solution, the system fills in the blanks.

0% Production engineers configure and program the system.

Deployment Scenario: Furniture Assembly

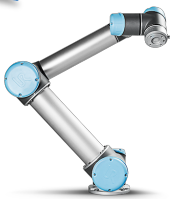
I can **assemble** parts!

Manufacturing goal:



I can **deliver medium-sized** parts!

I can **deliver small-sized** parts!

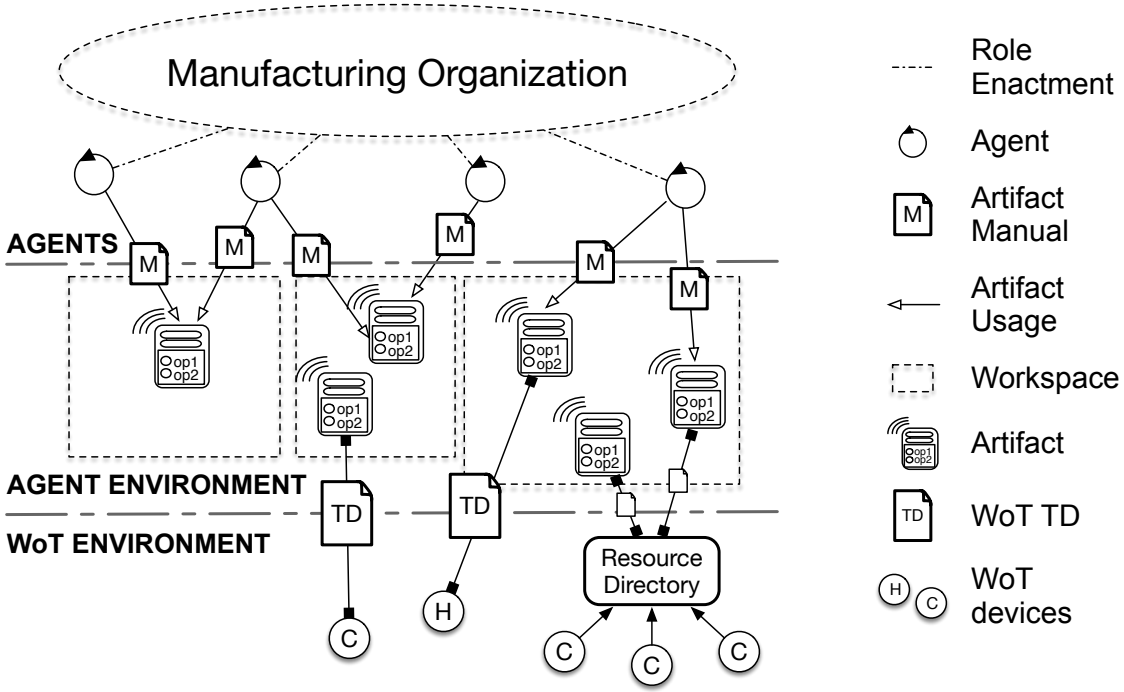


Universal Robotics UR5
(via ROS [Quigley et al., 09])

Fischertechnik ROBO TX
(via Siemens S7-300 PLC)

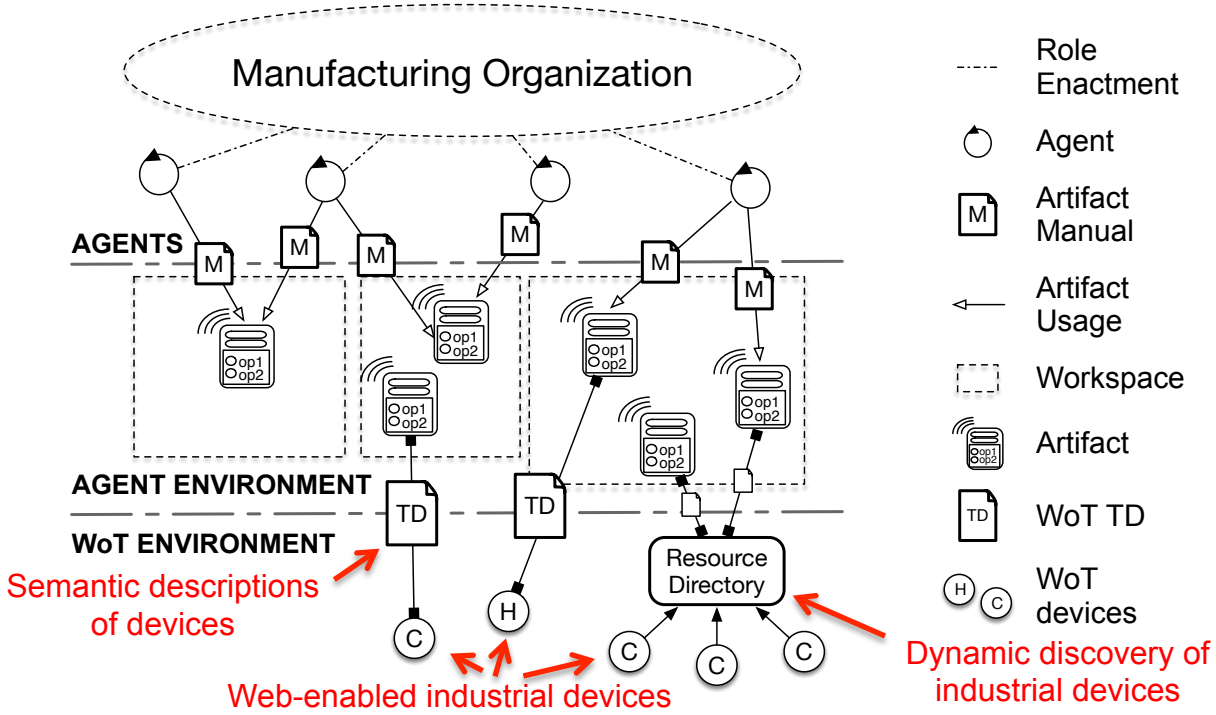
Agent-based Manufacturing for the WoT [Ciortea et al., 2018]

Programming dimensions: **Agent** & **Environment** & **Organization**
(see JaCaMo meta-model [Boissier et al., 2013])



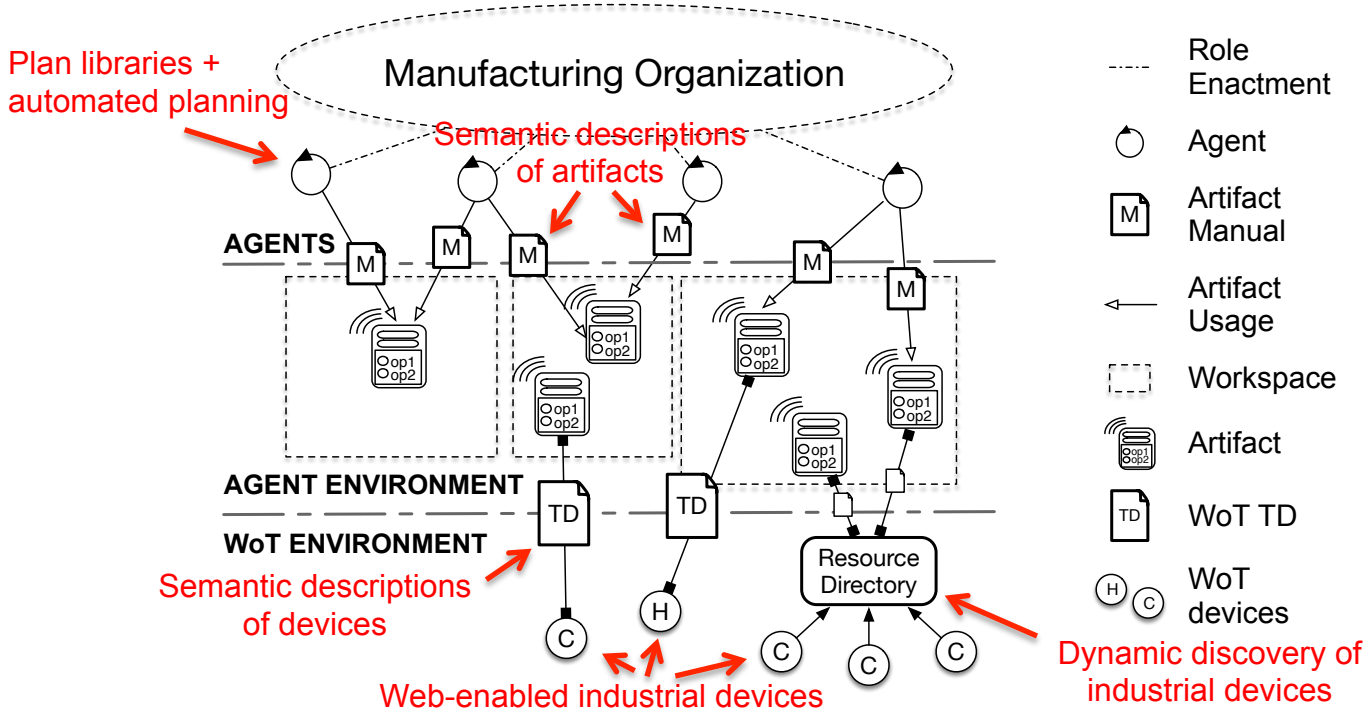
Agent-based Manufacturing for the WoT [Ciortea et al., 2018]

Programming dimensions: **Agent** & **Environment** & **Organization**
(see JaCaMo meta-model [Boissier et al., 2013])



Agent-based Manufacturing for the WoT [Ciortea et al., 2018]

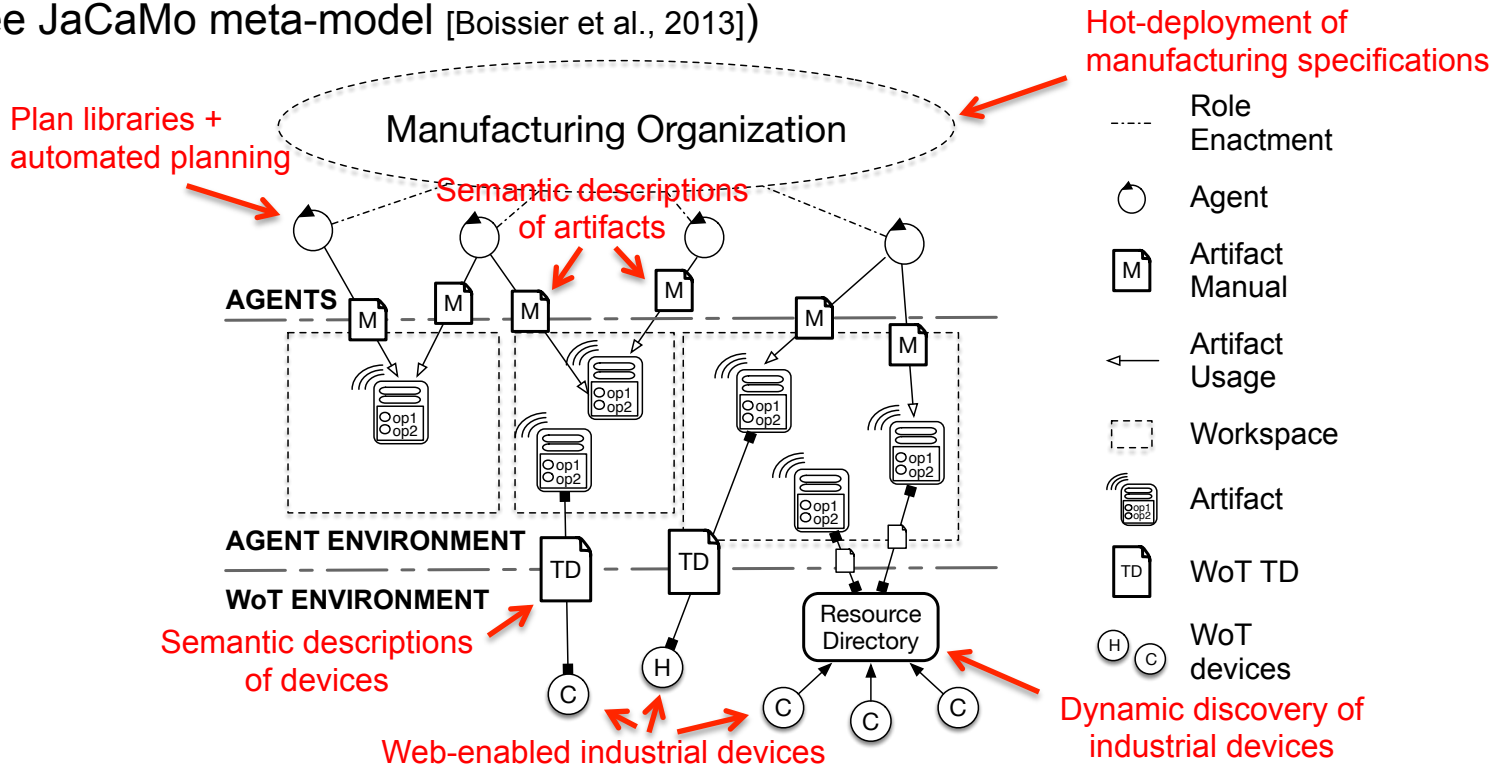
Programming dimensions: **Agent** & **Environment** & **Organization**
(see JaCaMo meta-model [Boissier et al., 2013])



Agent-based Manufacturing for the WoT [Ciortea et al., 2018]

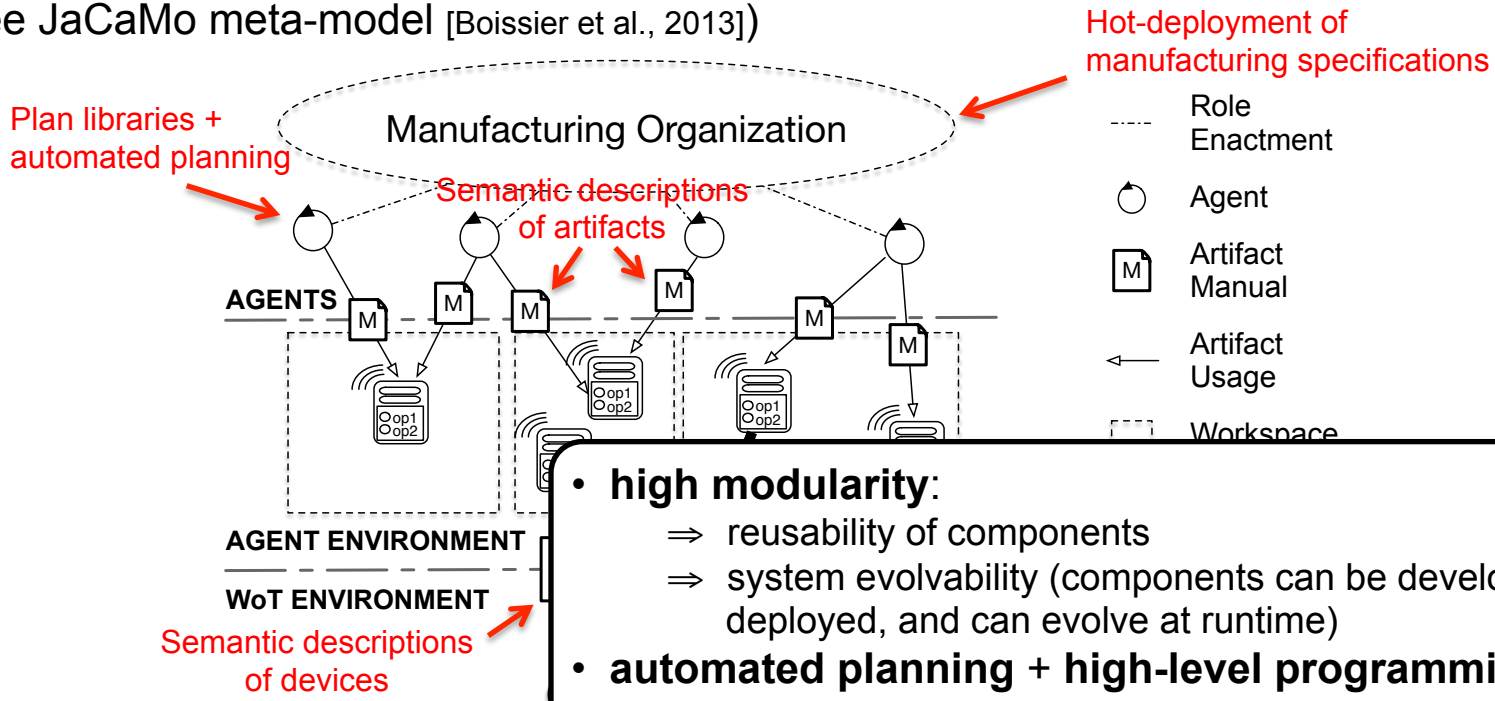
Programming dimensions: **Agent** & **Environment** & **Organization**

(see JaCaMo meta-model [Boissier et al., 2013])



Agent-based Manufacturing for the WoT [Ciortea et al., 2018]

Programming dimensions: **Agent** & **Environment** & **Organization**
(see JaCaMo meta-model [Boissier et al., 2013])

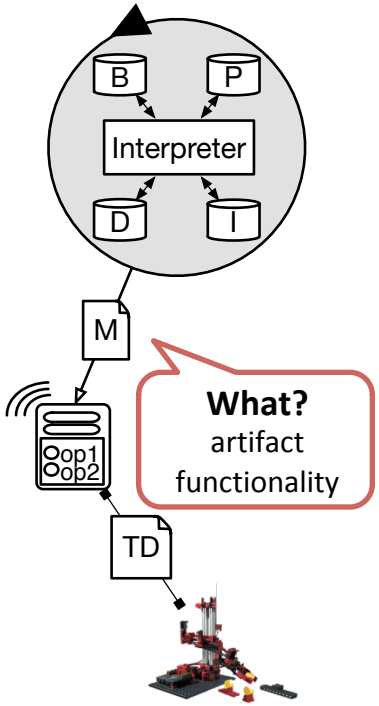


- **high modularity**:
  - ⇒ reusability of components
  - ⇒ system evolvability (components can be developed, deployed, and can evolve at runtime)
- **automated planning + high-level programming**:
  - ⇒ facilitates repurposing

BDI Agents & Planning for Web-based Artifacts

E.g., using *N3 rules*[1] to describe artifact operations:

```
{
  Precond: gripper is empty ∧
  destination (x,y,z) reachable
}
=>
{
  [ a cartago:Operation ;
     cartago:hasName "move" ;
     cartago:hasInputParameters [
        a rdf:Seq ;
        rdf:_1 "?x"^^xsd:decimal ;
        rdf:_2 "?y"^^xsd:decimal ;
        rdf:_3 "?z"^^xsd:decimal ;
     ]
  ] .
   Postcond:
  gripper at destination
} .
```

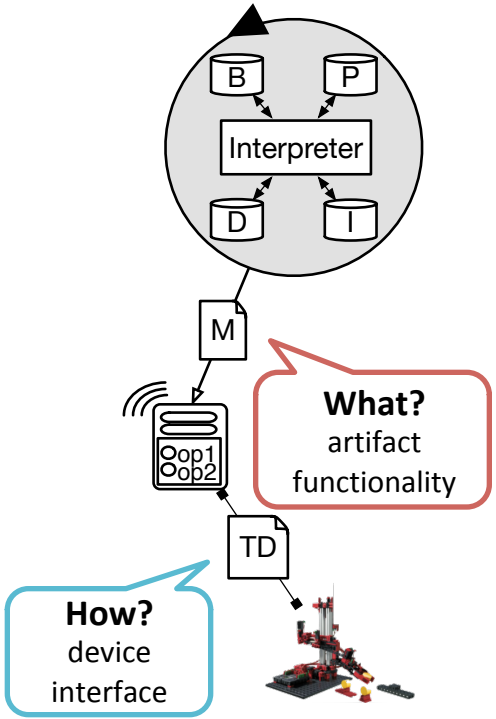


[1]https://www.w3.org/TeamSubmission/n3/

BDI Agents & Planning for Web-based Artifacts

E.g., using *N3 rules*[1] to describe artifact operations:

```
{
  Precond: gripper is empty ∧
  destination (x,y,z) reachable
}
=>
{
  [ a cartago:Operation ;
    cartago:hasName "move" ;
    cartago:hasInputParameters [
      a rdf:Seq ;
      rdf:_1 "?x"^^xsd:decimal ;
      rdf:_2 "?y"^^xsd:decimal ;
      rdf:_3 "?z"^^xsd:decimal ;
    ]
  ] .
  Postcond:
  gripper at destination
} .
```



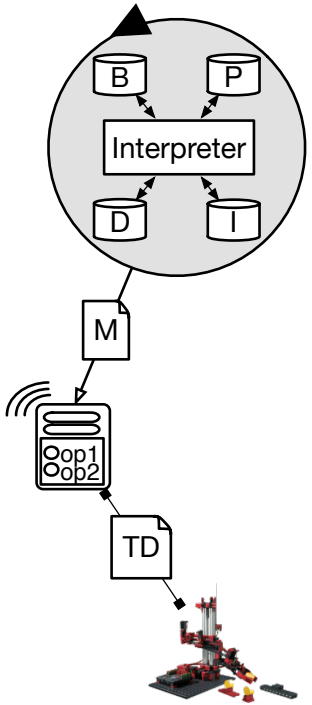
E.g., using *WoT TDs*[2]:

- Uniform interface:
  - observable properties
  - observable events
  - actions

Artifact model ⟷ WoT TD model:
⇒ conceptual bridge between MAS and WoT systems
[Ciortea et al., WoT 2017]

[1]https://www.w3.org/TeamSubmission/n3/
[2]https://www.w3.org/TR/wot-thing-description/

BDI Agents & Planning for Web-based Artifacts

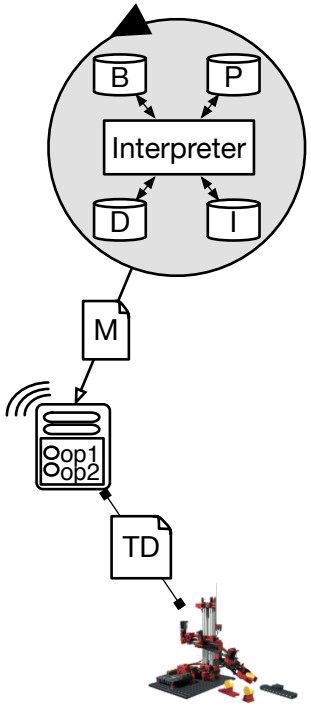


1 Achieve goal by selecting and executing a plan from the plan library.

BDI Agents & Planning for Web-based Artifacts

1 Achieve goal by selecting and executing a plan from the plan library.

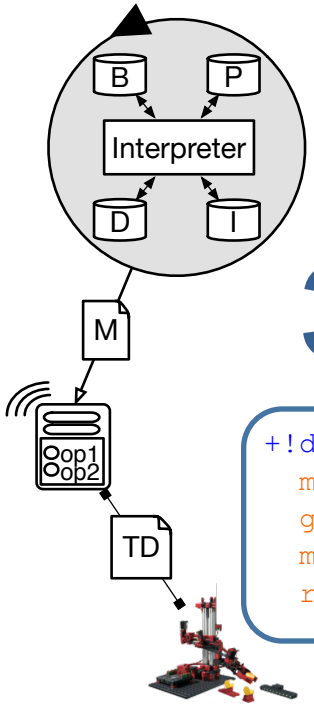
2 If there is no applicable plan, synthesize a plan using the manuals of known artifacts.



BDI Agents & Planning for Web-based Artifacts

1 Achieve goal by selecting and executing a plan from the plan library.

2 If there is no applicable plan, synthesize a plan using the manuals of known artifacts.



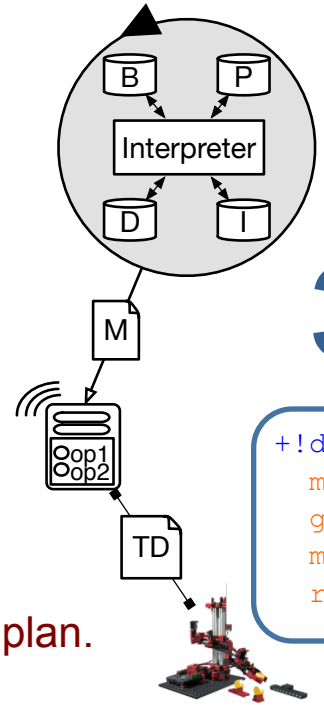
3 If a solution is found, translate the synthesized plan to AgentSpeak and add it to the plan library.

```
+!deliver_pad1 : true <-
  move(-3, 20, 14)[artifact_name(”plc_arm")];
  grab[artifact_name(”plc_arm")];
  move(20, -3, 14)[artifact_name(”plc_arm")];
  release[artifact_name(”plc_arm")];
```

BDI Agents & Planning for Web-based Artifacts

1 Achieve goal by selecting and executing a plan from the plan library.

2 If there is no applicable plan, synthesize a plan using the manuals of known artifacts.



3 If a solution is found, translate the synthesized plan to AgentSpeak and add it to the plan library.

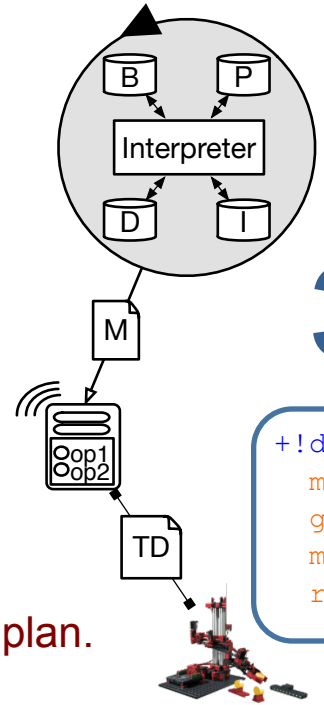
```
+!deliver_pad1 : true <-
  move(-3, 20, 14)[artifact_name(”plc_arm")];
  grab[artifact_name(”plc_arm")];
  move(20, -3, 14)[artifact_name(”plc_arm")];
  release[artifact_name(”plc_arm")];
```

4 Retry to achieve goal by executing the synthesized plan.

BDI Agents & Planning for Web-based Artifacts

1 Achieve goal by selecting and executing a plan from the plan library.

2 If there is no applicable plan, synthesize a plan using the manuals of known artifacts.



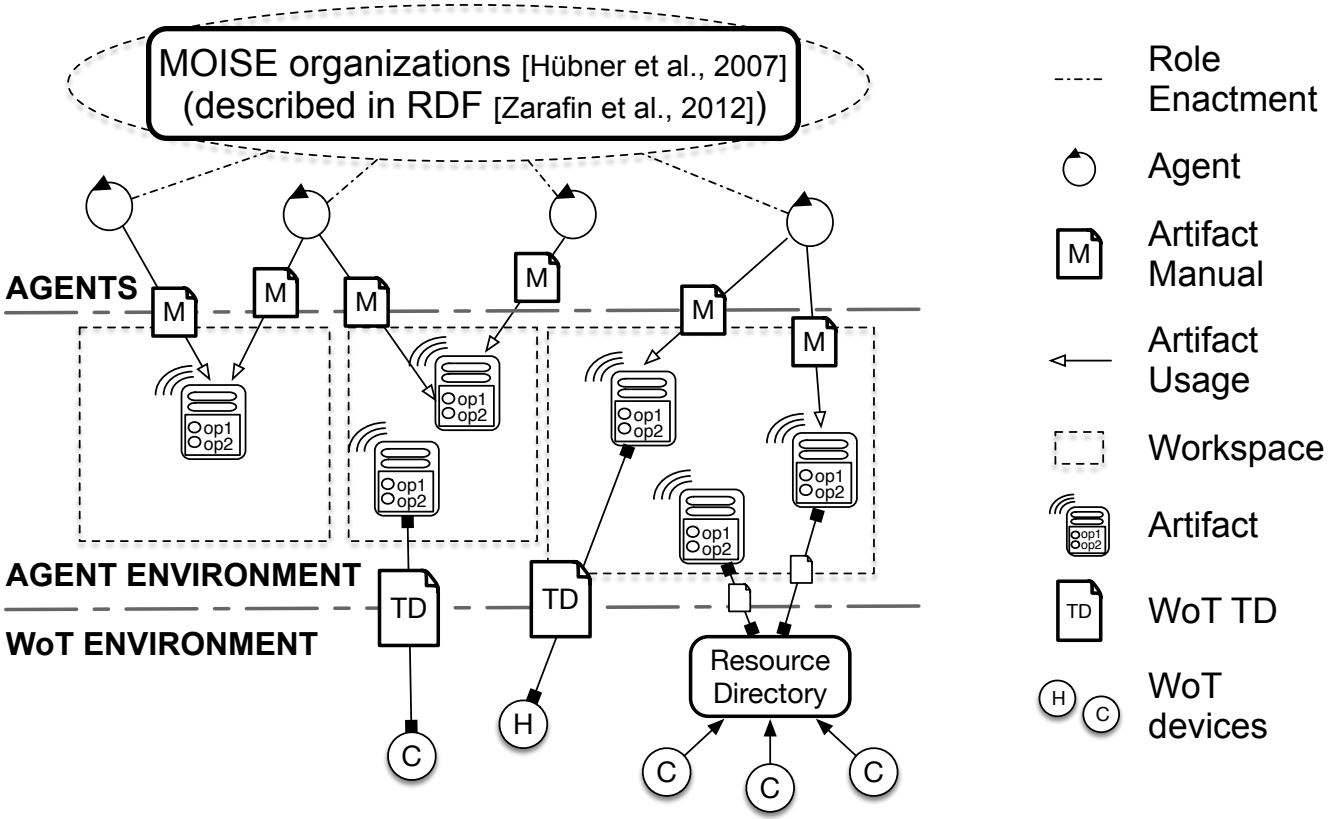
3 If a solution is found, translate the synthesized plan to AgentSpeak and add it to the plan library.

```
+!deliver_pad1 : true <-
  move(-3, 20, 14)[artifact_name("plc_arm")];
  grab[artifact_name("plc_arm")];
  move(20, -3, 14)[artifact_name("plc_arm")];
  release[artifact_name("plc_arm")];
```

Engineers can inspect and edit inferred plans in AgentSpeak

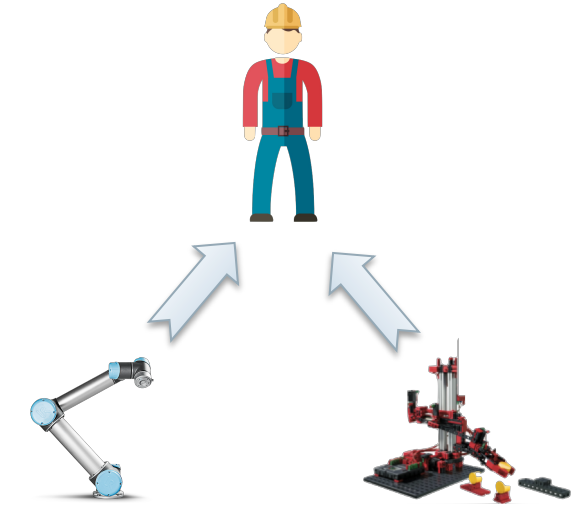
4 Retry to achieve goal by executing the synthesized plan.

Manufacturing Organizations



Manufacturing Organizations: Furniture Assembly

Mission:
Mount pads on legs
Attach legs to stool



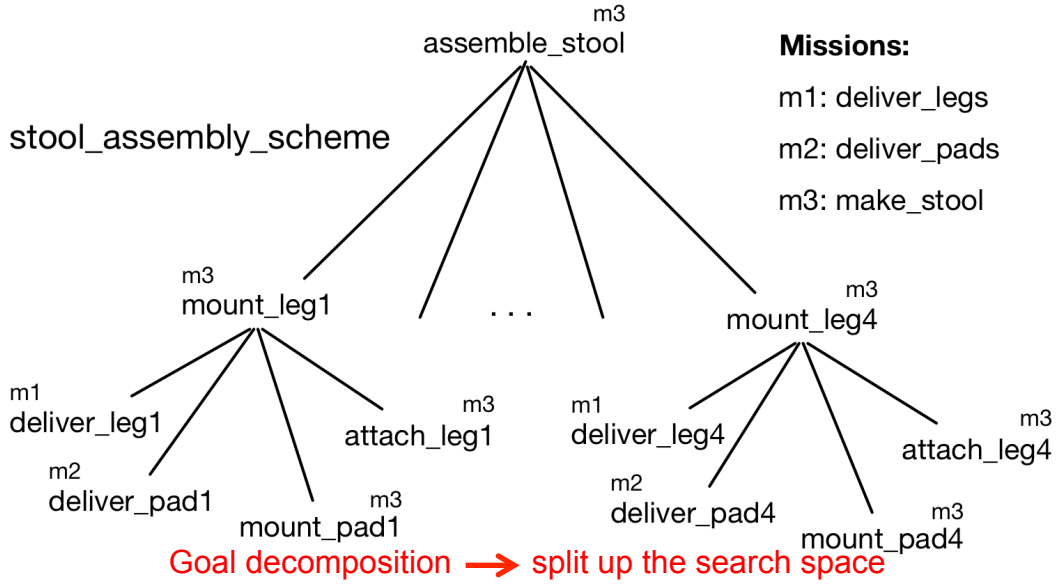
Mission:
Deliver legs

Mission:
Deliver pads

MOISE [Hübner et al, 07] **organizational specification:**

- *functional dimension*: one manufacturing scheme
- *structural dimension*: one group (with 3 roles)

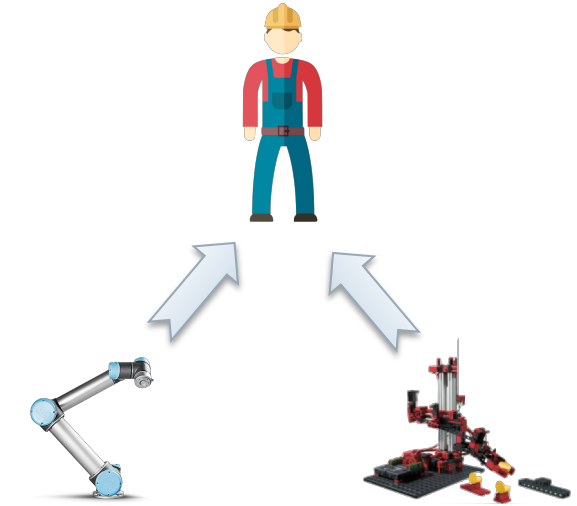
  `assembly_worker, leg_transporter, pad_transporter`
- *normative dimension*: norms assigning missions to roles



Goal decomposition → split up the search space

Manufacturing Organizations: Furniture Assembly

Mission:
Mount pads on legs
Attach legs to stool



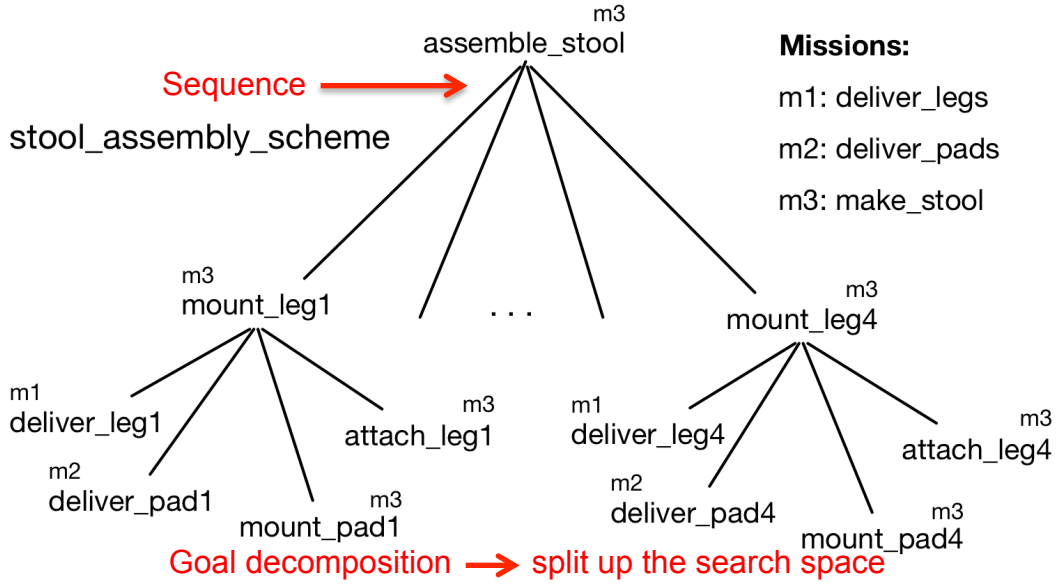
Mission:
Deliver legs

Mission:
Deliver pads

MOISE [Hübner et al, 07] **organizational specification:**

- *functional dimension*: one manufacturing scheme
- *structural dimension*: one group (with 3 roles)

  `assembly_worker, leg_transporter, pad_transporter`
- *normative dimension*: norms assigning missions to roles



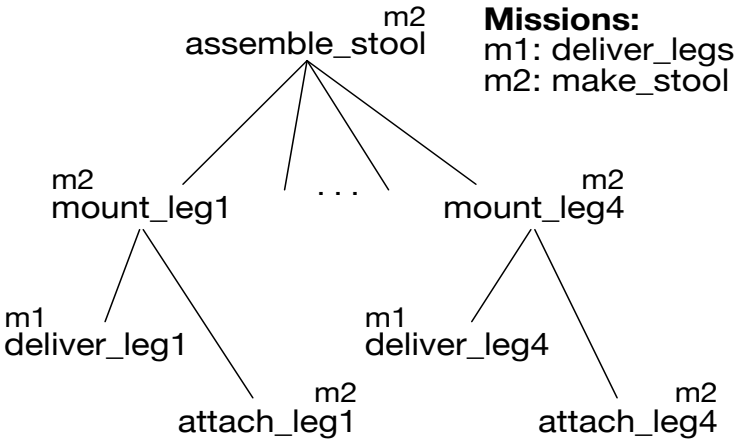
Manufacturing Organizations: Probing (“git diff”)

Stool with 4 legs

Stool with 4 padded legs

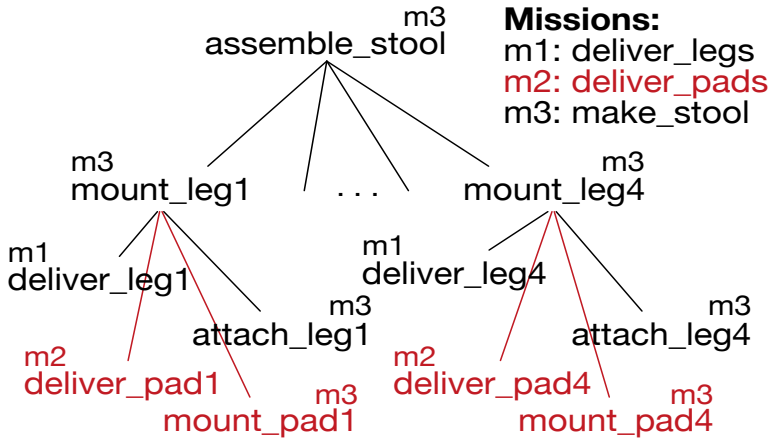
Manufacturing group:
`assembly_worker, leg_transporter`

Manufacturing scheme:



Manufacturing group:
`assembly_worker, leg_transporter, pad_transporter`

Manufacturing scheme:



Demo video:
https://youtu.be/tfAVDpYn_ow

Outline

- Introduction
- Hypermedia MAS
- Agent-based Manufacturing for the WoT
- Autonomy in the WoT: Challenges and Opportunities
- Conclusions

Autonomy in the WoT: Challenges & Opportunities

Autonomy in the WoT: Challenges & Opportunities

First Workshop on Hypermedia Multi-Agent Systems

San Francisco, May 13

http://hyperagents.org @hyperagents *Weaving a Web for People and Artificial Agents*

Autonomy in the WoT: Challenges & Opportunities

- Interaction as a first-class abstraction
  - W3C WoT TD, Hydra, etc. pave the way for declarative specifications of interactions on the Web

Autonomy in the WoT: Challenges & Opportunities

- Interaction as a first-class abstraction
  - W3C WoT TD, Hydra, etc. pave the way for declarative specifications of interactions on the Web

"Hypertext does not need to be HTML on a browser. Machines can follow links when they understand the data format and relationship types."

"(...), then automated agents can traverse these applications almost as well as any human. There are plenty of examples in the linked data communities. (...) and thus we can design the protocols to support both machine and human-driven applications by following the same architectural style."

Roy Fielding, 2008

https://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven

Autonomy in the WoT: Challenges & Opportunities

- Interaction as a first-class abstraction
  - W3C WoT TD, Hydra, etc. pave the way for declarative specifications of interactions on the Web
  - Declarative specification and enactment of interactions has been studied to large extent in MAS research (e.g., see [Baldoni et al., 2018; Chopra and Singh, 2016])

Autonomy in the WoT: Challenges & Opportunities

- Interaction as a first-class abstraction
  - W3C WoT TD, Hydra, etc. pave the way for declarative specifications of interactions on the Web
  - Declarative specification and enactment of interactions has been studied to large extent in MAS research (e.g., see [Baldoni et al., 2018; Chopra and Singh, 2016])

- Regulation as a first-class abstraction
  - data licensing policies, terms of service, API rate limiting, etc.?

Conclusions

The introduction of **interaction affordances** is an important step in the evolution of the Web: it supports and motivates autonomous agents on the Web (cf. original Semantic Web vision [Berners-Lee et al., 2001]).

All the elements required to design and deploy **Hypermedia Multi-Agent Systems** are already available.

Thank you!

andrei.ciortea@unisg.ch

@andreiciortea

Selected publications:

- A. Ciortea, S. Mayer, F. Gandon, O. Boissier, A. Ricci, A. Zimmermann: **A Decade in Hindsight: The Missing Bridge Between Multi-Agent Systems and the World Wide Web**, AAMAS 2019
- S. Mayer, A. Ciortea, A. Ricci, M. I. Robles, M. Kovatsch: **Hypermedia to Connect them All – Autonomous Hypermedia Agents and Socio-Technical Interactions**, Internet Technology Letters, 2019
- A. Ciortea, S. Mayer, F. Michahelles: **Repurposing Manufacturing Lines On-the-fly with MAS for the WoT**, AAMAS 2018
- A. Ciortea, O. Boissier, A. Ricci: **Engineering World-Wide Multi-Agent Systems with Hypermedia**, EMAS 2018
- A. Ciortea, O. Boissier, A. Ricci: **Beyond Physical Mashups: Autonomous systems for the Web of Things**, WoT 2017

References

Berners-Lee et al., 2001. *The Semantic Web*. Scientific American.

Boissier et al., 2013. *Multi-agent oriented programming with JaCaMo*. Science of Computer Programming vol. 78, issue 6, 747-761.

Bordini et al., 2007. *Programming Multi-Agent Systems in AgentSpeak using Jason*. John Wiley & Sons.

Ciortea et al., 2017. *Beyond Physical Mashups: Autonomous systems for the Web of Things*. Proceedings of the Eighth International Workshop on the Web of Things (WoT). ACM, 2017.

Ciortea et al., 2018. *Repurposing Manufacturing Lines On-the-fly with MAS for the WoT*. AAMAS 2018.

Georgeff et al., 1987. *Reactive reasoning and planning*. AAAI. Vol. 87.

Hübner et al., 2007. *Developing Organised Multiagent Systems using the MOISE+ Model: Programming Issues at the System and Agent Levels*. International Journal of Agent-Oriented Software Engineering 1, 3-4 (2007), 370–395.

Zarafin et., 2012. *Integrating Semantic Web Technologies and Multi-Agent Systems: A Semantic Description of Multi-Agent Organizations*. In Proceedings of the First International Conference on Agreement Technologies (CEUR WS), Vol. 918. 296–297. http://ceur-ws.org/Vol-918/111110296.pdf