Simulation and Development of Multi-Agent Systems

Henrique Lopes Cardoso
FEUP/LIACC
hlc@fe.up.pt

November 2016

Outline

- Agents: what are they?
- Agent-Based Modeling and Simulation (ABMS)
- Multi-Agent Systems (MAS)
- Blending approaches (ABMS+MAS)
- SAJaS
- MASSim2Dev
- Summary and Extensions

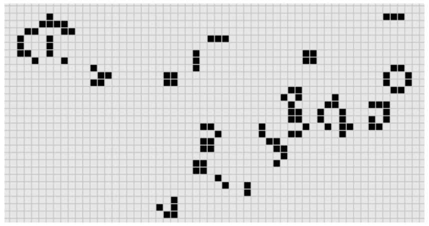
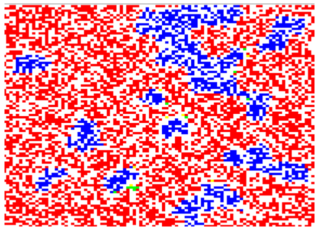
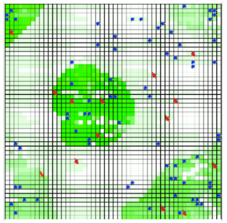
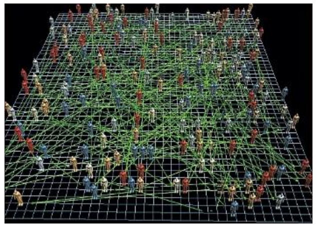
What are Agents?

- Agent-based computing
  - Software agent
  - Intelligent agent
  - Active unit in a complex system
  - Interacting social component in a multi-agent environment
  - Modeling, design and programming paradigm (AOP)

- Different perspectives depending on your background
  - Multi-agent systems (MAS), a subfield of AI
    - Computer science, software engineering
  - Agent-based modeling and simulation (ABMS)
    - Social sciences (e.g. sociology, psychology, economics, demography), biological sciences, environmental modeling, …

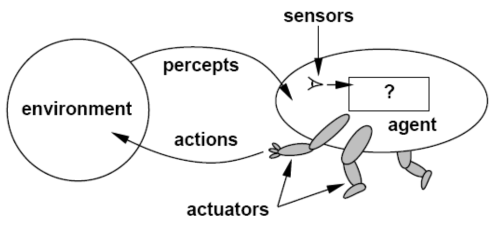
ABMS Perspective

- Agent-based model
  - computational model for simulating the actions and interactions of autonomous agents
  - assess overall properties or evolution of the system as a whole
- Goal
  - understand global or emergent phenomena associated with complex adaptive systems
- Agents are mostly homogeneous, simple, reactive…
  - … but there are exceptions to this

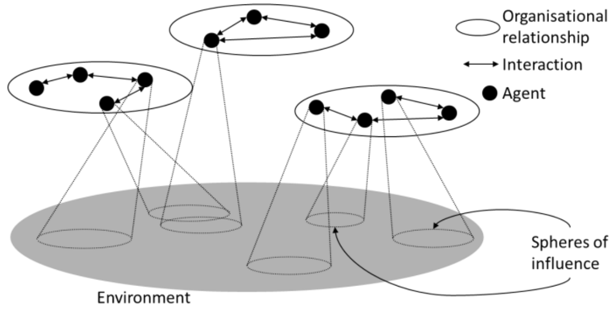





MAS Perspective

- An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives



- Intelligent agent
  - Reactivity: respond in timely fashion
  - Pro-activeness: goal-directed behavior
  - Social ability: interaction with other agents



- Multi-agent system
  - multiple interacting intelligent agents within an environment
  - solve problems that are difficult to model or impossible to solve using a monolithic system

- Focus on modeling actors in a system, and their interaction/coordination
  - agents are heterogeneous (architecturally, functionally), complex, adaptive
  - engineering perspective: validation of the future operation of actual agents

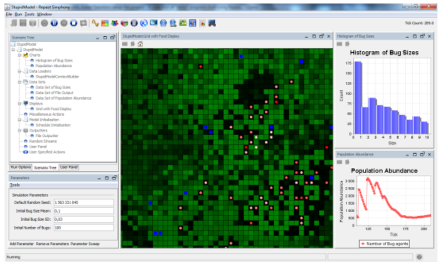
AGENT-BASED MODELING AND SIMULATION

Elements of an ABMS Tool

- Simulation concepts
  - Agents
  - Environment (space)
  - Model
- Scheduler
  - discrete-event simulator
  - time/tick stepped
  - agent-based
    - dynamic processes of agent behavior and interaction are simulated repeatedly over time
- Data collection and visualization
- Environment displays

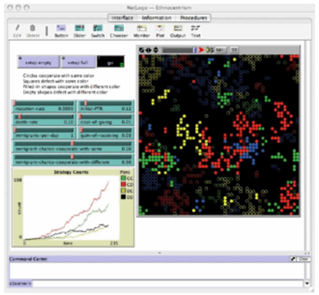
ABMS Tools

- Three of the most widely used ABMS tools:



repast

  - a family of advanced agent-based modeling and simulation platforms
    - Repast Simphony [North *et al.*, 2005]

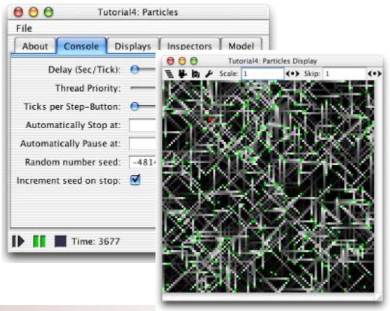


NetLogo

  - NetLogo [Tisue and Wilensky, 2004]
    - a multi-agent programmable modeling environment

M A S O N

  - MASON [Luke *et al.*, 2005]
    - a fast discrete-event multiagent simulation library core in Java



RepastS Model Constructs

- Context: a group of agents
  - Projections: impose structure on agents and the space where they are situated
    - Apply to all agents in the Context
    - Continuous Space, GIS, Grid, Network
    - Projections provide an API for moving, neighboring, connecting, …
  - Sub-contexts
    - Agents in a sub-context also exist in the parent context, but the reverse is not necessarily true
- Agent
  - POJOs
- `ContextBuilder` interface (Data Loader)

  ```
  Context build(Context<Object> context);
  ```
  - Add agents, create projections, define sub-contexts

RepastS Scheduler

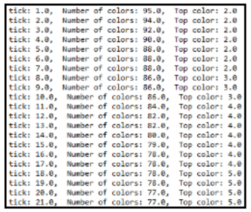
- Schedule agent actions
  - Methods of each class of objects that are in a Context
  - Three ways to work with the scheduler
    - Directly schedule a method invocation via the API (`ScheduleParameters`, `Schedule`)
    - Using Java annotations
    - Using Watchers (notifications of state changes in other agents)

❖ Note: In Repast3, model actions (e.g. display updates, data recording, snapshots) had to be scheduled through Java code as well. In Repast Simphony this is done via the GUI, and these actions do not feature in the code at all (they are managed by Repast's runtime infrastructure).

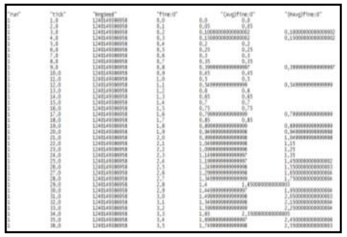
RepastS Data Collection/Vis.

- Data Sets
  - Tabular data where each column represents a data source
  - Data sources
    - Standard (e.g. tick count)
    - Method invocations on objects that are inside a context
    - Custom
  - Aggregate / Non-aggregate
  - Schedule parameters: start time, priority, interval, …

- Writing data: Text Sinks
  - A sink is associated with a data set
  - File / Console
  - Line / Tabular

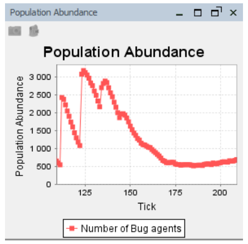


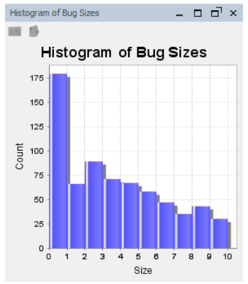
```
tick: 1.0,  Number of colors: 95.0,  Top color: 2.0
tick: 2.0,  Number of colors: 94.0,  Top color: 2.0
tick: 3.0,  Number of colors: 92.0,  Top color: 2.0
tick: 4.0,  Number of colors: 90.0,  Top color: 2.0
tick: 5.0,  Number of colors: 88.0,  Top color: 2.0
tick: 6.0,  Number of colors: 88.0,  Top color: 2.0
tick: 7.0,  Number of colors: 88.0,  Top color: 2.0
tick: 8.0,  Number of colors: 86.0,  Top color: 3.0
tick: 9.0,  Number of colors: 86.0,  Top color: 3.0
tick: 10.0,  Number of colors: 86.0,  Top color: 3.0
tick: 11.0,  Number of colors: 84.0,  Top color: 4.0
tick: 12.0,  Number of colors: 82.0,  Top color: 4.0
tick: 13.0,  Number of colors: 82.0,  Top color: 4.0
tick: 14.0,  Number of colors: 80.0,  Top color: 4.0
tick: 15.0,  Number of colors: 79.0,  Top color: 4.0
tick: 16.0,  Number of colors: 78.0,  Top color: 4.0
tick: 17.0,  Number of colors: 78.0,  Top color: 4.0
tick: 18.0,  Number of colors: 78.0,  Top color: 5.0
tick: 19.0,  Number of colors: 78.0,  Top color: 5.0
tick: 20.0,  Number of colors: 77.0,  Top color: 5.0
tick: 21.0,  Number of colors: 77.0,  Top color: 5.0
```



[illegible]

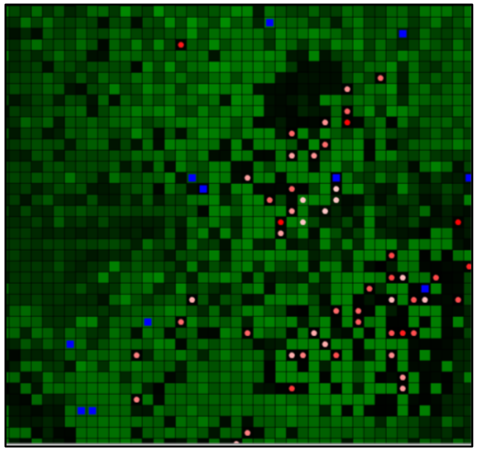
- Visualizing data: Charts
  - A chart is associated with a data set
  - Time Series
  - Histogram (non-aggregate data)

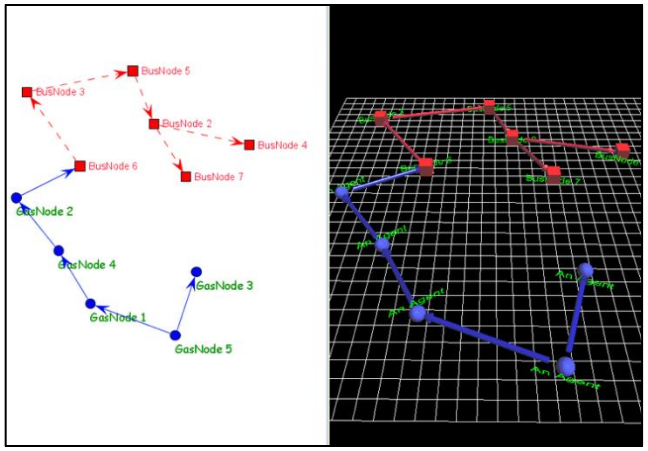


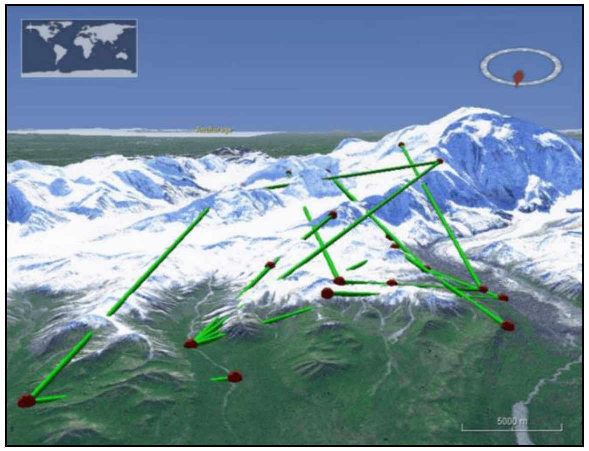


RepastS Environment Displays

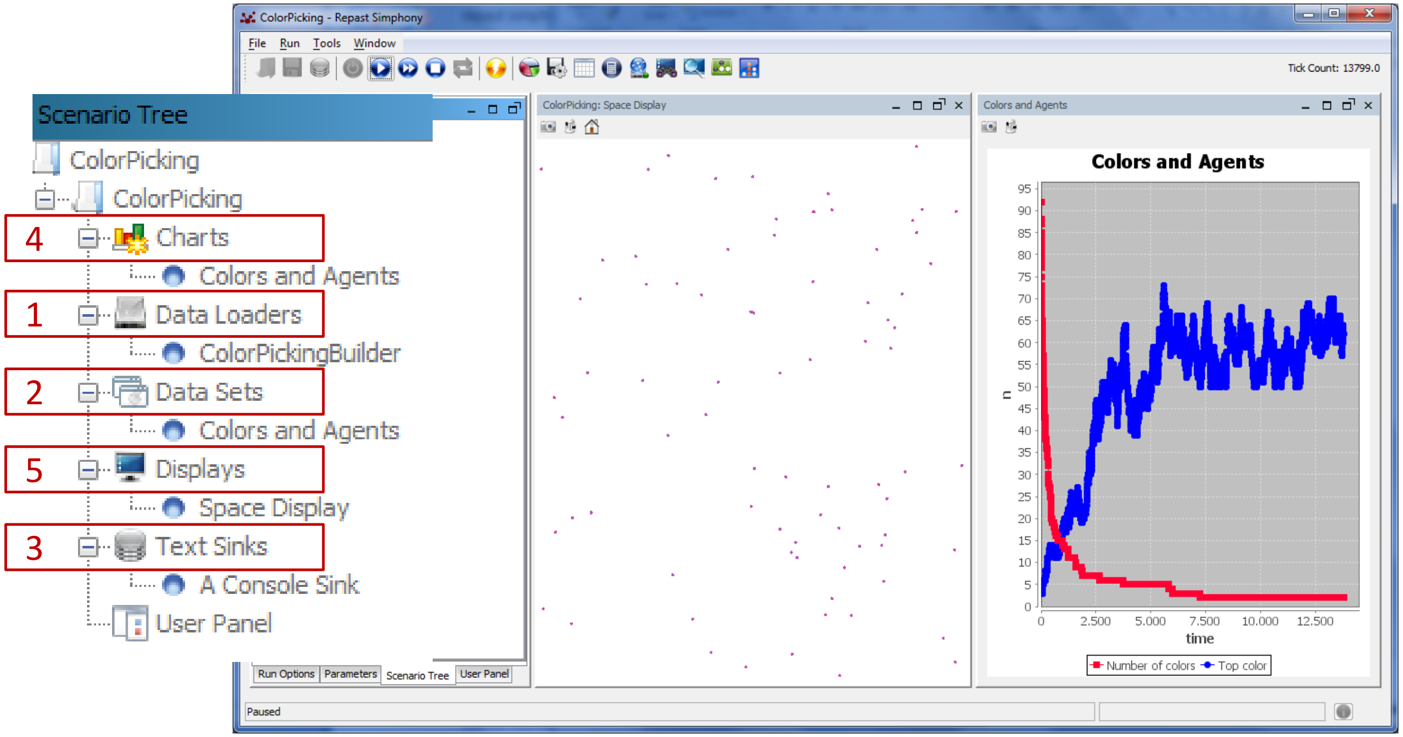
- Displays
  - Associated with projections
  - Chose which and how agents will be visualized
    - Class name
    - Style
  - Schedule parameters: start time, priority, interval, …
  - 2D, 3D







RepastS GUI


ColorPicking - Repast Simphony
File Run Tools Window
Tick Count: 13799.0
Scenario Tree
ColorPicking
ColorPicking
4 Charts
Colors and Agents
1 Data Loaders
ColorPickingBuilder
2 Data Sets
Colors and Agents
5 Displays
Space Display
3 Text Sinks
A Console Sink
User Panel
Run Options Parameters Scenario Tree User Panel
ColorPicking: Space Display
Colors and Agents
Colors and Agents
n
time
Number of colors
Top color
Paused

MULTI-AGENT SYSTEMS

MAS Software Engineering

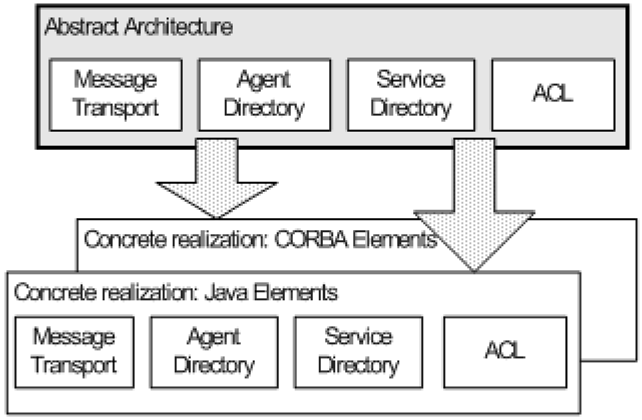
- AOSE (Agent-Oriented Software Engineering)
  - Abstractions: agent, environment, interaction protocol, context, roles, organizations, BDI
  - Methodologies: Gaia, MaSE, Prometheus, Tropos, …
- MAS programming constructs
  - Agents (internal architecture and building blocks)
  - Infrastructure
    - Environment
    - Interaction artifacts/protocols (communication)
    - Distribution, mobility
- Development tools
  - IDE plugins, debugging
  - Agent and MAS visualization

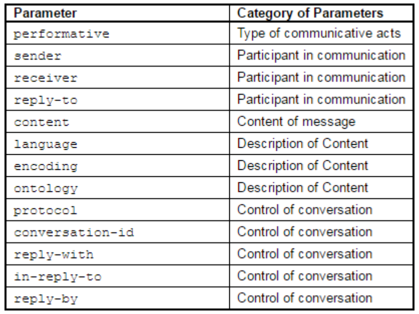
MAS Development

- Some examples of platforms…
  - JADE
  - Jadex
  - Cougaar
  - Brahms
- …and languages…
  - Jason (AgentSpeak)
  - 2APL
  - Concurrent MetateM
- …and organizational/environment modeling and programming
  - Moise
  - CArtAgO

Standardization: FIPA

- FIPA Abstract Architecture Specification

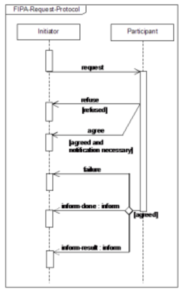


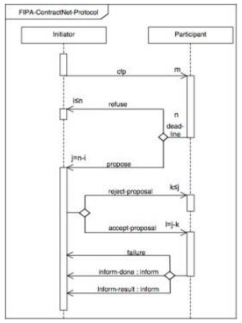


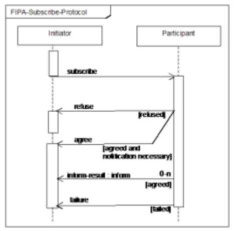
Parameter	Category of Parameters
performative	Type of communicative acts
sender	Participant in communication
receiver	Participant in communication
reply-to	Participant in communication
content	Content of message
language	Description of Content
encoding	Description of Content
ontology	Description of Content
protocol	Control of conversation
conversation-id	Control of conversation
reply-with	Control of conversation
in-reply-to	Control of conversation
reply-by	Control of conversation

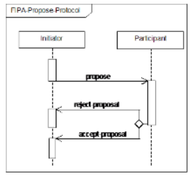
- Agent Communication Language (ACL) Specifications
  - Message Structure, Communicative Act Library, Content Languages, Interaction Protocols

- Agent Management
- Agent Message Transport



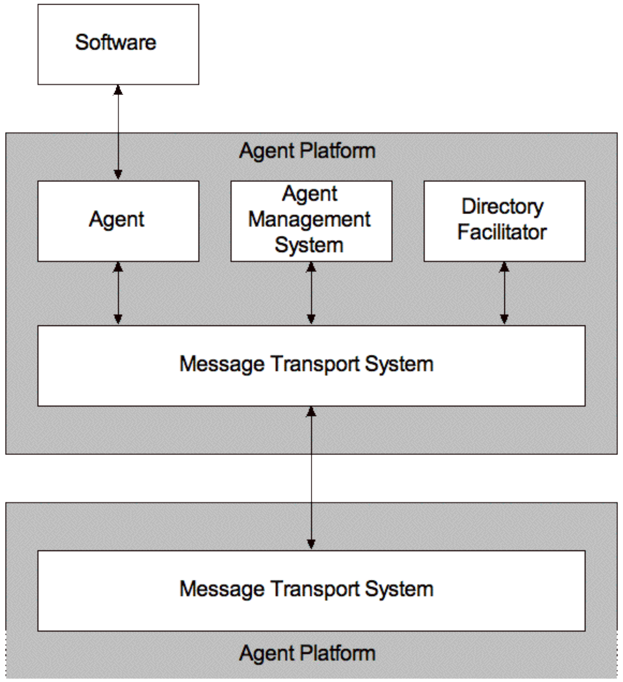




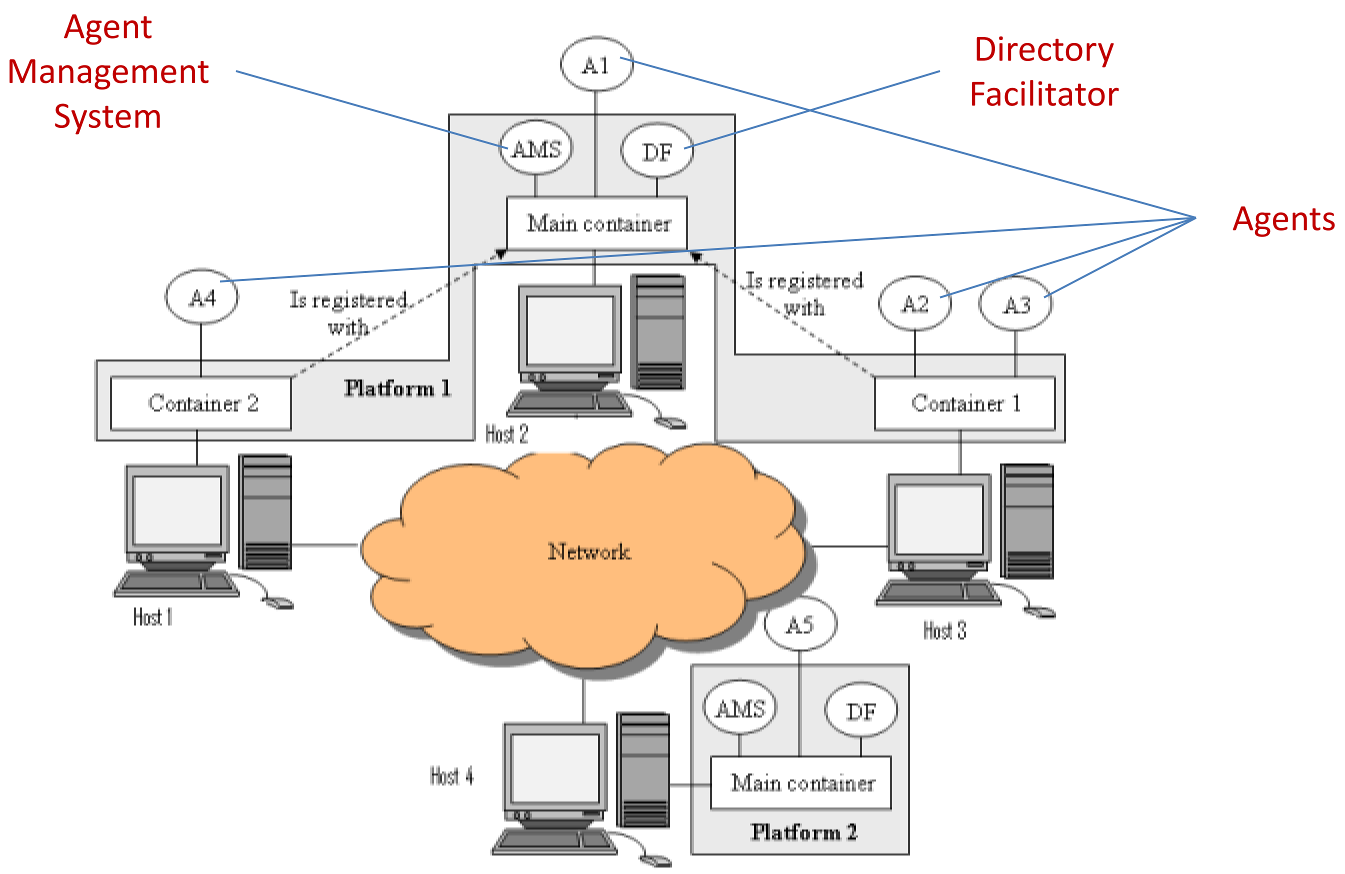


JADE

- An Java framework for developing multi-agent systems
- FIPA-compliant
  - Agent Platform
    - Agent Management System (AMS)
    - Directory Facilitator (DF)
    - Message Transport System (MTS)
  - Agent Communication Language (ACL)
  - Interaction Protocols



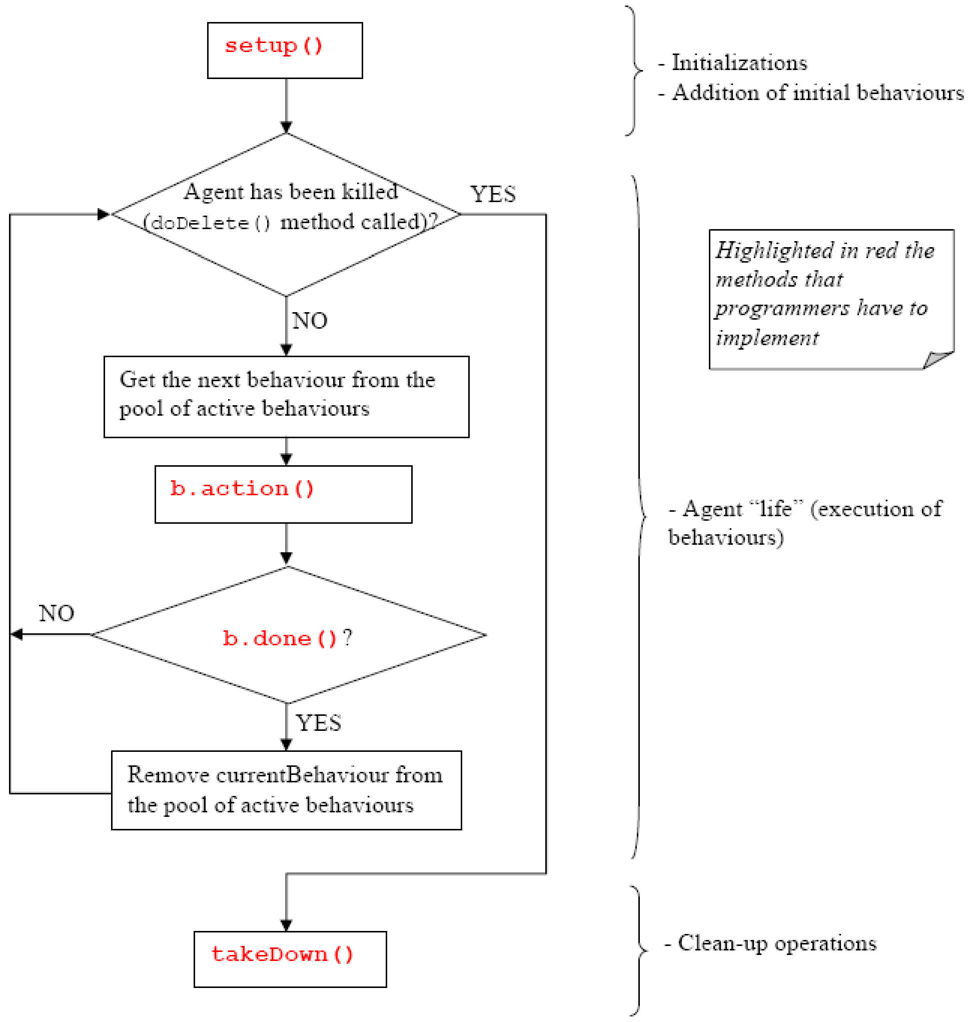
JADE Architecture



JADE Programming

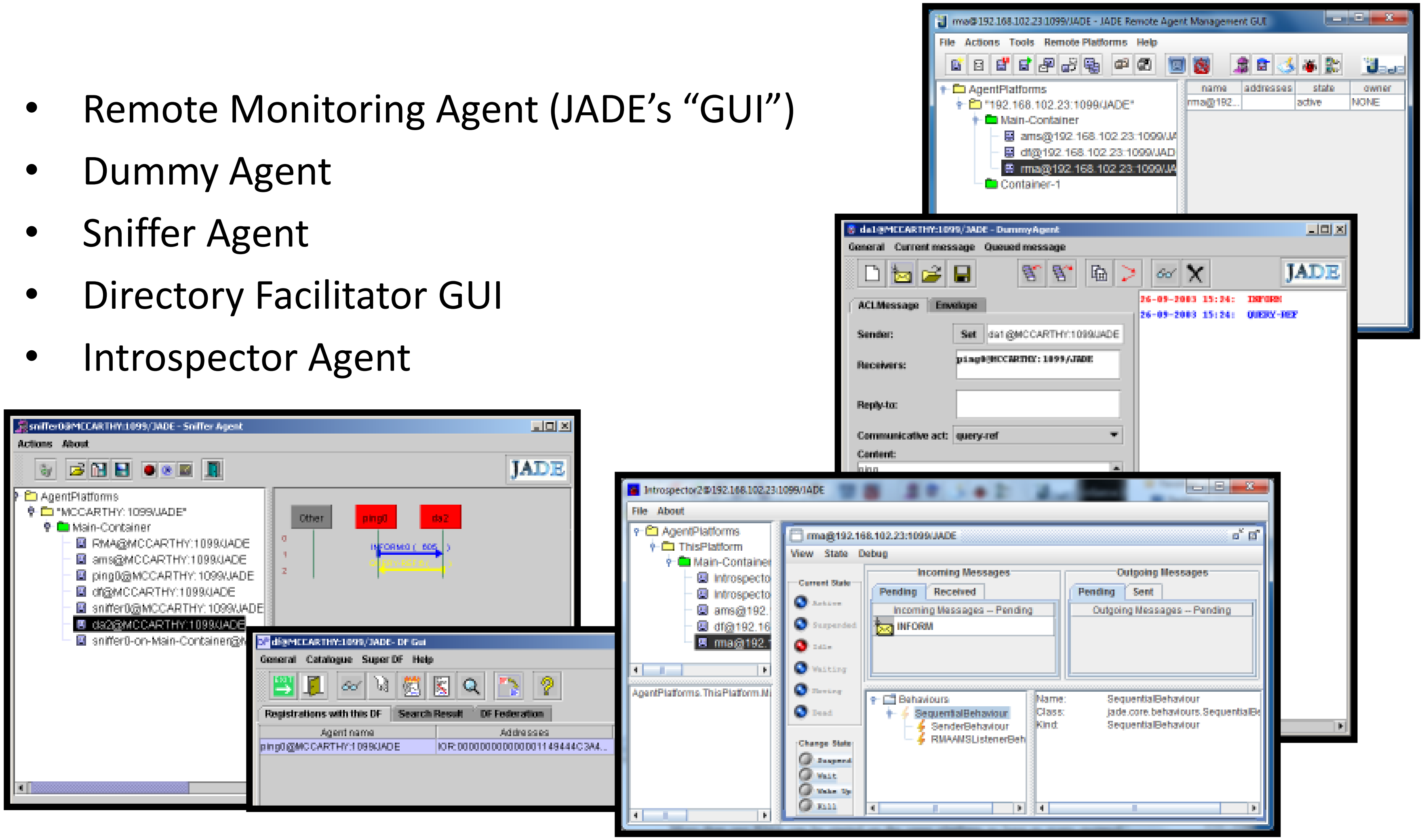
- Agents
  - 1 Agent = 1 Thread
  - Main construct: behaviour
    - Tasks, executed concurrently
  - Behaviour scheduling
    - Not preemptive, but “cooperative” (thread sharing)
    - Conceptually, behaviours should be seen as parallel
  - Communication using ACLMessages
  - Mobility and cloning
    - Agents can migrate throughout containers
- JADE API includes, among other things:
  - Several behaviour classes
  - ACL: messages, interaction protocols, ontologies
  - DFService / AMSService
- Deployment: distributed MAS
  - Agents execute within containers (JVMs)

JADE Agent Execution


setup()
- Initializations
- Addition of initial behaviours
Agent has been killed (doDelete() method called)?
YES
NO
Highlighted in red the methods that programmers have to implement
Get the next behaviour from the pool of active behaviours
b.action()
- Agent "life" (execution of behaviours)
NO
b.done()?
YES
Remove currentBehaviour from the pool of active behaviours
takeDown()
- Clean-up operations

JADE Tools

- Remote Monitoring Agent (JADE’s “GUI”)
- Dummy Agent
- Sniffer Agent
- Directory Facilitator GUI
- Introspector Agent


rma@192.168.102.23:1099/JADE - JADE Remote Agent Management GUI
File Actions Tools Remote Platforms Help
da1@MCCARTHY:1099/JADE - DummyAgent
General Current message Queued message
ACLMessage Envelope
Sender: Set da1@MCCARTHY:1099/JADE
Receivers: ping0@MCCARTHY:1099/JADE
Reply-to:
Communicative act: query-ref
Content:
26-09-2003 15:24: INFORM
26-09-2003 15:24: QUERY-REF
sniffer0@MCCARTHY:1099/JADE - Sniffer Agent
Actions About
Other ping0 da2
df@MCCARTHY:1099/JADE- DF Gui
General Catalogue Super DF Help
Registrations with this DF Search Result DF Federation
Agent name Addresses
ping0@MCCARTHY:1099/JADE
Introspector2@192.168.102.23:1099/JADE
File About
Incoming Messages Outgoing Messages
Pending Received Pending Sent
Behaviours SequentialBehaviour SenderBehaviour RMAAMSListenerBeh
Name: SequentialBehaviour
Class: jade.core.behaviours.SequentialBe
Kind: SequentialBehaviour

BLENDING APPROACHES

Rationale

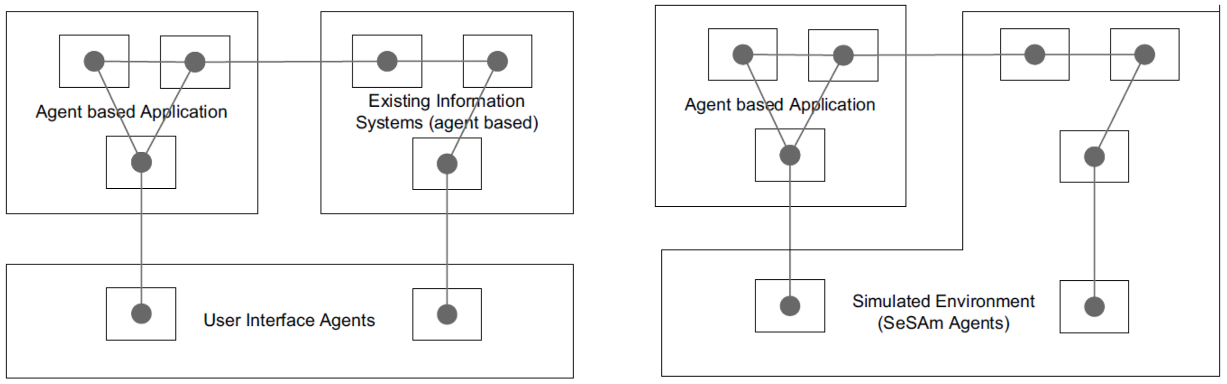
- MAS provide powerful abstractions and mechanisms for effectively modelling real-world applications that are highly complex and dynamic
  - manufacturing, e-commerce, network management, distributed sensing and control, information retrieval, …
- MAS need to be validated before being deployed and executed in real operating environments
  - scale and complexity of systems are too demanding to be managed in real execution testing scenarios
- Methodologies that support system validation through simulation are required
  - discrete-event simulation, agent-based simulation

Approaches

- ABMS for agent-based software development
  - SeSAm [Klügl *et al.*, 2003]
- Multi-agent based simulation (MABS)
  - Domain-specific
    - MATSim [Balmer et al., 2008]
    - PlaSMA [Warden *et al.*, 2010]
    - MASeRaTi [Ahlbrecht *et al.*, 2014]
  - General purpose
    - Jadex [Braubach *et al.*, 2012]
- Extensions
  - NetLogo+BDI+ACL agent programming [Sakellariou *et al.*, 2008]
  - JADE+simulation
    - MISIA [García *et al.*, 2011]
    - JRep [Gormer *et al.*, 2011]
    - PlaSMA [Warden *et al.*, 2010]

SeSAm

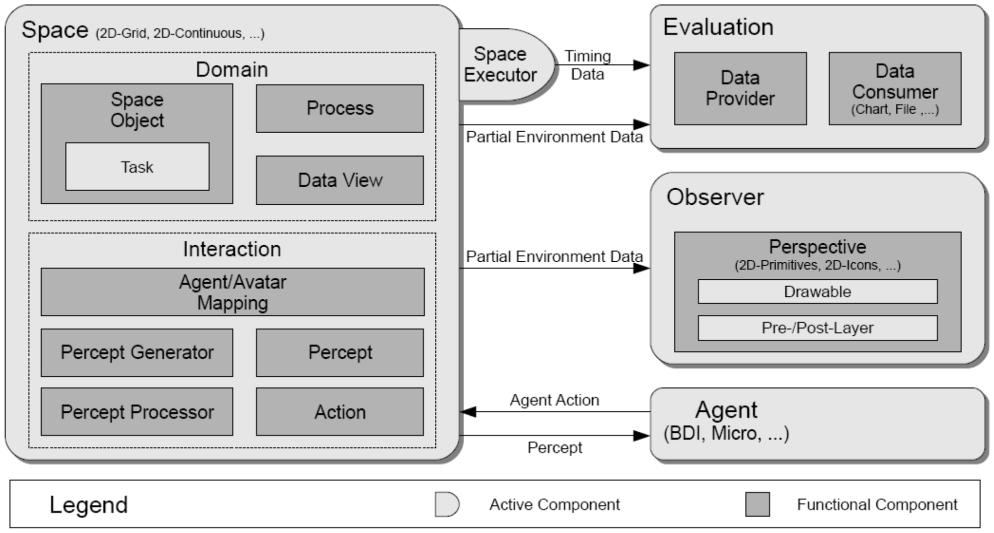
- Shell for Simulated Agent Systems
  - Relating agent-based simulation and software development
  - Virtual environments for agent based software

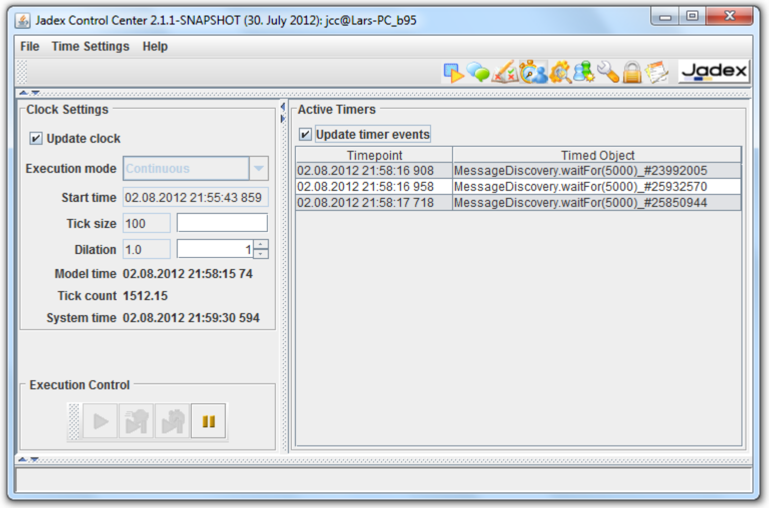

Agent based Application
Existing Information Systems (agent based)
User Interface Agents
Agent based Application
Simulated Environment (SeSAm Agents)

(a) Real world scenario (b) Simulated testbed scenario

Jadex

- Active components: unified execution infrastructure for agents and workflows
- BDI agents
- Applications executable as simulations as well as real time




Jadex Control Center 2.1.1-SNAPSHOT (30. July 2012): jcc@Lars-PC_b95
File
Time Settings
Help
Jadex
Execution Control

Clock Settings

☑ Update clock

Execution mode: Continuous
Start time: 02.08.2012 21:55:43 859
Tick size: 100
Dilation: 1.0 | 1
Model time: 02.08.2012 21:58:15 74
Tick count: 1512.15
System time: 02.08.2012 21:59:30 594

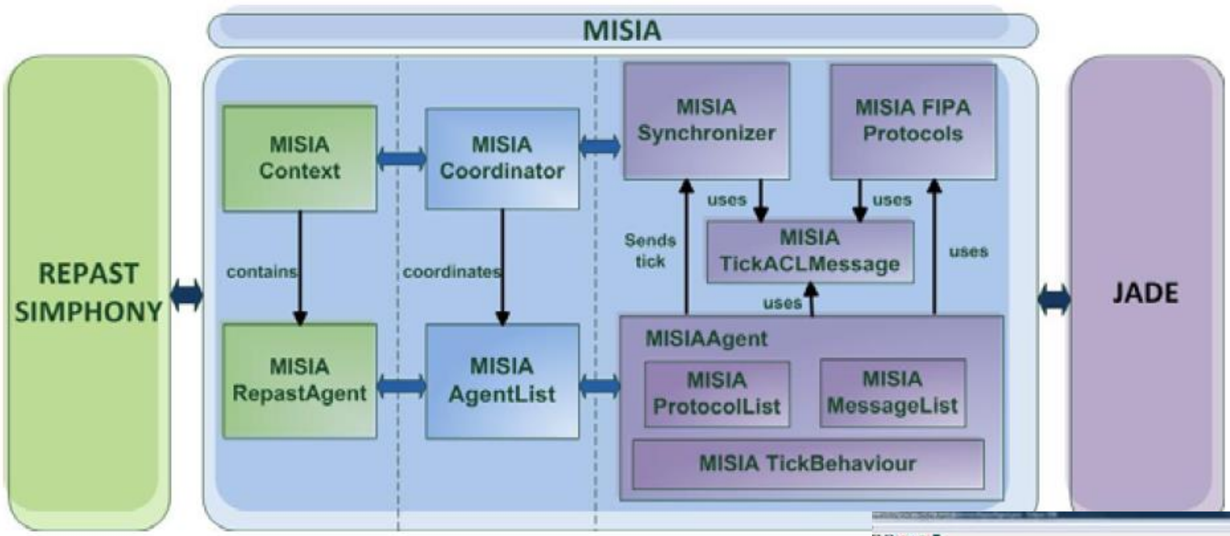
Active Timers

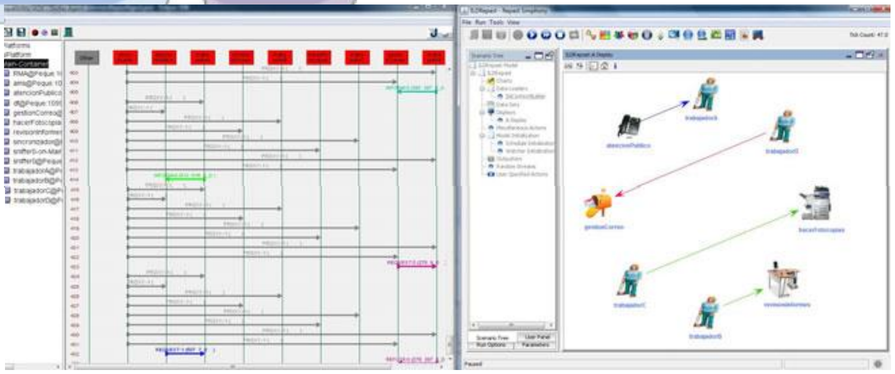
☑ Update timer events

Timepoint	Timed Object
02.08.2012 21:58:16 908	MessageDiscovery.waitFor(5000)_#23992005
02.08.2012 21:58:16 958	MessageDiscovery.waitFor(5000)_#25932570
02.08.2012 21:58:17 718	MessageDiscovery.waitFor(5000)_#25850944

MISIA

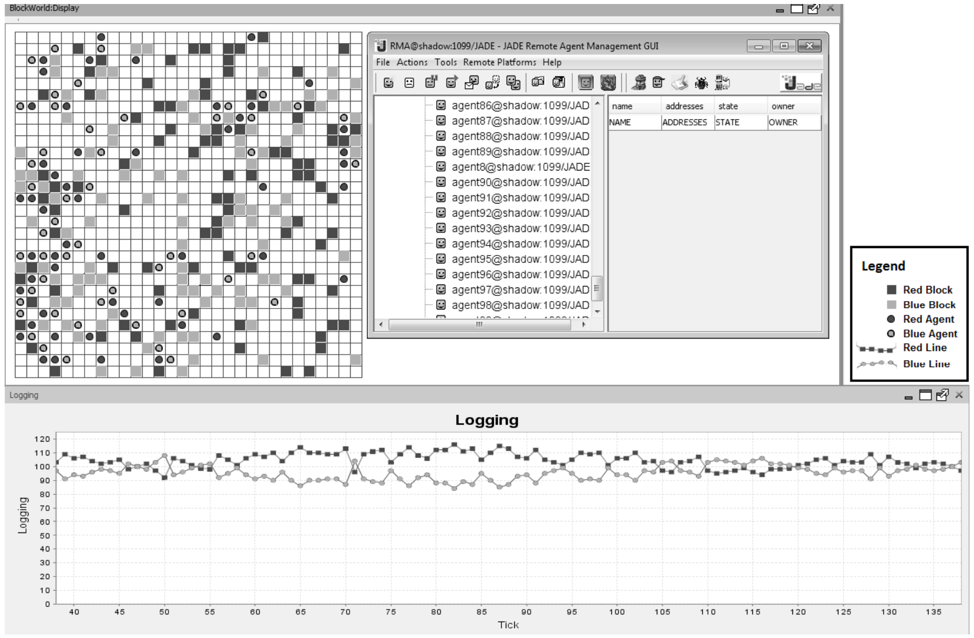
- Middleware Infrastructure to Simulate Intelligent Agents





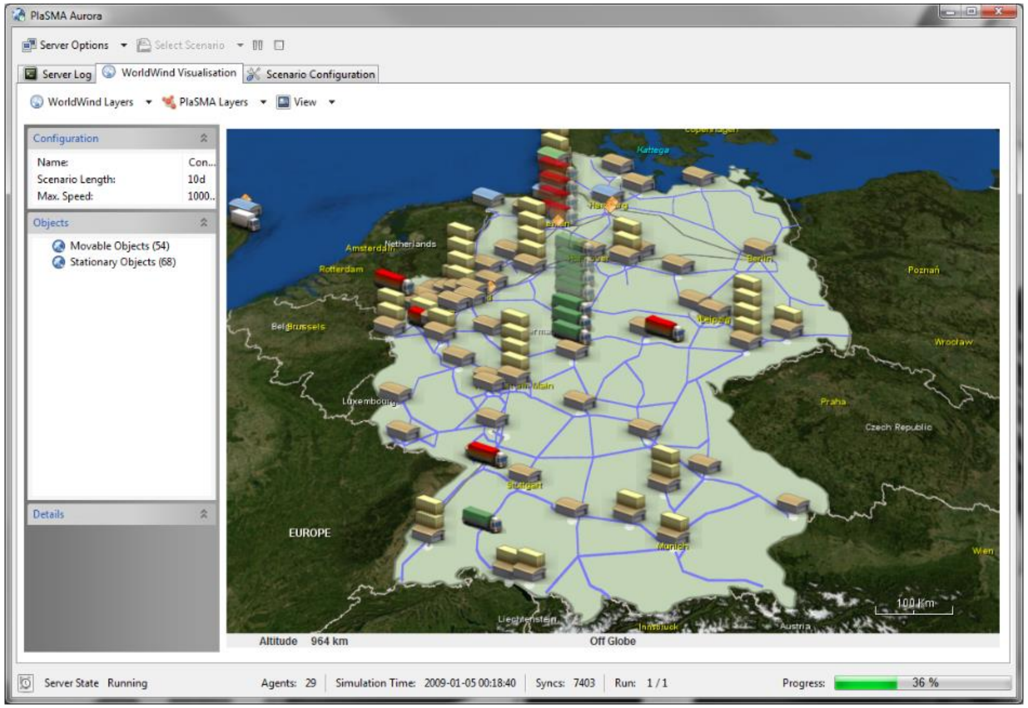
JRep

- Integration of JADE and Repast Simphony



PlaSMA

- Platform for Simulations with Multiple Agents (logistics domain)
  - JADE extension: simulation control (synchronization) and world model (ontology)



Name: Con...
Scenario Length: 10d
Max. Speed: 1000..

Altitude 964 km   Off Globe

Server State Running   Agents: 29   Simulation Time: 2009-01-05 00:18:40   Syncs: 7403   Run: 1 / 1   Progress: 36 %

Simple API for JADE-based Simulations

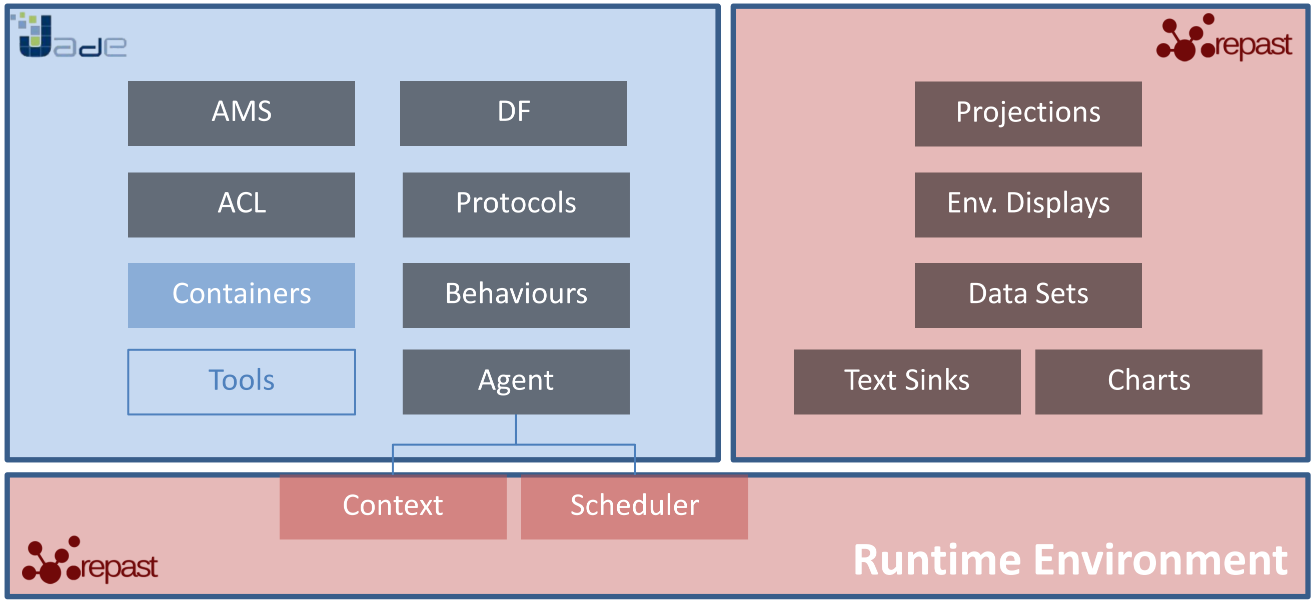
http://web.fe.up.pt/~hlc/doku.php?id=sajas

SAJaS

SAJaS: why?

- JADE:
  - Multi-agent systems development
  - Not suited for multi-agent based simulation (MABS): scalability
- Repast:
  - Agent-based simulation
  - Lack support for agent programming and multi-agent features (communication, infrastructure, …)
- However:
  - Need to **simulate while developing** a full-featured MAS, for testing purposes

Architecture

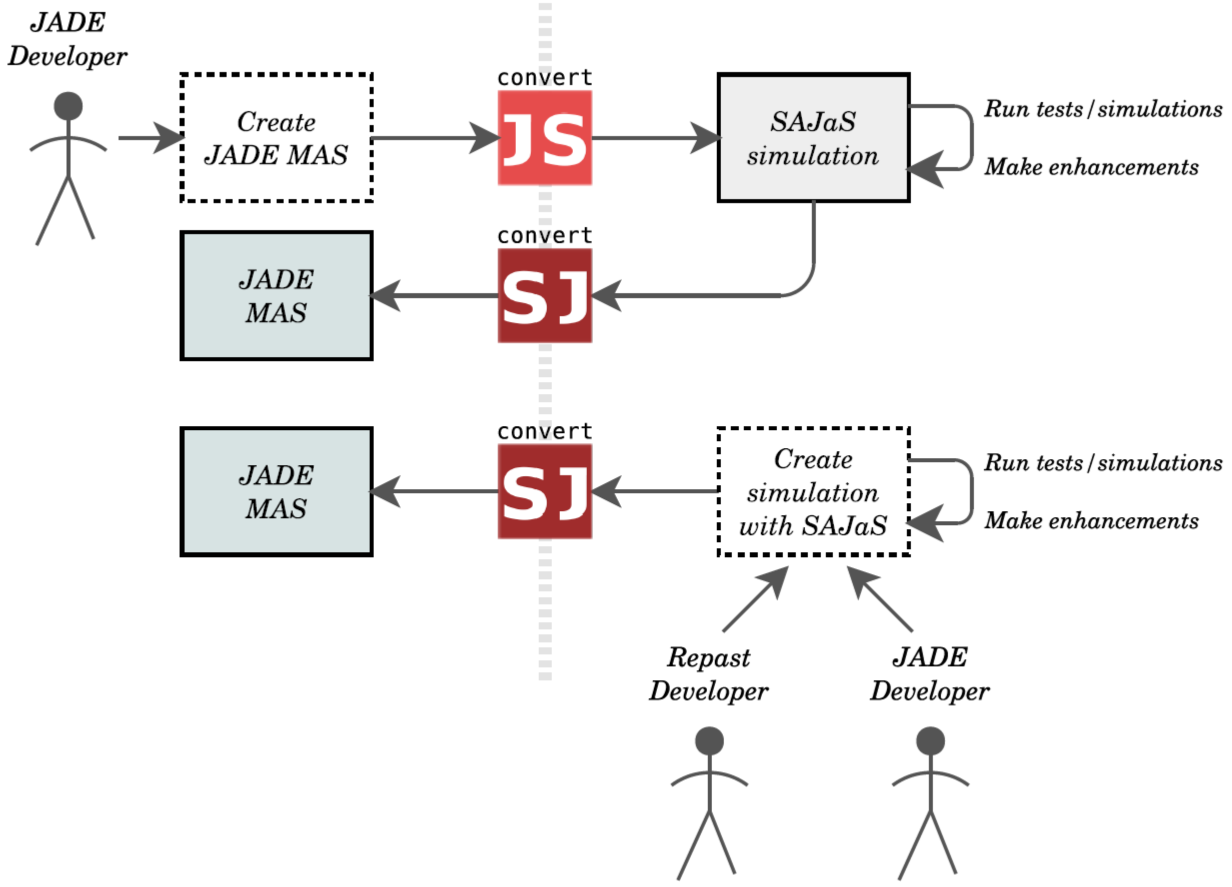


SAJaS Features

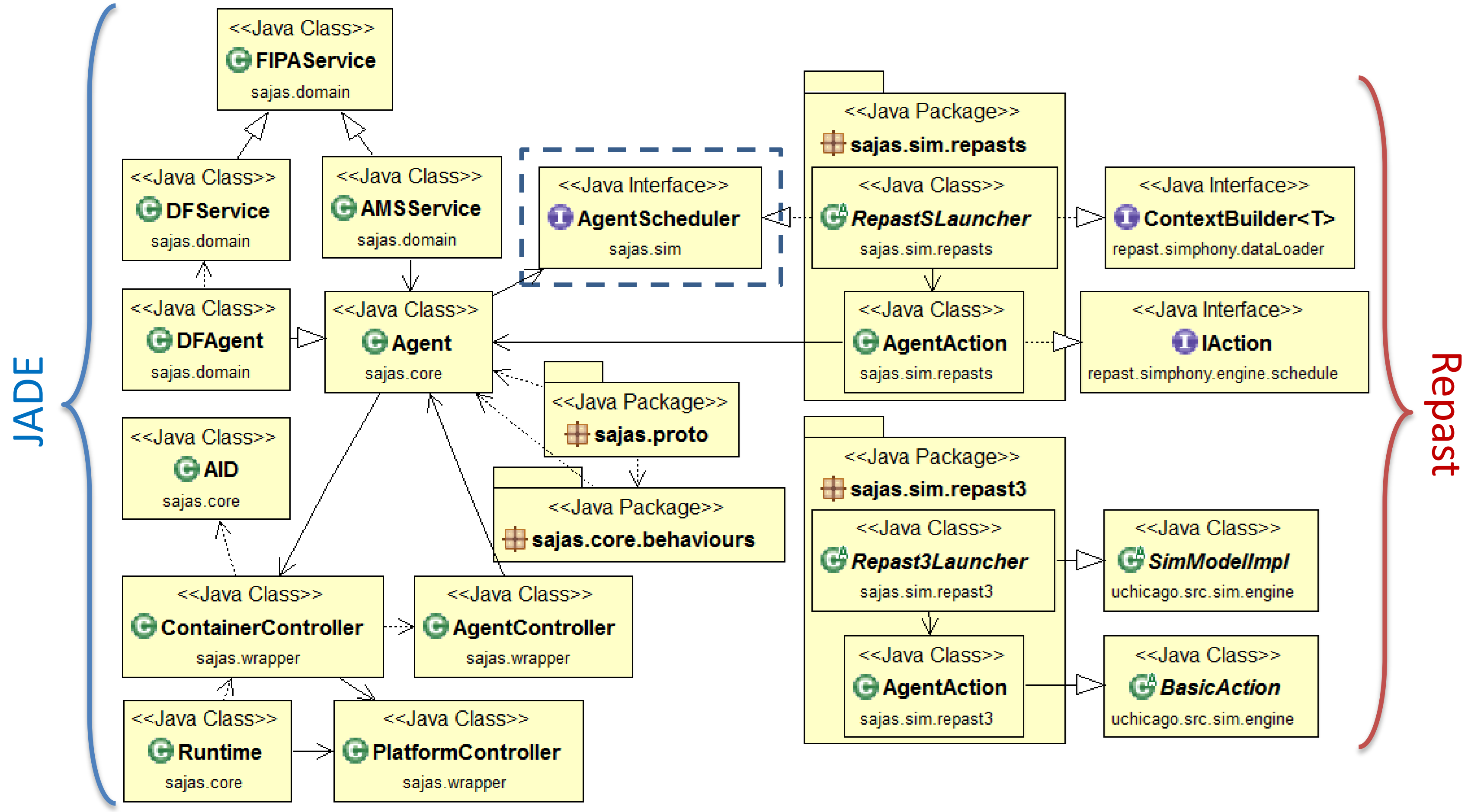
	JADE	Repast
Distributed	Yes	No
Simulation Tools	No	Yes
Scalability	Limited	High
Open Source	Yes	Yes
Agent Execution	Behaviours	Scheduler
	Multi-thread	Single-thread
	Event-driven	Tick-driven
	Asynchronous	Synchronous
Interaction	FIPA ACL	Method calls
		Shared resources
Ontologies	Yes	No

☐ from the simulation scheduler point of view

Usage

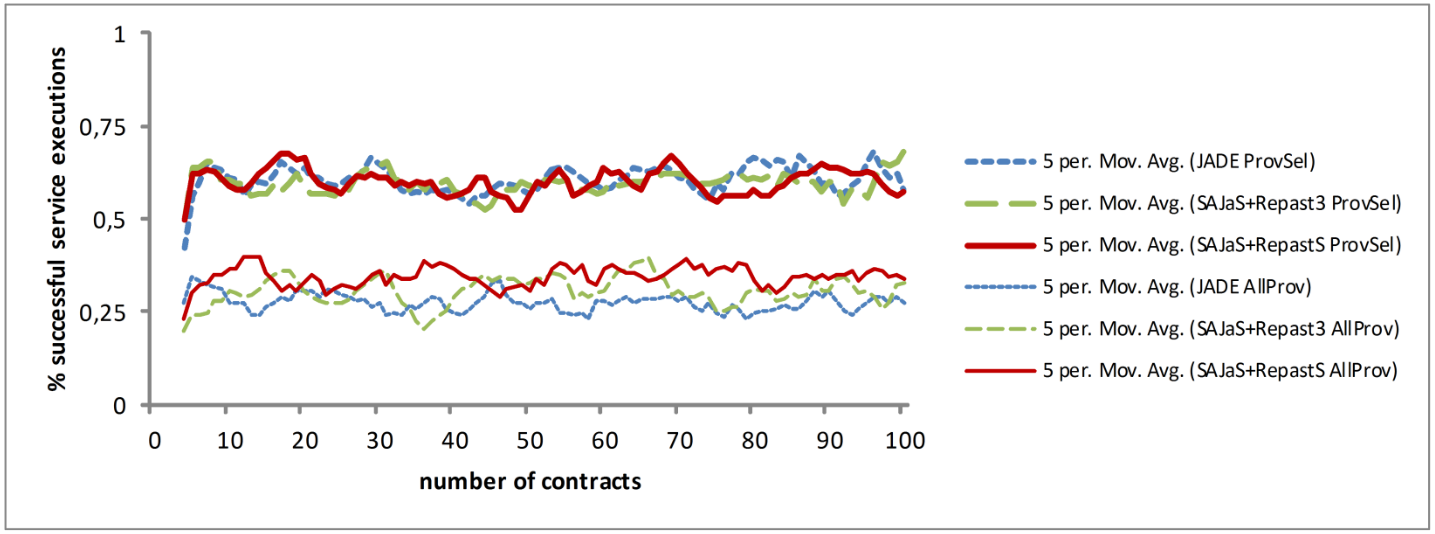

JADE Developer
Create JADE MAS
convert
JS
SAJaS simulation
Run tests / simulations
Make enhancements
convert
SJ
JADE MAS
convert
SJ
JADE MAS
Create simulation with SAJaS
Run tests / simulations
Make enhancements
Repast Developer
JADE Developer

SAJaS API Overview


JADE
Repast
<<Java Class>> FIPAService sajas.domain
<<Java Class>> DFService sajas.domain
<<Java Class>> AMSService sajas.domain
<<Java Interface>> AgentScheduler sajas.sim
<<Java Package>> sajas.sim.repasts
<<Java Class>> RepastSLauncher sajas.sim.repasts
<<Java Interface>> ContextBuilder<T> repast.simphony.dataLoader
<<Java Class>> DFAgent sajas.domain
<<Java Class>> Agent sajas.core
<<Java Class>> AgentAction sajas.sim.repasts
<<Java Interface>> IAction repast.simphony.engine.schedule
<<Java Package>> sajas.proto
<<Java Class>> AID sajas.core
<<Java Package>> sajas.core.behaviours
<<Java Package>> sajas.sim.repast3
<<Java Class>> Repast3Launcher sajas.sim.repast3
<<Java Class>> SimModelImpl uchicago.src.sim.engine
<<Java Class>> ContainerController sajas.wrapper
<<Java Class>> AgentController sajas.wrapper
<<Java Class>> AgentAction sajas.sim.repast3
<<Java Class>> BasicAction uchicago.src.sim.engine
<<Java Class>> Runtime sajas.core
<<Java Class>> PlatformController sajas.wrapper

Benchmark Scenario

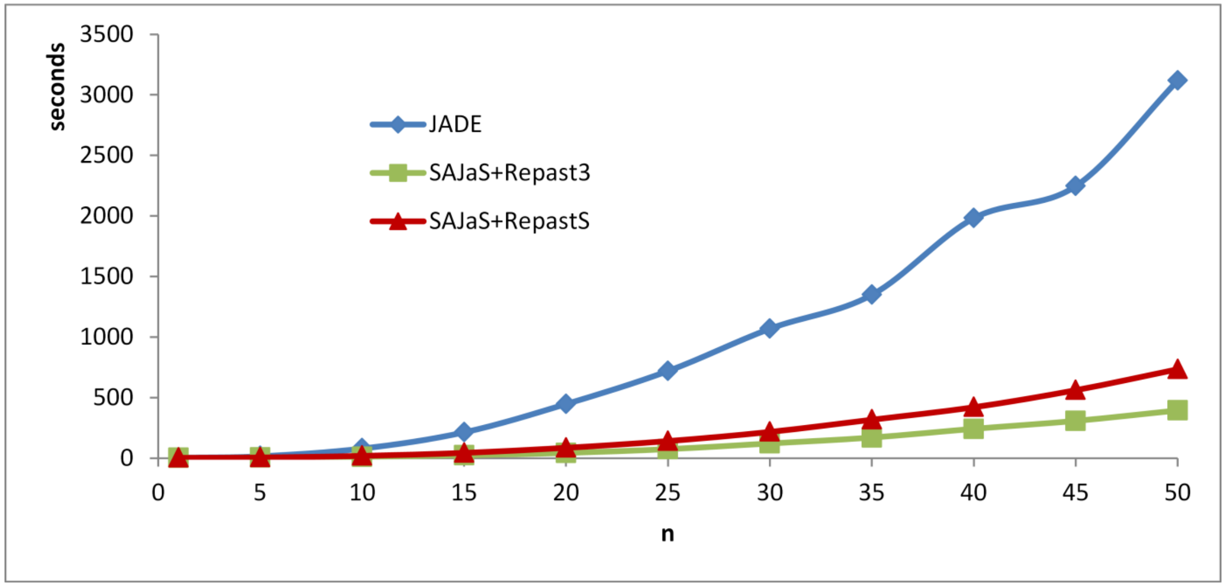
- Service Consumer/Provider
  - $5 \times n$ providers; $2 \times n$ consumers (AllProv, ProvSel)
  - DF, behaviours, FIPA-protocols, ACL, ontologies



Henrique Lopes Cardoso (2015). “SAJaS: Enabling JADE-Based Simulations”, *Transactions on Computational Collective Intelligence XX*, N.T. Nguyen et al. (Eds.), LNCS 9420, pp. 158-178, Springer.

Benchmark Scenario

- Service Consumer/Provider
  - $5 \times n$ providers; $2 \times n$ consumers (AllProv, ProvSel)
  - DF, behaviours, FIPA-protocols, ACL, ontologies



Henrique Lopes Cardoso (2015). "SAJaS: Enabling JADE-Based Simulations", *Transactions on Computational Collective Intelligence XX*, N.T. Nguyen et al. (Eds.), LNCS 9420, pp. 158-178, Springer.

Main Advantages

- MABS programmer
  - has a rich set of multi-agent programming features offered by JADE
  - may explore simulation-related features offered by the simulation infrastructure (e.g. Repast)
- Same implementation can be used both for simulation and deployment purposes
  - checkout *MASSim2Dev*
- Simulation performance gains in certain scenarios: high communication-to-computation ratio

MAS Simulation to Development

https://web.fe.up.pt/~hlc/doku.php?id=massim2dev

MASSim2Dev

SAJaS and MASSim2Dev

- SAJaS re-implements some core JADE classes
  - Core classes
    - Agent, AID
  - Runtime infrastructure
    - Runtime, PlatformController, ContainerController, AgentController
  - FIPA services
    - FIPAService, AMSService, DFService, DFAgent
  - Agent dependencies
    - Behaviours, protocols

- MASSim2Dev Eclipse plugin
  - Goal: automate project conversion: JADE ⇔ SAJaS
  - Currently supports three types of conversion:
    - SAJaS → JADE
    - JADE → SAJaS+Repast3
    - JADE → SAJaS+RepastS

Steps

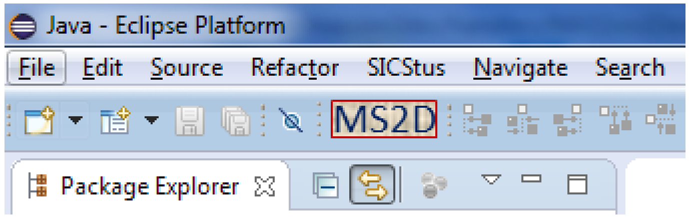
1. Clone Java project

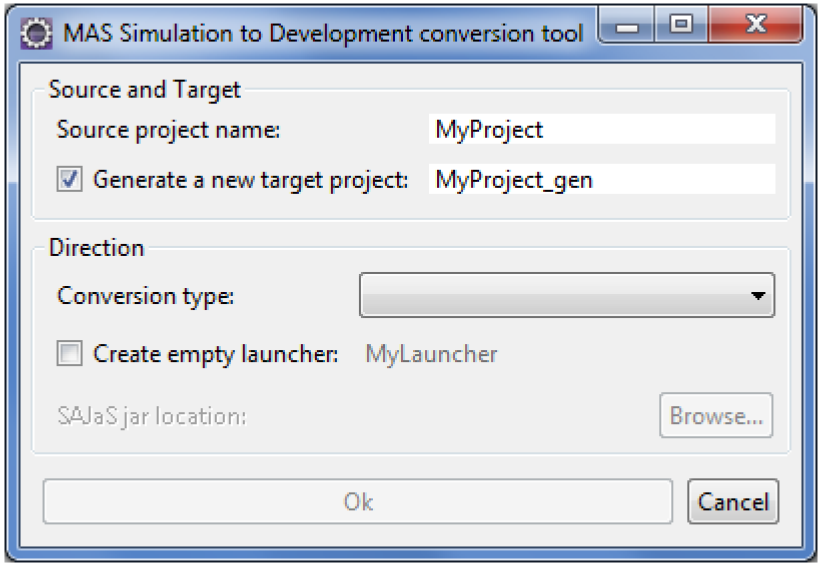
2. Refactor source files
   - Change project dependencies on JADE/SAJaS to the equivalent classes in SAJaS/JADE
   - JADE-SAJaS dictionary contains a mapping of classes

3. Create empty launcher
   - JADE, Repast3 or RepastS

4. Set build path
   - If JADE $\rightarrow$ SAJaS, add SAJaS library

GUI





Source and Target

Source project name: MyProject

[✓] Generate a new target project: MyProject_gen

Direction

Conversion type:

[ ] Create empty launcher: MyLauncher

SAJaS jar location: Browse...

Ok Cancel

SUMMARY

MAS Simulation vs Development

- Multi-Agent based Simulation (MABS)
  - Computer simulation where entities are modeled and implemented as agents
  - Agent-based simulation tools
    - Discrete-events, focus on performance, large scale, interaction environment
    - Lack of support for agent programming and MAS infrastructures
- Multi-Agent System
  - System composed of autonomous, intelligent and interacting agents
  - Development tools
    - Support for communication, distribution, standards (FIPA)
    - Multi-threaded, limited scalability, not appropriate for simulation
- MABS can be useful while developing MAS applications

SAJaS+MASSim2Dev

- Bridge simulation and development of MAS
  - Develop high-performance simulations using MAS development features: “MAS-like MABS”
  - Convert MABS into MAS automatically (write-once, simulate and deploy)

- Approach
  - Simple API for JADE-based Simulations (SAJaS)
    - Light implementation of most JADE features
    - Integration with simulation framework (e.g. Repast)
  - MAS Simulation to Development (MASSim2Dev)
    - Eclipse plugin conversion tool: JADE ⇔ SAJaS

Extensions

- Enhancement of facilities included in SAJaS/MASSim2Dev
  - portable data collection and visualization tools
    - between Repast and JADE, through MASSim2Dev
  - additional simulation and conversion options
- Large-scale JADE-based BDI Simulation
  - enrich SAJaS with BDI reasoning agents
    - Jason
  - BDI simulation scalability
    - distributed vs shared BDI reasoning engine