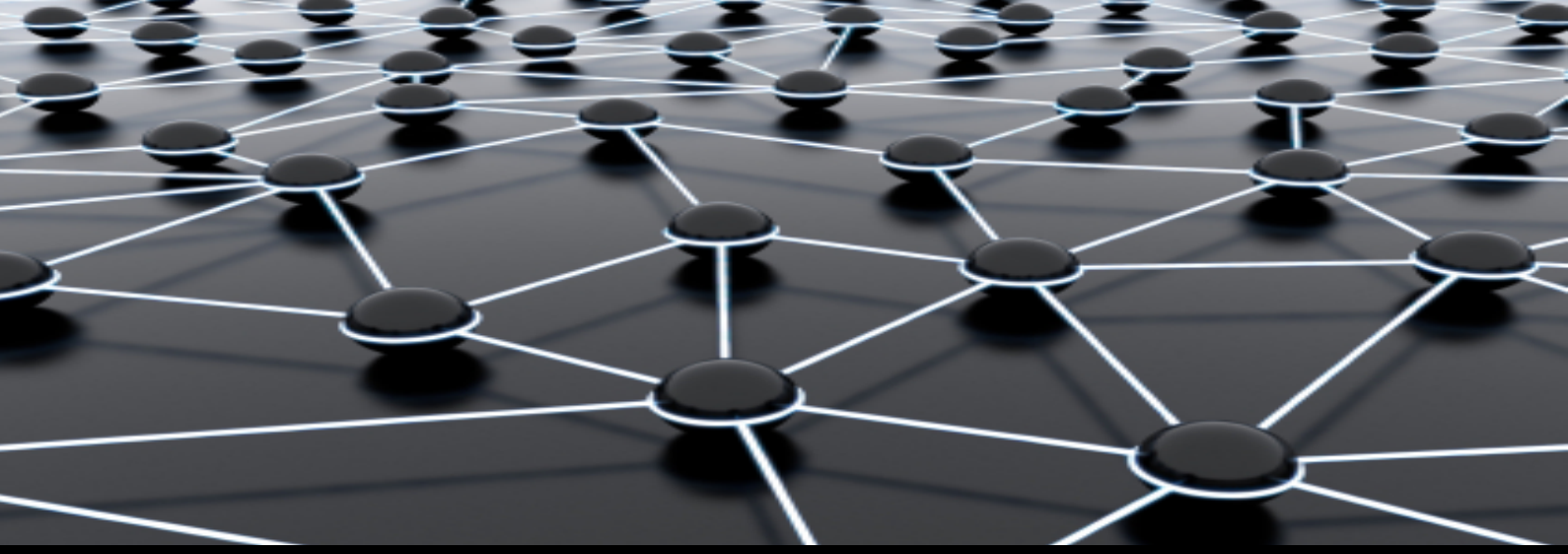

UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

Introduction to SARL

Lab Works #1

STÉPHANE GALLAND

Ref.: UTBM-INFO-IA54-LW1
Ver.: 1.0
Date: 2016/01/20
Status: Public

Laboratoire Systèmes et Transport
Multiagent Group
Université Bourgogne Franche-Comté
Université de Technologie de Belfort-Montbéliard
13, rue Ernest Thierry-Mieg
90010 Belfort cedex, France
http://www.multiagent.fr

1/ GOAL OF THIS LAB WORK SESSION

The goal of this lab work session is to introduce the basics concepts of agent-oriented programming with SARL.

You shall learn:

- How to write the agent.
- How to call an agent capacity.
- How to spawn an agent from an agent.
- How to define new capacities and skills.
- How to exchange information between agents.

2/ FIRST STEP: PREPARE THE DEVELOPMENT ENVIRONMENT

In this section, you will find the tasks to do for preparing your development environment.

2.1/ INSTALLATION OF THE ECLIPSE TOOLS

Recommended version of SARL : 0.4.3 (stable)

1. Download the **Eclipse product** that contains the compilation tools for the SARL programming language : http://www.sarl.io
2. Uncompress the Eclipse product for SARL.
3. Launch the downloaded Eclipse product.
4. Open the wizard for creating a SARL project, with the menu:
   `> File > New > Project > SARL > Project`
5. Enter the name of the project.
6. Click on ”Finish”, the SARL project should be created.
7. You must ensure that the configuration of your SARL project is correct (may be still a bug in the SARL environment):
   a) Open the dialog of the properties of the SARL project by clicking on:
      `Right click on project > Properties > SARL > Compiler > Output Folder`
   b) Check if the field “Directory” is set to a source folder that is existing in your SARL project. If not change the property with):
      `src/main/generated-sources/sarl`

Your Eclipse environment is now ready for the lab work session.

2.2/ SARL Documentation

The documentation for the SARL syntax, and the provided elements is available at:
http://www.sarl.io/docs/suite/io/sarl/docs/SARLDocumentationSuite.html

2.3/ Bug Report and Questions

In case you have discovered an issue in the SARL tools, you could submit a description of it to the SARL development team via the Github interface:
https://github.com/sarl/sarl

In case you cannot discuss with your teacher, you could use the SARL forums to the SARL development team via the Google Group interface:
https://groups.google.com/forum/#!forum/sarl

3/ Work to be Done during the Lab Work Session

The following sections describe the work to be done during this lab work session.

3.1/ First Agent

First, you should write an agent that is able to display “Hello World” on the output console.

You should:

a) create the agent type with the name `Agent1` (with the wizard, or by hand).

b) create the event handler that is run when the `Initialize` event is received by the agent. This event is fired by the platform when the agent should initializee itself.

c) write the output statement in the event handler.

For running the agent, you should:

a) Open the dialog box of the “Run Configurations”.

b) On the left side, click on “SARL Application”.

c) Click on the “New launch configuration” button.

d) On the right side, select the project and the agent to launch.

e) Click on “Run”.

It is recommended to put a breakpoint in the event handler, and run the agent in debug mode.

3.2/ USE THE LOGGING BUILDIN CAPACITY

The goal of this exercise is to use a capacity that is provided by the run-time platform, aka. buildin capacity.

You should:

a) update the agent code for using the “Logging” capacity.

b) May anything else be changed in the code?

3.3/ SPAWNING ANOTHER AGENT

Most of the time, a system is composed by more than one agent. This exercise will enable you to launch another agent that is displaying “Welcome” on the output console also.

You should:

a) create a second agent `Agent2`, with its `Initialize` event handler that is displaying the welcome message.

b) Update the code of `Agent1` for using the `DefaultContextInteractions` buildin capacity. This capacity permits to do something with the context in which the agent is living, aka. the default context.

c) Update the `Initialize` event handler of `Agent1` for spawning an agent of type `Agent2`.

d) create a third agent `Agent3`, with its `Initialize` event handler that is displaying the welcome message.

e) Update the `Initialize` event handler of `Agent1` for spawning an agent of type `Agent3`.

3.4/ SAY HELLO TO EVERY ONES

Agents are social entities. They are supposed to interact with other agents. This exercice enables the `Agent1` to say hello to the other agents.

In SARL, the information exchanged between agents are carried out by events. The events are put inside an interaction space of a context. In this exercise, the default space of the default context will be used. It is automatically accessible when using the `DefaultContextInteractions` capacity.

You should:

a) create the definition of the event `Hello`.

b) update the code of `Agent1` for sending the event to every one after it has spawned the other agents.

c) update the code of `Agent2` and `Agent3` for displaying the welcome message when they are receiving the hello event.

3.5/ Say Hello to a Single Agent

Broadcasting events may not be the interaction mode between two agents. In this exercise, the `Hello` message will be sent only to `Agent3`.

You should:

a) update the code of `Agent1` for emiting the event with a scope retricted to `Agent3`.

3.6/ Say Localized Welcome

Agents may have different ways for doing a specific task. In this exercise, the agents will be enable to say ”Hello” according to their own skills, i.e. their languages.

The concepts of Capacity and Skill are suitable for this task. A capacity is the definition of functions (similar to an interface in object-oriented programming) that could be invoked by the agents. A capacity never defines the code of a function, only its prototype. A skill is a specific implementation of a capacity (similar to object implementing an interface in object-oriented programming). The skill must provide a code for each function of the implemented capacity.

You should:

a) define the `SayHello` capacity.

b) define the `SayHelloSkill` skill, that permits to say hello in English.

c) define the `DireBonjourSkill` skill, that permits to say hello in French.

d) update the code of the agents, that are defined previously, for using the `SayHello` capacity.

e) update the code of the agents for calling the function provided by the `SayHello` capacity.

f) update `Initialize` event handler of `Agent2` for defining its skill that is corresponding to the `SayHello` capacity. It should speak English.

g) update `Initialize` event handler of `Agent3` for defining its skill that is corresponding to the `SayHello` capacity. It should speak French.

3.7/ Contract Net Protocol (optional)

In this excercise, you must define a multiagent system that is running a Contract Net Protocol:

1. A customer wants to by a product.
2. The customer sends a request to a broker for finder the best provider.
3. The broker asks to a couple of provider they offers for the product.

4. The broker selects the best offer.
5. The broker notifies the selected provider about its acceptance with the ID of the customer.
6. The broker notifies the not selected providers about their rejections.
7. The broker notifies the customer with the ID of the provider.

You should:

a) Define the capacity of the provider.

b) Define the capacities of the broker.

c) Write the provider code.

d) Write the broker code.

e) Write the customer code.

Stéphane Galland

Introduction to SARL
Lab Works #1

Ref.: UTBM-INFO-IA54-LW1
Ver.: 1.0
Date: 2016/01/20
Status: Public

Laboratoire Systèmes et Transport
Multiagent Group
Université Bourgogne Franche-Comté
Université de Technologie de Belfort-Montbéliard
13, rue Ernest Thierry-Mieg
90010 Belfort cedex, France

Contact
Stéphane Galland, Ph.D.
✆ +33 384 583 418
stephane.galland@utbm.fr
http://www.multiagent.fr