

1. Arbiter пользуется данными Observer о физических (например, скорость) и геометрических (например, координаты) характеристиках роботов.
2. Arbiter пользуется данными Configuration: ограничения игры
   1. геометрические характеристики (например, размеры поля)
   2. состав игры
   3. временные ограничения

Состав команд:

* количество игроков
* id команды
* id игроков

1. Deploy пользуется данными Configuration: В Configuration содержится информация о количестве команд и их составе и здесь же содержатся данные об id роботов.
2. Observer в начале игры обращается к Configuration для получения информации о привязке роботов к их маркерам.
3. Logger записывает данные о действиях Arbiter: запуск и останов игры, выявленные нарушения, счёт.
4. Logger записывает данные, полученные от Observer, для записи игры: видео, координаты и т.п.
5. В Configuration определяется, какие именно данные должен записывать Logger.
6. Wrap отправляет данные в Logger во время простоя канала коммуникации.
7. Сервис Analytics извлекает данные из Logger для проведения необходимого анализа (временные зависимости, графы обращений робот->робот итд).
8. Arbiter может передать сообщение роботу или всей команде через Wrap:
   1. Начать игру
   2. Прекратить игру
   3. Нарушение

Для этого существует интерфейс, который должна реализовать программа любого робота.

StartGame() –начало игры/тайма

StopGame()–конец игры/тайма

Pause() – приостановка игры по техническим причинам.

OnPosition() – Встать на исходную позицию (начало игры/ пенальти)

QuitGame() – Покинуть игру (в случае окончания игры или нарушения отправиться в специально отведенное на карте место)

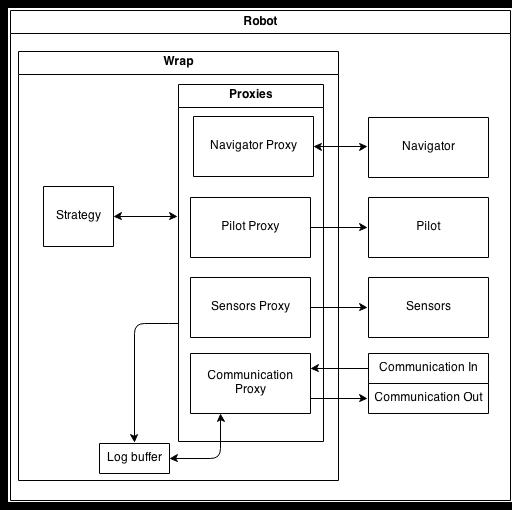
Caution(DateTime time, Level level) – Предупреждение о нарушении с количеством времени на исправление и с уровнем серьезности нарушения. В случае самых серьезных нарушений после этого сообщения вызывается QuitGame(), в случае несерьезных нарушений счетчик предупреждений увеличивается на 1.

1. Сервис Analytics извлекает данные из Configuration о типах данных, полученных из Wrapp и доступных для анализа.
2. Client может получить аналитические данные, в соответствии с п. 11
3. Client может отправлять сервису Deploy программы роботов. Deploy регистрирует программу помещает её в робота и регистрирует её (связывает её с номером робота).
4. Перед игрой клиенты договариваются друг с другом об ограничениях игры, и администратор вносит эти ограничения в Configuration.
5. Любой клиент может наблюдать за ходом игры, т.е. получать видеопоток.

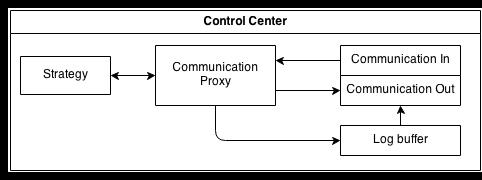
Wrap –оболочка для программы робота. Wrap фиксирует команды программы робота к другим роботам или датчикам, двигателям и т.п. и временную метку. Wrap копит эту информацию и отправляет периодически на сервер (во время простоя канала коммуникации). Wrap может перехватывать все обращения к lejos. Api и отправлять данные об этом в logger. Перед игрой можно настроить глубину с которой Wrap исследует команды программы робота.

Analytics- сервис, предоставляющий данные для анализа поведений роботов. Команды и зависимость их от веремени, графы обращений роботов друг к другу, отработанные роботами сценарии итд

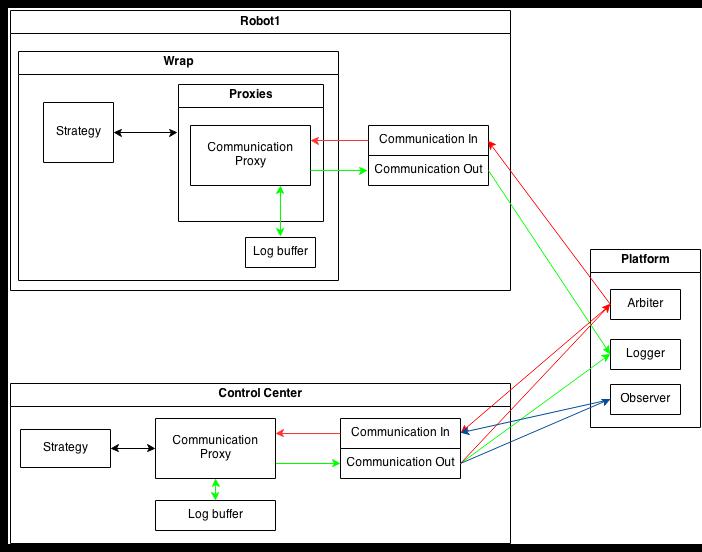
Robot



Control Center



Communications



Communication Protocol

Message

+ From: byte

+ To: byte

+ Category: bool

+ Prefix: byte

+ Content: byte[]

Id:

0x0 - broadcast

0x1 – 0x10 – teams

0x11-0x8C -robots. Robot id=teamNumber\*10+robotNumber

Category: 0 – сообщение от цу к роботу или между роботами.

1. Команда от арбитра обязательная к исполнению. Обязательно учитывается префикс команды

Prefix

|  |  |
| --- | --- |
| Start game | 0x1 |
| Stop Game | 0x2 |
| Pause | 0x3 |
| Move to position | 0x4 |
| PenaltyLevel0 | 0x5 |
| PenaltyLevel1 | 0x6 |
| PenaltyLevel2 | 0x7 |
|  |  |
| Game ended | 0x65 |
| Game server failure | 0x66 |
| Game server failure | 0x67 |
|  | 0x68 |
|  | 0x68 |