Multi-Agent Oriented Programming

– Practical Work –

The JaCaMo Platform

http://jacamo.sourceforge.net/

Olivier Boissier

ENS Mines Saint-Etienne
http://www.emse.fr/~boissier

Ecole Nationale
Supérieure des Mines
SAINT-ETIENNE

Web Intelligence Master — Oct 2010

Part of this work comes from the EASSS 2010 Multi-Agent Programming Course (O. Boissier, R. Bordini, M. Dastani, J.F. Hubner, A. Ricci)

Outline

Context

- Giacomo wants to build a house
- Two main phases are considered:
  1. Contracting specialised companies
     Giacomo hires various companies specialised in different aspects of house construction. The companies are independant and want to keep their autonomy.
  2. Building the house
     Contracted companies execute the main workflow for building the house under Giacomo's supervision.

Contracting specialised companies

- Tasks for which companies should be contracted:
  - (a) Site preparation
  - (b) Lay floors
  - (c) Build walls
  - (d) Build roof
  - (e) Fit windows
  - (f) Fit doors
  - (g) Install the plumbing
  - (h) Install the electrical system
  - (i) Paint the exterior of the house
  - (j) Paint the interior of the house
- The same company can be hired for more than one task

Building the house

- After the companies have been contracted, they have to execute their tasks on time and in coordination with each other
- Some tasks depend on others and some tasks can be done in parallel as represented by the workflow (";" for sequence and "|" for parallel).

$$a; b; c; (d|e|f); (g|h|i); j$$

Objective

- The objective of this practical work is to use the different abstract dimensions participating in the definition of Multi-Agent Oriented Programming to program a Multi-Agent System that could support the execution of this scenario.
- Given the constrainted framework of a course, this complex application is simplified while keeping the following features:
  - companies and Giacomo delegated part of their decisions to agents: one or several agents for Giacomo, one or several agents for each company
  - agents attached to Giacomo or Company are autonomous: no central control is possible
  - open system (companies may leave or enter the system at any moment) and dynamic environment

1. Scenario
2. Multi-Agent Oriented Modelling
   - Contracting Phase
   - Building Phase
3. Multi-Agent Oriented Programming

Agents

- House Owner Agent (Giacomo's assistant):
  - provides the requirements for the house, with budget limitations
  - helps Giacomo in the different decisions: contracting companies, managing and supervising the workflow execution
- Company Agent:
  - offer the house building services, skills and resources of the company
  - if hired, execute the house building tasks

Modelling Contracting Phase

- Interaction Modelling: since the different companies aim at keeping their autonomy, choice of a negotiation protocol in order to build contracts between Giacomo and the different companies for each of the tasks
- Organisation Modelling: flat organisation is defined with no particular role
- Environment Modelling: no particular environment concerns are considered in this phase

Interactions

- Use of Electronic Auctions to hire the required companies
- One auction for each task
- Each auction is started with:
  - the task description
  - the maximum value the owner can pay for it
- By the end of an auction, the company to be hired for that task is determined

Modelling Building Phase

- Interaction Modelling: At the end of the contracting phase, Giacomo contact the hired companies to enter into the execution phase and enact the negotiated workflow
- Organisation Modelling: a virtual organisation is created to assist with coordination and cooperation in the execution of the global workflow
- Environment Modelling: Graphical Interface to show the global state of the building

Organisation

- $\mathcal{M}$OISE functional specification is used to define the workflow
- $\mathcal{M}$OISE structural specification is used to define the role and group structures
- $\mathcal{M}$OISE normative specification is used to distribute the tasks of the workflow to the roles

Functional Specification

- The functional specification simply defines a social scheme for the global workflow
- $a; b; c; (d|e|f); (g|h|i); j$
- One mission for each task except for the painting of the exterior and of the interior of the house that are grouped into the same mission
- A task for the management of the execution of the workflow is also added

Structural Specification

- Roles: `house_owner`, `building_company` (abstract role), `site_prep_contractor`, `bricklayer`, `roofer`, `window_fitter`, `door_fitter`, `plumber`, `electrician`, `painter`
- Inheritance Hierarchy: `building_company` abstract role, specialised into: `site_prep_contractor`, `bricklayer`, `roofer`, `window_fitter`, `door_fitter`, `plumber`, `electrician`, `painter`
- Group: roles are used in a `house_group` group

Structural Specification

In group house_group,

- cardinalities are:
  - (1,1) for house_owner, site_prep_contractor, roofer, window_fitter, door_fitter, plumber, electrician, painter
  - (1,2) for bricklayer ;
- the same agent can play more subroles $\rightsquigarrow$ the role building_company is compatible with building_company
- role house_owner has authority over the building_company role
- a communication link connects the role build_company to house_owner

Normative Specification

- any agent playing the role `house_owner` is obliged to commit to mission `mManagement`
- role `site_prep_contractor` to the mission concerning the site preparation goal
- role `bricklayer` to the mission of laying the floors
- role `bricklayer` is also obliged to commit to the mission of building the walls
- role `roofer` is obliged to the mission of building the roof
- role `window_fitter` is obliged to the mission related to fitting the windows
- role `door_fitter` is obliged to commit to the mission of fitting the doors
- role `plumber` is obliged to installing the plumbing
- role `electrician` is obliged to installing the electrical system
- role `painter` is obliged to the mission concerning the painting of the house

1. Scenario
2. Multi-Agent Oriented Modelling
3. Multi-Agent Oriented Programming
   - Contracting Phase
   - Building Phase

Programming Agents

- The House owner agent is programmed in *Jason*
- The Company Agents are programmed in *Jason* (and 2APL)
  $\rightsquigarrow$ Heterogeneous Agent System

Programming Interaction in the Environment

- Encapsulation of the auction mechanism in an auction artifact
- Giacomo creates instances of such artifacts for creating/managing the various auctions; one such auction is used for hiring companies for each of the house building tasks
- Companies can perceive those artifacts and bid according to their competence and following their own strategies
- After some time Giacomo decides to finish the auction, observing the current best bid shown on the artifact

Auction Artifact

- observable properties:
  - task description
  - maximum payment value
  - current best bid (lower service price)
  - current winning agent ID
- operation:
  - bid(p): places a new bid for doing the service for price p (used by company agents to bid in a given auction)

Programming Interaction in the Agent Side

- Giacomo Agent:
  - Plans to launch all auctions by creating the corresponding auction artifacts
  - After some time looks at the best bid in each auction artifact and awards a contract for the winning company
- Company Agents;
  - Plans to look for the auction artifacts of their interest
  - Plans defining their own bidding strategy

Programming Agents for the Building Phase

- Giacomo Agent is enriched with:
  - Plan to send messages to the hired companies to enter into the execution phase, after all auctions are over.
  - Plan to build the virtual organisation based on the result of the contracting phase: allocating the different roles to the contracted company agents
- Company Agent:
  - Plans to enter the organisation, adopt the role corresponding to their contract and to catch the different events generate by the Organisation
  - Plans to execute autonomously the various actions related to the goals related to the missions they are committed to in the organisation scheme
  - NB: agents are benevolent with respect to the organisation, i.e. they don't violate the norms

Programming Organisation

- Moise functional specification is used to define the workflow
- Moise structural specification is used to define the role and group structures
- Moise normative specification is used to distribute the tasks of the workflow to the roles

Programming Environment

- Artifacts that model the state of the environment (e.g., model the state of the construction of a wall)