The Financial Market Simulator (FMS) was installed and configured according to the instructions available for the 0.1.9 version.

After the configuration steps were completed, the runalltests.py file was executed. This step was taken in order to ensure that all functionalities of the simulator are in place as expected. Unfortunately, after running for 2 hours, the test_csvdelimiter_option was still executing. This issue will be addressed in the near future. However, when used, the simulator compiled and executed without any errors or warnings.

A very important note obtained from the available documentation, concerns the daylength parameter that is specified in the configuration file of an experiment. This represents the number of times the engine gives an agent the chance to place an order. Depending on the market model, the order might trigger a transaction or not.

When writing the configuration file for an experiment, the attributes for an agent are as follows:

The attributes block starts with number, which gives the number of individual agents of the specified class.

Money and stocks describe the initial portfolio of the agent, in cash and stocks.

Number, money and stocks are compulsory attributes, regardless of the chosen agent class.

The last attribute, args, is a list of optional attributes, the meaning of which depends on the agent class. For theZeroIntelligenceTrader class, the attributes are the maximum price and quantity an agent is allowed to use in an order, i.e. these are limits for the random process generating the orders.

For this stage of the analysis several experiments were executed using the basic outline for an experiment with FMS. The configuration file included information about all parts of the FMS model: the world, the engines/markets and the agents.

Given the fact that a random agent class is used, the experiment lasts for as many turns as the daylength and the engine class gives speak to a randomly chosen agent each turn before stopping. More precisely, what happens is:

- the engine chooses an agent randomly among the number provided upon initialization
- as the agent class is purely random, the chosen agent puts a random order (either sell or buy, a price and a quantity)
- the order was collected in the related book (sellbook or buybook)
- the clearing method ran on the books, comparing the best limits and triggering transactions if necessary

After running a first series of experiments with the given agents and markets (results can be observed in the experiments PDF), it was concluded that the next step that needs to be taken is the implementation of different trading algorithms that would be able to run on this simulator.