Workshop Agent Technology

intro

1. java -cp jade.jar jade.Boot -gui (-detect-main false)
2. java -cp jade.jar jade.Boot -container -host <your ip>
3. start some dummy agents
4. registrer agents
5. search agents
6. construct and send messages
7. experiment with the sniffer (agent)
8. experiment with the introspector (agent)
9. experiment with the logmanager (agent)
10. move an agent to another container

Panic application

requirements

- smileys only “see” messages within range
- door and smileys send their location to the room
- smileys periodically send a movement to the room
- room informs smileys near the door about its location
- room informs smileys who their neighbours are
- smileys pass the door location to their neighbours

Smileys try to reach the exit

Parts

Room

- w/h, gui
- receive messages behavior
- update gui behavior
- inform smileys behavior

Door

- x/y
- receive messages behavior

Smileys

- x/y, dx/dy
- receive messages behavior
- movement behavior

environment

AGENT-LIBRARY

sources for generic code

libraries for CNS- projects

WORKSHOP

uses AGENT-LIBRARY

build.properties

build.xml → all

run (check properties → host/port):

ant -f run.xml <target>

targets:

runall = run local, config/jadeall.properties

runPlatform = main-container with room and door,
config/jademain.properties

runMediator = mediator container, config/jademediator.properties

runSmilies = container with smileys, config/jadesmilies.properties

ide classpath

Jade.jar

commons-logging-1.1.jar

AGENT-LIBRARY.jar

commons-codec-1.3.jar

build:

AGENT-LIBRARY: ant

WORKSHOP: ant

startup

study

run.xml

config/jade*.properties

start agents

commandline (don't)

config

Java code (more flexibility)

- create with classname
- instantiate first, accept in container

hands-on

solve TODO's in Main.java

agents

study

WorkshopAgent.java

AGENT-LIBRARY
javadoc (ant javadoc)

hands-on

solve TODO's in DoorAgent.java

behaviors

An agent has behavior

Behavior has a goal

goal reached? behavior stops

hands-on

solve TODO's in MovementBehavior.java

search

yellow pages (DF)

registrer 'services'

- name
- type
- language (codec, fipa-sl)
- ontology (datamodel)

hands-on

solve TODO 1 in SmileyAgent.java

study

CNSHelper.java
(AGENT-LIBRARY)

CNSAgentInitializer

find*

ontologies

datamodel

things: Concept

statements: Predicate

actions: AgentAction

validations: Facet (regex, cardinality,...)

Object ↔ String

study

CNSMessageHandler

- handle*

CNSReceiveMessageBehavior

- getMessageTemplates

communication package

- Vocabulary
- Ontology
- rest

hands-on

solve TODO 2 in SmileyAgent.java

messages

ACLMessage

performative: purpose of message (inform/ask/...)

sender/receiver(s)

content

    codec: syntax (fipa-sl)

    ontology: meaning

study

CNSHelper

    createMessage*

hands-on

solve TODO 3 in SmileyAgent.java

run

local

ant -f run.xml runall

distributed

pc1:	ant -f run.xml runPlatform
pc2:	put ip of pc1 in jademediator.properties
	ant -f run.xml runMediator
pc3 – pcn:	put ip of pc2 in jadesmileys.properties
	ant -f run.xml runSmilies

experiment

vary numsmiles and withinrange

make up extra behavior/intelligence