

Practical Session

Multi-Agent Programming Tutorial @ EASSS 2010

Olivier Boissier	(EMSE, France)
Rafael H. Bordini	(INF-UFRGS, Brazil)
Mehdi Dastani	(Utrecht University, Netherlands)
Jomi F. Hübner	(DAS-UFSC, Brazil)
Alessandro Ricci	(Bologna University, Italy)

Scenario (I)

- Giacomo wants to build a house
- We consider two main phases:
 1. Contracting specialised companies
Giacomo **hires** various companies specialised in different aspects of construction
 2. Building the house
Contractors **execute** the main workflow for building the house under Giacomo's supervision

Scenario (2)

- Phase I: Contracting specialised companies
- The objective here is to hire one company for each of these tasks:
 - (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

NB: The same company can be hired for more than 1 task

Scenario (3)

- Phase 2: Building the house
 - After the companies have been hired, 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

Scenario (3)

- Phase 2: Building the house

- After the companies have been hired, they have to execute their tasks in a sequence and in coordination with each other.
- Some tasks depend on other tasks. Some tasks can be done in parallel. The workflow is represented by the workflow diagram below, where "—" for sequence, "&" for parallel sequence and "|" for parallel tasks.

- (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

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

A MAS for House Building

- To make the scenario suitable for a course like this, we introduced some simplifications in the system design
- We summarise the solution using our approach, commenting on:
 - Agents
 - Contracting Phase (using Environment)
 - Building Phase (using Organisation)

Agents (I)

- Two types of agents:
 - **House owner** (Giacomo): provides the requirements for the house, with budget limitations
 - **Companies**: they will offer their service and, if hired, will execute the house building tasks; they are characterised by their competences in house building

Agents (2)

- Implementation
 - The house owner is programmed in Jason
 - Companies are programmed in Jason (and 2APL)
 - A heterogeneous agent system

Contracting Phase (I)

- Electronic auctions will be used 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

Contracting Phase (2)

- Implementation choices:
 - An **auction artifact** encapsulates the auction mechanism
 - 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

Contracting: Environment Side

- An auction artifact has the following
 - observable properties:
 - task description
 - maximum payment value
 - the current best bid (lower service price)
 - the 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)

Contracting: Agent Side

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

Building Phase

- After all auctions are over, Giacomo sends messages for the hired companies to enter into the execution phase
- A virtual organisation is created to assist with coordination and cooperation in the execution of the global workflow
- Implementation choice:
 - The organisation is specified using the Moise organisation modelling language

Exercises

1. Change the code of the auction artifact to:
 - (a) create a new observable property that shows the state of the auction (open or closed)
 - (b) add a new operation `clearAuction` (after `clearAuction`, the state of the auction becomes closed and attempts to use the `bid` operation will fail)
2. Change the house owner program so that the agent uses the new `clearAuction` operation

Exercises

2. Create a new company for one of the tasks and give it any bidding strategy you like
3. Change the auction artifact so that it shows and manages the bidding deadline and has a new operation for starting the auction; the `clearAuction` operation is no longer needed

Exercises

4. Choose a new auction mechanism and implement a new auction artifact that implements that mechanism; you should not change the agents for this exercise
5. Now choose another mechanism that will require different strategies in the agents and implement them

Downloads

- Jason

<http://jason.sourceforge.net>

- Cartago

<http://cartago.sourceforge.net>

- Initial project

<http://sourceforge.net/projects/jacamo/files>

Moise Specification

Building: Organisation Side

- 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

Functional Specification

- The functional specification simply defines a social scheme for the global workflow

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

- 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

Functional Specification

- The functional specification simplifies the social scheme for the global workflow

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

- One **mission** for each task except painting of the exterior and of the interior of the house that are grouped into a mission
- A task for the management of the workflow is also added

- (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

Structural Specification (I)

- Role hierarchy
 - `house_owner`
 - `building_company`
 - abstract role, specialised into:
`site_prep_contractor`,
`bricklayer`, `roofer`,
`window_fitter`, `door_fitter`,
`plumber`, `electrician`, `painter`

Structural Specification (2)

- The roles are used in a **group** called 'house_group' where:
 - house_owner has cardinality (1,1)
 - site_prep_contractor has cardinality (1,1)
 - bricklayer has cardinality (1,2)
 - roofer has cardinality (1,1)
 - window_fitter has cardinality (1,1)
 - door_fitter has cardinality (1,1)
 - plumber has cardinality (1,1)
 - electrician has cardinality (1,1)
 - painter has cardinality (1,1)

Structural Specification (3)

- Notes:
 - the role `building_company` is compatible with `building_company` so that the same agent can play more subroles
 - role `house_owner` has authority over the `building_company` role
 - a communication link connects the role `build_company` to `house_owner`

Normative Specification (I)

- Given the definition of the **missions**, the following **norms** are defined:
 - any agent playing the **role** `houseOwner` 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
 - `bricklayer` is also obliged to commit to the mission of building the walls
 - `roofer` is obliged to the mission of building the roof

Normative Specification (2)

- `role window_fitter` to the mission related to fitting the windows
- `door_fitter` is obliged to commit to the mission of fitting the doors
- `plumber` to installing the plumbing
- `electrician` to installing the electrical system
- `painter` to the mission concerning the painting of the house

Building: Agent Side

- Owner agent is equipped with the plans to construct the virtual organisation based on the result of contracting phase
- Company agents have plans to enter the organisation, adopt the role corresponding to their contract and to catch the different events generated by the OMI
- Companies have 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

Building: Environment Side

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

Exercises

- I. Do the following changes in the organisation specification:
 - (a) tasks `site_preparation` and `lay_floors` can be done in parallel
 - (b) all tasks have to be done in sequence

Exercises

2. Develop an agent that tries to adopt roles related to tasks he is not supposed to (malevolent agent!)

(e.g. Giacomo trying to play some company role)

3. Develop an agent that does not fulfill the tasks

Exercises

4. Change the Giacomo agent so that it reacts to the norm violation
 - Giacomo should create a new auction for that task and forbid the violating company from taking part in the new auction
5. Change the system to build two houses in parallel

Exercise

6. Change the Giacomo agent so that it reads the Moise specification and creates the necessary auction artifacts based on the specified tasks
7. Change Giacomo so that it is able to monitor the building of the house and check whether the tasks are being done appropriately

Downloads

- Jason

<http://jason.sourceforge.net>

- Cartago

<http://cartago.sourceforge.net>

- Moise

<http://moise.sourceforge.net>

- Initial project

<http://www.emse.fr/~boissier/map-tut/>

Conclusions

- Separation of concerns
 - agent, environment, organisation, interaction
 - $MAP = AOP + EOP + OOP + IOP$
- Design == Implementation
- In our code, (almost) no (direct) communication in the MAS!