Introduction to Multi-Agent Systems
Implementation - First Delivery
Team 5

Victor Gimenez Abalos - victor.gimenez.abalos@est.fib.upc.edu
Daniel Felipe Ordonez Apraez - daniel.ordonez@est.fib.upc.edu
Albert Rial Farràs - albert.rial@est.fib.upc.edu
Jordi Armengol Estapé - jordi.armengol.estape@est.fib.upc.edu
Joan Llop Palao - joan.llop@est.fib.upc.edu

January 15, 2020

1 Introduction

In this document we will explain the tasks we have done for this first implementation of our system, how we have divided these tasks and the meetings we have done and what we have decided in each one. Recall that the source code is attached to this delivery.

2 Tasks done

2.1 User agent

Our current implementation of the user agent correctly handles:

- Setup: In the setup the user agent properly sets up its registry in the JADE Directory Facilitator (DF), which can be thought as the "Yellow Pages" of JADE. It also instantiates the Manager Agent.
- General behaviour: The user agent implements a *cyclic behaviour* such that:
  - It reads the user input:
    - Training: `T <config file name>`
    - Predicting: `P`
  - It reads the configuration file.
  - It sends requests with instructions to the Manager Agent.

2.2 Manager agent

Our current implementation of the agent can both dynamically instantiate classifier agents and receive and send orders from the user agent and to te classifiers, respectively.

2.3 Classifier agents

Our classifiers are actually working in this release. They can be trained when required. Specifically, we have implemented a J48-type classifier. In addition, we have enabled a communication protocol for both training and predicting.

3 Future work

3.1 User agent

As future work for the user agent, we are planning to further develop the control of the agent's own life cycle (inactive, active, down). In addition, we have to complete the FIPFA Request Protocol by being able to receive "refused" or "agreed" responses from the Manager and receiving failures and informs as well.

3.2 Manager agent

We are planning to establish more complex strategies for managing the classifiers. For instance, dynamically giving more weight to the classifiers in which we have more confidence.

3.3 Classifier agents

As future work, we suggest creating classifier variants apart from J48.

4 System execution

In this sections we detail the instructions for executing our system in Windows and Linux.

4.1 Instructions for executing our system in Windows:

1. Compile the agents:

   ```
   javac -classpath lib\jade.jar;lib\weka.jar -d src\out\production\IMAS_test\ src\src\*.java
   ```

2. Execute Jade with User Agent (the rest will be created dynamically):

   ```
   java -cp lib\jade.jar;lib\weka.jar;src\out\production\IMAS_test\ jade.Boot -gui -agents
   user:UserAgent
   ```

3. Use the command line to train or predict:

   ```
   USAGE: T <config_file> | P
   Example: T imas.settings
   ```

4.2 Instructions for executing our system in Linux:

1. Compile the agents:

   ```
   javac -cp lib/jade.jar:lib/weka.jar -d src/out/production/IMAS_test src/src/*.java
   ```

2. Execute Jade with User Agent (the rest will be created dynamically):

   ```
   java -cp lib/jade.jar:lib/weka.jar:src/out/production/IMAS_test/ jade.Boot -gui -agents
   user:UserAgent
   ```

3. Use the command line to train or predict:

   ```
   USAGE: T <config_file> | P
   Example: T imas.settings
   ```

5 E-Portfolio

Our E-Portfolio tool is Github. Our repository is hosted at: `https://github.com/jordiae/imas-mai`. We have leveraged Github Issues for managing this project. The repository history with all the commits done on the project can be found also in Github, at `https://github.com/jordiae/IMAS-MAI/commits/master`.

5.1 Meetings

For the meetings we use the Github Wiki, to track them and to write the topics discussed and the decisions made. In this first implementation we have done three meetings. Below we describe its information.

5.1.1 Meeting 1

- Date: 01/11/19
- Duration: 1 hour
- Assistants: All team members
- Topics:

  - Implementation general ideas
  - Communication framework draft
  - Work splitting

- Decisions:
  - Making a wiki page in our Github for the communication protocols of the agents, to be expanded as we require more.
  - Splitting the work into subtasks: mostly each type of agent. We will pool them and distribute them as required.

5.1.2 Meeting 2

- Date: 08/11/19
- Duration: 3 hours
- Assistants: All team members
- Summary: After individually studying the task assigned to each of us, we shared knowledge and went deep on our discussions.
- Topics:
  - Communication protocol between Agents
  - Discussion about the dynamical creation of Agents
  - Discussion about the proper way of reading and sending the configuration file properties
- Decisions:
  - For the communication protocol we decided to use the FIPA Request Interaction Protocol.
  - For reading and sending the configuration file, we will use an XML Parser and construct manually a string with the parameters, divided by some kind of character like ”@”. We decided to study on the future a way to send a Serializable object instead of doing it manually, but it is not our priority.
  - We left also the dynamical creation of Agents for later. First, we will instantiate them via command line.

5.1.3 Meeting 3

- Date: 15/11/19
- Duration: 2 hours
- Assistants: All team members
- Summary: We reviewed the code that we had and commented our thoughts about what we should improve.
- Topics:
  - Classifier Agents future work
  - Discussion about the dynamical creation of Agents
  - Discussion about what was left for the first implementation
- Decisions:
  - We decided that the Classifier Agent that we had was good enough for this first implementation.
  - The dynamical creation of Classifiers and Manager should be done for the first implementation. User should instantiate the Manager and this one the Classifiers.
  - We decided that the Manager should send the training request to Classifiers. However, we decided that the prediction was not going to be part of this first approach.

5.2 Tasks assigned to each member

Managing a group of 5 members is a non-trivial task. For this reason, apart from general tasks that were assigned to the whole group (eg. writing this document, preparing the slides and the presentation...), we created three sub-groups. Each one of them was in charge of the development of one agent. Specifically:

- Víctor developed the classifier agent.
- Daniel and Albert developed the user agent.
- Joan and Jordi developed the manager agent.

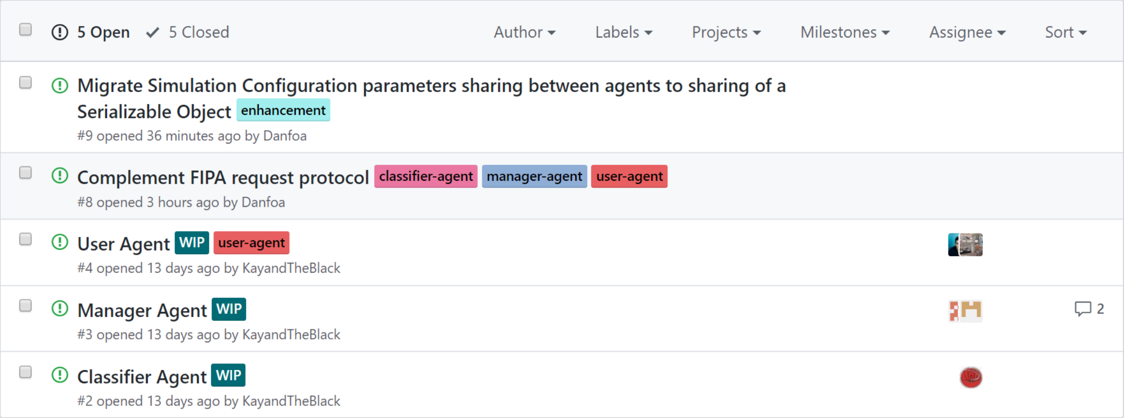
However, it is very important to notice the following considerations:

- Even though the tasks done can be reviewed in the commit history, it is worth noting that when working in pairs, it could happen that one member of the sub-group did not appear in the commits (eg. in cases in which the sub-group met physically and the commits were done by one of the members of the sub-group).
- At the end of the day, since the agents implementation was not completely independent, everyone ended up a bit of other agents, apart from the assigned one.

For holding the tasks we use also the repository in Github. The repository holds organizational information for each of the agent types considered, and for all of the features, bugs and enhancements during the process of design and development.

Figures 1 and 2 depict the open and closed issues in the repository, indicating abstractly which features were already addressed and which are still missing.

Finally, Figures 3, 4 and 5 show the status of development of each of the classifier types considered.



□ Migrate Simulation Configuration parameters sharing between agents to sharing of a Serializable Object enhancement
#9 opened 36 minutes ago by Danfoa

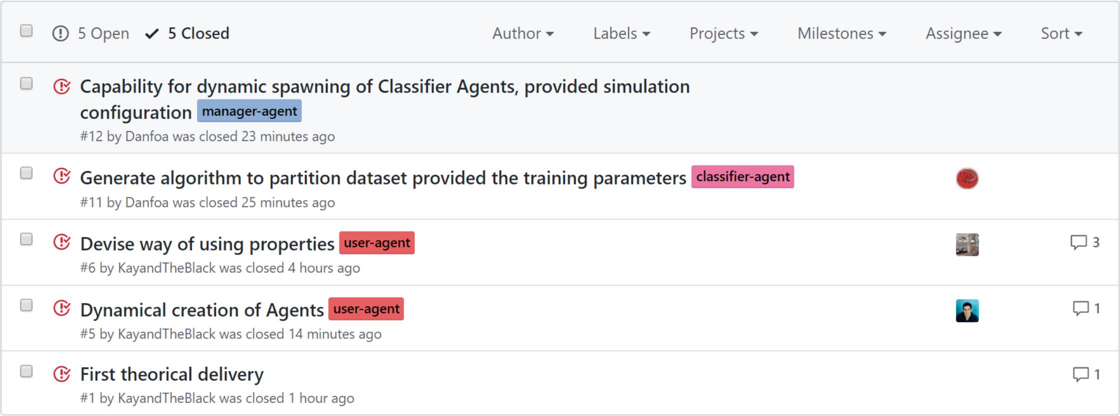
□ Complement FIPA request protocol classifier-agent manager-agent user-agent
#8 opened 3 hours ago by Danfoa

□ User Agent WIP user-agent
#4 opened 13 days ago by KayandTheBlack

□ Manager Agent WIP 2
#3 opened 13 days ago by KayandTheBlack

□ Classifier Agent WIP
#2 opened 13 days ago by KayandTheBlack

Figure 1: Open issues in project repository



Capability for dynamic spawning of Classifier Agents, provided simulation configuration manager-agent
#12 by Danfoa was closed 23 minutes ago

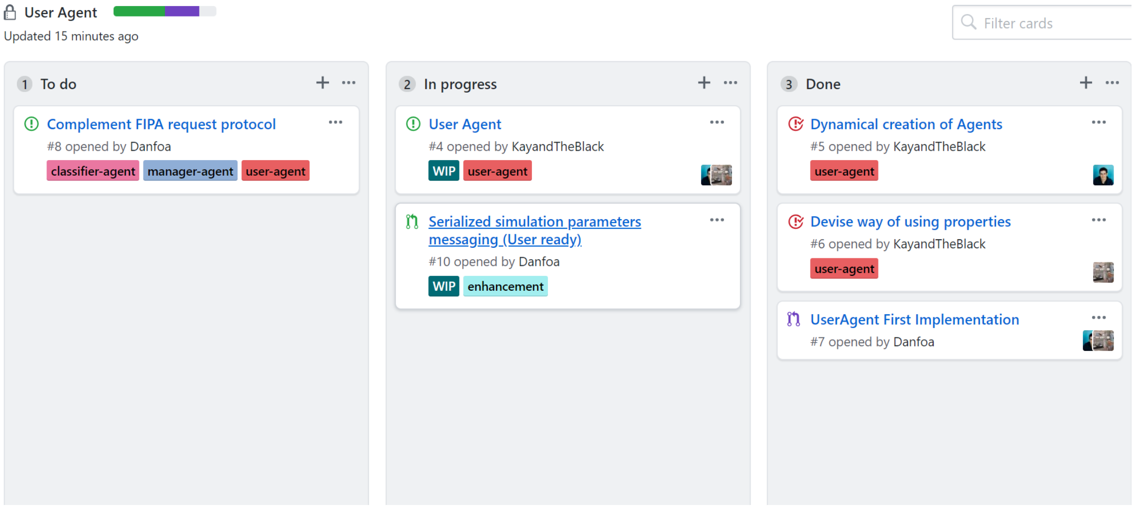
Generate algorithm to partition dataset provided the training parameters classifier-agent
#11 by Danfoa was closed 25 minutes ago

Devise way of using properties user-agent 3
#6 by KayandTheBlack was closed 4 hours ago

Dynamical creation of Agents user-agent 1
#5 by KayandTheBlack was closed 14 minutes ago

First theorical delivery 1
#1 by KayandTheBlack was closed 1 hour ago

Figure 2: Closed issues in project repository



Complement FIPA request protocol
#8 opened by Danfoa
classifier-agent manager-agent user-agent

User Agent
#4 opened by KayandTheBlack
WIP user-agent

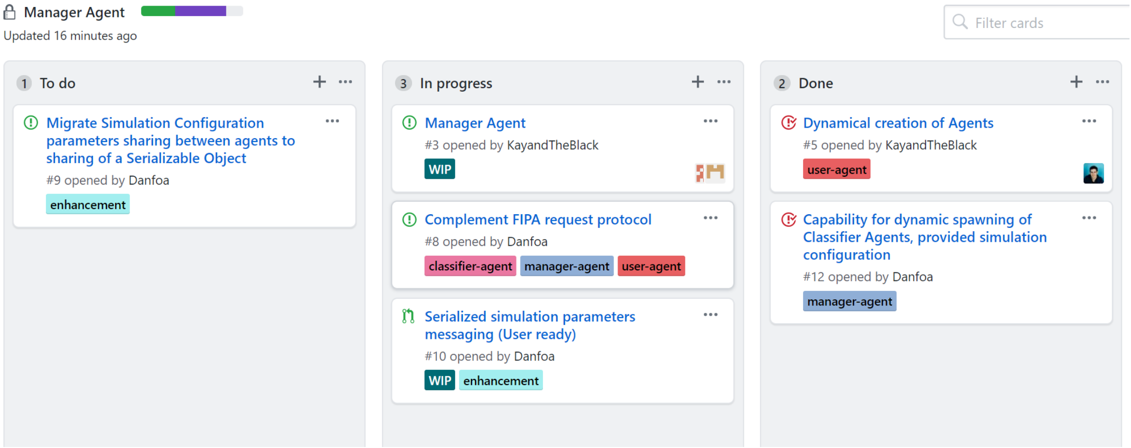
Serialized simulation parameters messaging (User ready)
#10 opened by Danfoa
WIP enhancement

Dynamical creation of Agents
#5 opened by KayandTheBlack
user-agent

Devise way of using properties
#6 opened by KayandTheBlack
user-agent

UserAgent First Implementation
#7 opened by Danfoa

Figure 3: User agent sub-project status



Migrate Simulation Configuration parameters sharing between agents to sharing of a Serializable Object
#9 opened by Danfoa
enhancement

Manager Agent
#3 opened by KayandTheBlack
WIP

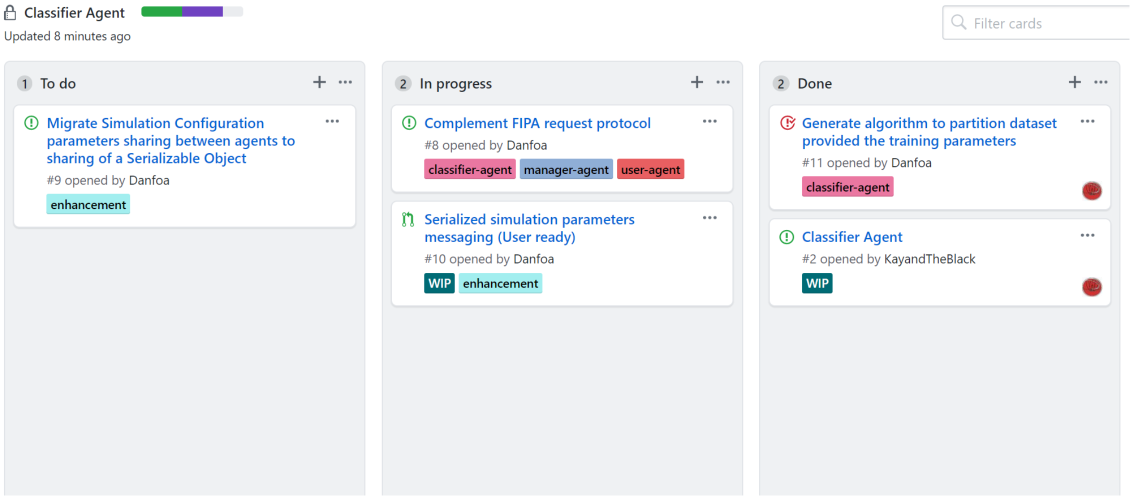
Complement FIPA request protocol
#8 opened by Danfoa
classifier-agent manager-agent user-agent

Serialized simulation parameters messaging (User ready)
#10 opened by Danfoa
WIP enhancement

Dynamical creation of Agents
#5 opened by KayandTheBlack
user-agent

Capability for dynamic spawning of Classifier Agents, provided simulation configuration
#12 opened by Danfoa
manager-agent

Figure 4: Manager agent sub-project status



Classifier Agent

Updated 8 minutes ago

1 To do

- Migrate Simulation Configuration parameters sharing between agents to sharing of a Serializable Object
  #9 opened by Danfoa
  enhancement

2 In progress

- Complement FIPA request protocol
  #8 opened by Danfoa
  classifier-agent manager-agent user-agent
- Serialized simulation parameters messaging (User ready)
  #10 opened by Danfoa
  WIP enhancement

2 Done

- Generate algorithm to partition dataset provided the training parameters
  #11 opened by Danfoa
  classifier-agent
- Classifier Agent
  #2 opened by KayandTheBlack
  WIP

Figure 5: Classifier agent sub-project status