tATAmI-2 – a Flexible Framework for Modular Agents

Andrei Olaru – cs@andreiolaru.ro

University Politehnica of Bucharest

28.05.2015

tATAmI-2 – a Flexible Framework for Modular Agents

overview

. Andrei Olaru – cs@andreiolaru.ro
. CSCS'20
. Bucharest, Romania 28.05.2015

Problem Context

$\rightarrow$ AmI

· Context: building a MAS framework for MAS-based AmI applications

We require a framework with a lot of flexibility:

- agents must be able to run on various devices (PC, Android, iOS, Arduino)
- agents must be able to use various communication methods
  - TCP/IP · web services · web sockets · queues · other?
- agents structure must be able to be very light or more complex
  - behaviors · S-CLAIM · other AOP languages

Problem Context

- How to model agents regardless of their internal structure?
- How to model communication and mobility services?
- How to control agents and platforms?
- How to correctly load platforms and agents?

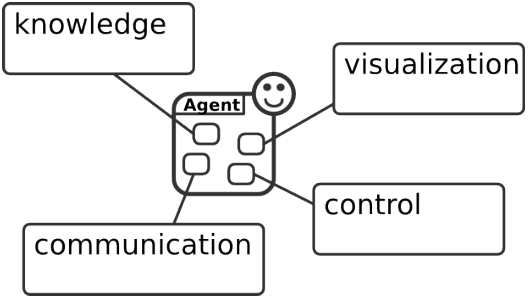
What are we doing?

CLAIM & SymPa	2010	2013	2015
	tATAmI-1	tATAmI-2	**tATAmI-2.5**
	S-CLAIM	S-CLAIM	platforms
A.Suna	Jade	ContextKB	
		Jade	
	A.Olaru	A.Olaru	A.Olaru
	M.T. Benea	M.T. Benea	C.Mihai
	T.T.N.Nguyen	E.Sevastian	

What are we doing?

S-CLAIM
Knowledge
Visualization
Base
Jade agent

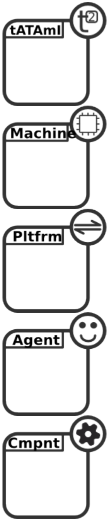
$\rightarrow$



tATAmI-1
class inheritance layers

tATAmI-2
flexible modules/components

tATAmI-2 Architecture



The tATAmI-2 framework connects all platforms and agents, accross multiple machines.

A machine that is part of the framework; it hosts one or more containers, which host agents.

A platform spans multiple machines and offers communication, discovery and mobility services to agents.

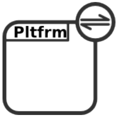
An agent runs inside a container, being loaded on a platform.

A component runs inside an Composite agent an implements functionality.

---

tATAmI-2 Architecture

The platform is an entity that offers various types of services to agents.



· tATAmI-2 sees it as:

PlatformLoader

.start()
.stop()
.loadAgent()
.recommendComponent()

· an agent sees it as *platform link*
- can only be used by specialized components

Loading a platform:
create instance → start → create containers
→ create *link* agents → load agents

AI-MAS Group
Andrei Olaru – cs@andreiolaru.ro
CSCS'20
Bucharest, Romania 28.05.2015

tATAmI-2 Architecture

The agent is a persistent, autonomous entity that percieves, acts and communicates

· tATAmI-2 sees it as:

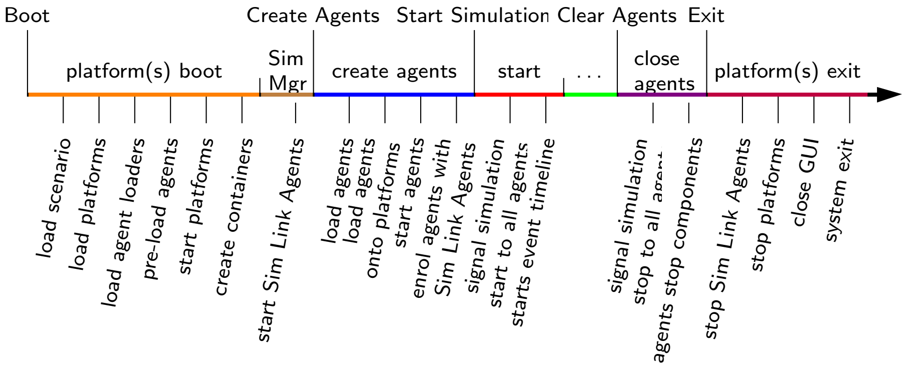
AgentManager

.start()
.stop()
.setPlatformLink()

· the platform is contacted by the agent's specialized components

An agent is loaded by an *AgentLoader*:
create the agent loader → *pre-load* the agent → *load* the agent
→ load the agent on the platform → start → enrol → start simulation

tATAmI-2 Architecture



tATAmI-2 Features

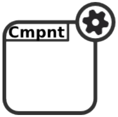
Use an XML scenario file to completely specify the initial configuration

```
<scen:platform><scen:parameter name="name" value="local" /></scen:platform>

<scen:initial><scen:container name="Container">
<scen:agent>
<scen:component name="parametric" />
<scen:component name="visualizable" />
<scen:component name="messaging" />
<scen:component name="testing" classpath="...PingBackTestComponent">
<scen:parameter name="other␣agent" value="AgentB" />
<scen:parameter name="initiator" value="true" />
</scen:component>
<scen:parameter name="loader" value="composite" />
<scen:parameter name="name" value="AgentA" />
</scen:agent>
<scen:agent>
<scen:component name="parametric" />
<scen:component name="visualizable" />
<scen:component name="messaging" />
<scen:component name="testing" classpath="...PingBackTestComponent" />
<scen:parameter name="loader" value="composite" />
<scen:parameter name="name" value="AgentB" />
</scen:agent>
</scen:container></scen:initial>
```

tATAmI-2 Features

Composite agents are formed of components which communicate by means of an event queue.



· and agent sees it as:

AgentComponent
.parentChangeNotifier()
.signalEvent()

· tATAmI-2 sees it as

.initialize()
.preload()

Loading a component: initialize → preload → add to agent
→ agent start → simulation start

tATAmI-2 Features

Example: the Messaging Component – abstracts messaging services

· A message is abstracted as a content sent between two endpoints
· An endpoint has an *external path* and an *internal path*

$$\underbrace{\texttt{jade:platform-1/AgentA}}_{\text{external path (agent address)}}\underbrace{\texttt{/VISUALIZATION/CONTROL}}_{\text{internal path}}$$

- can be extended by any component offering messaging services
- provides metods such as *send()*, *registerMessageHandler()*, *getAgentAddress()*
- is able to access the platform by using the *platform link*
- each platform is able to recommend a corresponding messaging component

Andrei Olaru – cs@andreiolaru.ro
CSCS'20
Bucharest, Romania 28.05.2015

Implementation

- tATAmI-2 core
- local messaging platform + corresponding component
- Jade messaging platform + corresponding component
- WebSocket messaging platform + corresponding component
- visualization, control, S-CLAIM interpreter
- various test components

Implementation

Andrei Olaru AI-MAS Group

- architecture
- main development

Marius-Tudor Benea AI-MAS Group

- S-CLAIM development
- Android development

Amal El-Fallah Seghrouchni

- tATAmI-1 coordination

Emma Sevastian

- scenario implementation

Cosmin Mihai

- WebSocket messaging

Thi Thuy Nga Nguyen

- tATAmI-1 development

Adina Magda Florea AI-MAS Group

- coordination

Implementation

- multiple platforms running at the same time ; same agent communicating through different means
- web service messaging
- conversation support
- Android deployment (supported in tATAmI-1)

Thank You!

Any Questions?