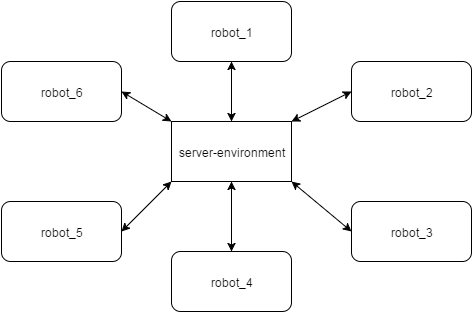
# Математико-компьютерная модель группировки спасательных роботов

Программа предназначена для моделирования поведения и группового взаимодействия спасательной группировки. «agent-robot» моделирует отдельного робота, «server-environment» - моделирует среду и каналы связи, библиотека utils описывает протокол взаимодействия между роботами, состояния роботов, команды.

Программное обеспечение является распределенным: каждый отдельный робот запускается на отдельной JAVA машине и является самостоятельной программой.



Роботы обмениваются сообщениями через сервер, что позволяет моделировать качество канала и моделировать помехи при обмене сообщениями между роботами.

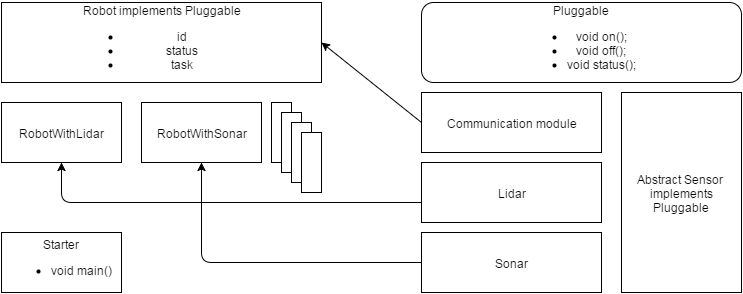
Пульт оператора реализован как Web приложение, и может быть запущен на любом устройстве с предустановленным браузером.

С пульта оператора можно получить информацию о любом роботе и группировке в целом.



## Структура модуля «agent-robot»

Робот представляет собой самостоятельную программу, запущенную на отдельной Java машине. Упрощенно схема программы робота представлена на рисунке.



Каждый робот реализует интерфейс Pluggable – содержащий методы on(), off(), status(). Эти методы являются базовыми для роботов и его внутренних устройств таких как сенсоры.

Каждый робот включает следующий обязательный набор полей:

* Int id;
* AgentStatus status;
* AgentTasks task.

Идентификатор робот однозначно идентифицирует его в системе. Статус робота позволяет определить его состояние: включен, выключен, потеря работоспособности, частично неисправен. Статусы состояния робота описаны в файле AgentStatus.java. AgentTask представляет собой задачу выполняемую роботом. Также у робота есть поле communicationModule – представляющее собой экземпляр класса модели средства коммуникации робота, через который он производит общение с остальной системой. Для реализации взаимодействия между роботами (подключения и отправки сообщений) была выбрана технология WebSocket.

От базового класса робота наследуются его более частные реализации, например, робот, оснащенный лазерным сканирующим дальномером. От базового класса робота он отличается наличием специального сенсора и специфическим для него поведением.

Все сенсоры робота наследуются от базового класса AbstractSensor реализующего интерфейс Pluggable. Соответственно любой из сенсоров робота может быт включен, выключен, неисправен и т.п., что позволяет реализовывать различные сценарии поведения робота при изменении его конфигурации.

Входными данными для работы программы является:

данные полученные от программных моделей роботов: строки в формате JSON оформленные в соответствии с протоколом сообщений,

данные полученные от пульта управления оператора: строки в формате JSON оформленные в соответствии с протоколом сообщений.

## Структура модуля «server-environment»

Основная задача серверной части модели – это моделирование среды окружающей робота, моделирование канала связи между роботами и передача сообщений между ними.

Так как процесс обмена сообщениями между роботами и сервером реализован на базе технологии WebSocket, все соединения между роботами и сервером осуществляются через Socket.

Сообщения от робота к серверу могут быть двух типов:

* информация о действиях робота в среде и общение сенсоров со средой;
* сообщения, которыми обмениваются между собой роботы.

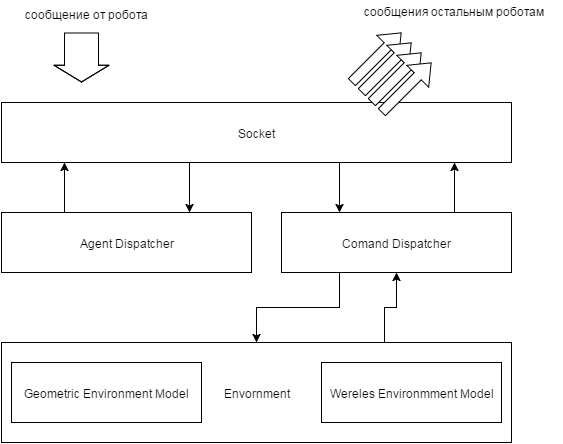


Рисунок - схема работы приложения server-environment.

Принципиальную важность играют сами сообщения, которыми обменивается робот и их формат.

В общем виде сообщение между роботами имеют следующий формат:

{

id:[integer],

command: [command] ,

data: {}

}

Данные передаются в формате JSON представляющий собой текстовый формат обмена данными, сообщение представляет собой набор полей и значений. Каждое сообщение должно включать в себя id робота который отправляет сообщение, команду, из списка команд группировки и специальное поле data куда включается вся вспомогательная информация.