The Environment Dimension

AI4Industry 2023, Saint-Étienne
JaCaMo Metamodel – Multi-Agent Concepts
This Session’s Agenda

- Environment as a First-Class Abstraction
- The Agents & Artifacts Metamodel

https://www.w3.org/community/webagents/
The Environment Dimension

Single-agent system perspective
[Russell & Norvig, 2020]

The Environment as the world external to the system

Multi-agent system perspective

The Environment becomes part of the system
(e.g.: communication and interaction infra.)

The Environment as a Design Abstraction

The environment is a first-class abstraction that provides the surrounding conditions for agents to exist and that mediates both the interaction among agents and the access to resources [Weyns et al., 2007].

**Reflection support** [Rici et al., 2011]: mechanisms to modify the functional behavior of the environment

- Example: creating and destroying artifacts

**Interaction-mediation support**: mechanisms to mediate, enact, and regulate interactions

- Example: pheromone infrastructure, e-institutions, rate limiting, etc.

**Abstraction support**: conceptual bridge between abstractions used to design and program agents and the deployment context

- Example: semantic models, domain-specific abstractions, etc.

**Basic interface support**: raw access to the deployment context

- Example: Web APIs, device interfaces, etc.

---

Engineering MAS: environment as a first-class design abstraction [Weyns et al., 2007].


Example: Flexible Industrial Manufacturing

**SIEMENS**

**Modularity and Encapsulation**

Uniformly **encapsulate** and **modularize** functionalities of the MAS outside of the agents

- **reusability** of components
- **system evolvability**: components can be developed, deployed, and can evolve independently

This Session’s Agenda

• Environment as a First-Class Abstraction
• The Agents & Artifacts Metamodel
Activity Theory

Activity (basic unit of analysis) is a **goal-directed interaction** with the world.

The activity is mediated through **tools** (or **artifacts**), which evolve over time based on the experience of subjects.

Roots in cultural-historical psychology (1920s and 1930s)

The Agents & Artifacts Metamodel

Key idea: separation of concerns
- agents encapsulate autonomous behavior
- artifacts encapsulate non-autonomous behavior

Programming MAS = Programming Agents
  + Programming the Environment

The agents’ environment is modelled as a dynamic set of artifacts grouped into workspaces
- the actions provided to agents are determined by the artifacts discovered at run time
- agents construct, share, and use artifacts to support their working activities
  ⇒ artifacts are mediating tools for goal-directed agents
  ⇒ agents can modify the functional behavior of the environment to meet their needs

The environment is a first-class design and programming abstraction

The Workspace Abstraction

A logical place containing artifacts and the working context of the agents’ activities

- provides a notion of locality and situatedness
- allow to structure complex/distributed environments

Agents can join, leave, and work in multiple workspaces at the same time

- agents are embodied and interact within the workspace through body artifacts
  ⇒ separation of concerns between the agent’s mind and the agent’s body
  ⇒ allows heterogeneous agents (implementing different architectures) to join and work in the same environment

Workspaces can be distributed over a network

The Artifact Abstraction

Artifacts as computational objects

- **usage interface:**
  - **observable properties**: state variables that can be perceived by agents
  - **observable events**: non-persistent signals that carry information and can be perceived by agents
  - **operations**: environmental actions provided to the agent
    - operations can update the values of observable properties or can generate signals
The Artifact Abstraction

Artifacts as computational objects
  – *usage interface*:
    • **observable properties**: state variables that can be perceived by agents
    • **observable events**: non-persistent signals that carry information and can be perceived by agents
    • **operations**: environmental actions provided to the agent
      – operations can update the values of observable properties or can generate signals

Agents can **focus** on artifacts to **perceive** observable properties and signals
The Artifact Abstraction

Artifacts as computational objects

– *usage interface*:
  
  • **observable properties**: state variables that can be perceived by agents
  
  • **observable events**: non-persistent signals that carry information and can be perceived by agents
  
  • **operations**: environmental actions provided to the agent
    – operations can update the values of observable properties or can generate signals

Agents can **focus** on artifacts to **perceive** observable properties and signals
Why is intentional focus useful?

Allows agents to select the parts of the environment that are relevant to their goals
- promotes scalability
  - agents can cope with larger environments
  - the environment infrastructure can serve more agents
- promotes autonomy from the environment

Artifacts as computational objects
- usage interface:
  - **observable properties**: state variables that can be perceived by agents
  - **observable events**: non-persistent signals that carry information and can be perceived by agents
  - **operations**: environmental actions provided to the agent
    - operations can update the values of observable properties or can generate signals

Agents can focus on artifacts to perceive observable properties and signals
Artifacts as computational objects

- **usage interface:**
  - **observable properties:** state variables that can be perceived by agents
  - **observable events:** non-persistent signals that carry information and can be perceived by agents
  - **operations:** environmental actions provided to the agent
    - operations can update the values of observable properties or can generate signals

- **link interface:**
  - used to connect artifacts

---

**Observable Properties:**
- temperature/1

**Signals:**
- new_state/1

**Operations:**
- startCooling/0
- startHeating/0
- stopAirConditioner/0

---

**hvac Artifact**

---

The Artifact Abstraction

**Linked Op.:**

```
msg("stopped hvac")
```

**Signals:**

- dweeter Artifact

---

**dweeter Artifact**
A Basic Taxonomy of Artifacts

Resource Artifacts
– some specific kind of resource that can be shared by agents

Coordination Artifacts
– artifacts specifically designed to provide coordination functionalities by enabling and managing in some way the interaction among agents

Boundary Artifacts
– artifacts that allow agents to interact with human users and, more generally, any actor or system that is external with respect to the MAS

Smart Room Scenario Revisited: Voting Machines

Room Workspace

**Beliefs:**
- status("open")
- options([21, 25, 30])
- deadline(4000)
- result(21)

Vote Machine

**vote0**: VotingMachine

- open(3)
- result(21)
- vote/1
- options/1
- deadline/1

Room Controller

**hvac0**: HVAC

- startCooling/0
- startHeating/0
- stopAirConditioner/0

Beliefs:
- status("open")
- options([21, 25, 30])
- deadline(4000)
- result(21)

Vote:
- vote(21)
- vote(30)
- open([21, 25, 30], [pa1, pa2, pa3], 4000)
- result(21)
Artifacts vs. Objects

Both artifacts and objects model nonautonomous entities and provide a usage interface.

But there are important differences:

- **transfer of control**:
  - in object-object interaction, a method call implies a transfer of control between the caller object and the callee object
  - in agent-artifact interaction, control is encapsulated inside agents and cannot be transferred
    • the execution of a triggered operation is carried out by another logical flow provided by the environment
    • on the agent side, the plan in execution is suspended until the action is either completed or failed (the agent can continue to pursue other intentions)

- **observable state**:
  - artifacts have observable state captured by observable properties
  - unlike public object instance fields, observable properties cannot be written directly (they can be updated by operations)

- **concurrency**: artifacts are thread-safe by design, which makes it easy to share them among agents.

The Agents & Artifacts Metamodel

The environment is a first-class design and programming abstraction

Programming MAS = Programming Agents + Programming the Environment

JaCaMo Metamodel – Multi-Agent Concepts
Any Questions / Comments / Doubts / Concerns?
Images

https://freepik.com