

Разработка ядра мультиагентной системы

Пономарев Олег 9 “А” класс

Научный руководитель:
Дединский Илья Рудольфович

Цель

Реализация программного комплекса, облегчающего создание мультиагентных систем.

Задачи

- 1) Создание системного загрузчика, позволяющего динамически подгружать и выгружать модули из системы.
- 2) Создание компонента, позволяющего загруженным модулям взаимодействовать между собой.
- 3) Создание динамической базы данных, в которой модули могут хранить разделяемые данные.
- 4) Создание системы отладки как модуля в отдельности, так и системы в целом.

Мультиагентные системы

Мультиагентная система (МАС) –

это система, образованная несколькими взаимодействующими интеллектуальными агентами.

Основные характеристики агентов в МАС:

- **Автономность:** агенты полностью или частично независимы друг от друга.
- **Ограниченность представления:** ни у одного из агентов нет представления о всей системе, или система слишком сложна, чтобы знание о ней имело практическое применение для агента.
- **Децентрализация:** нет агентов, управляющих всей системой.

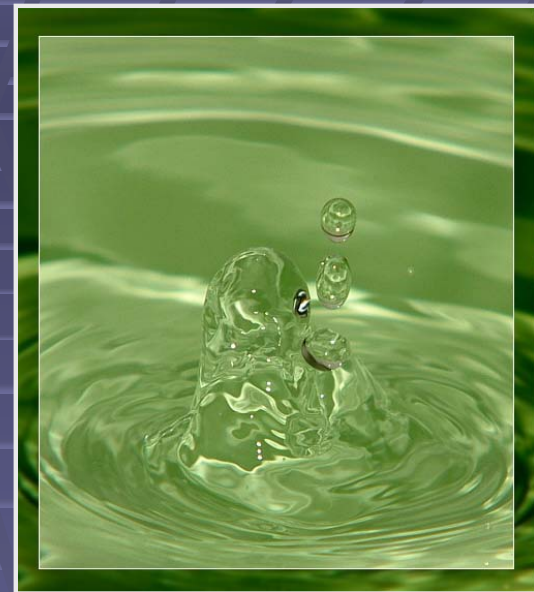
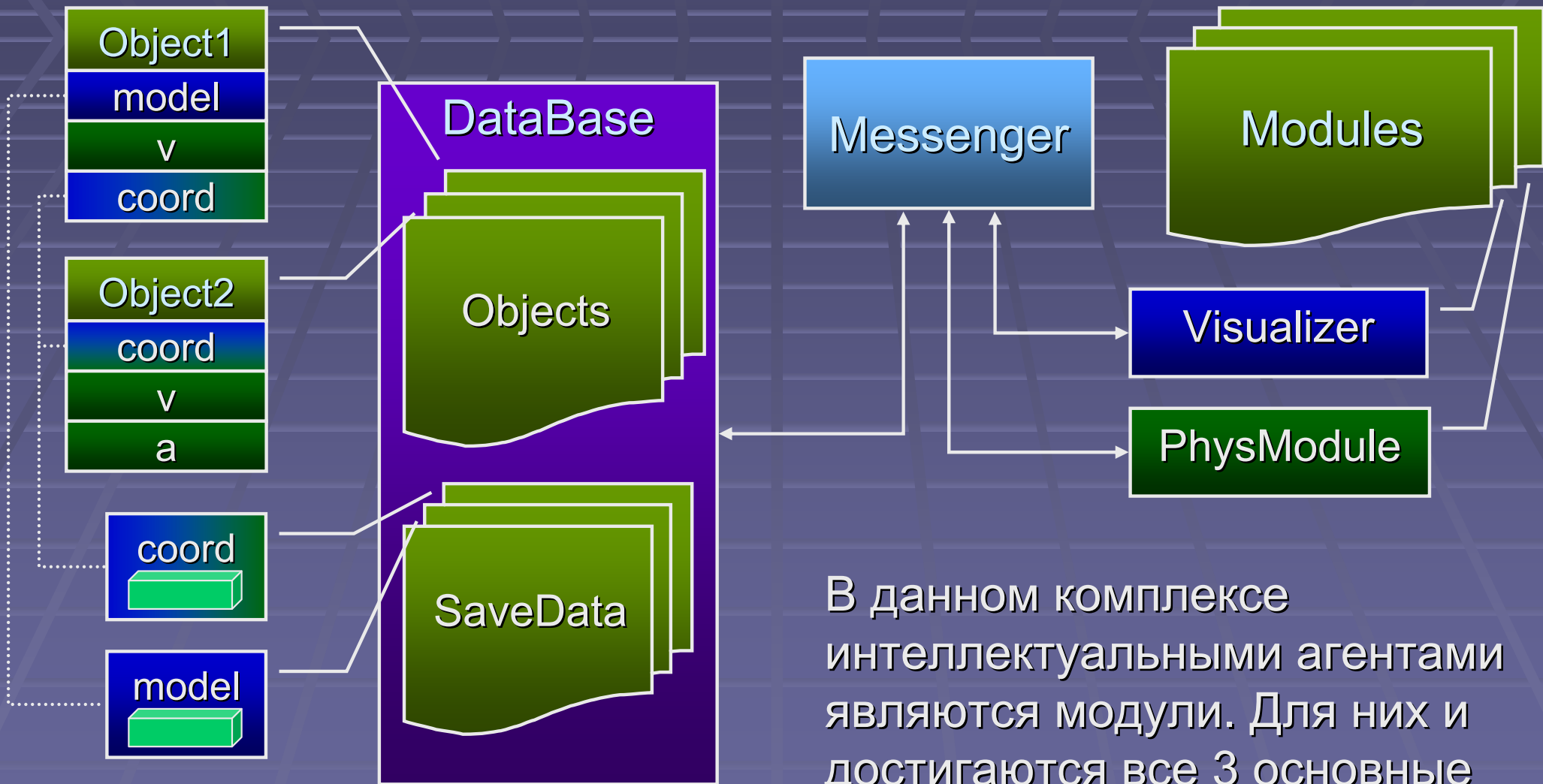


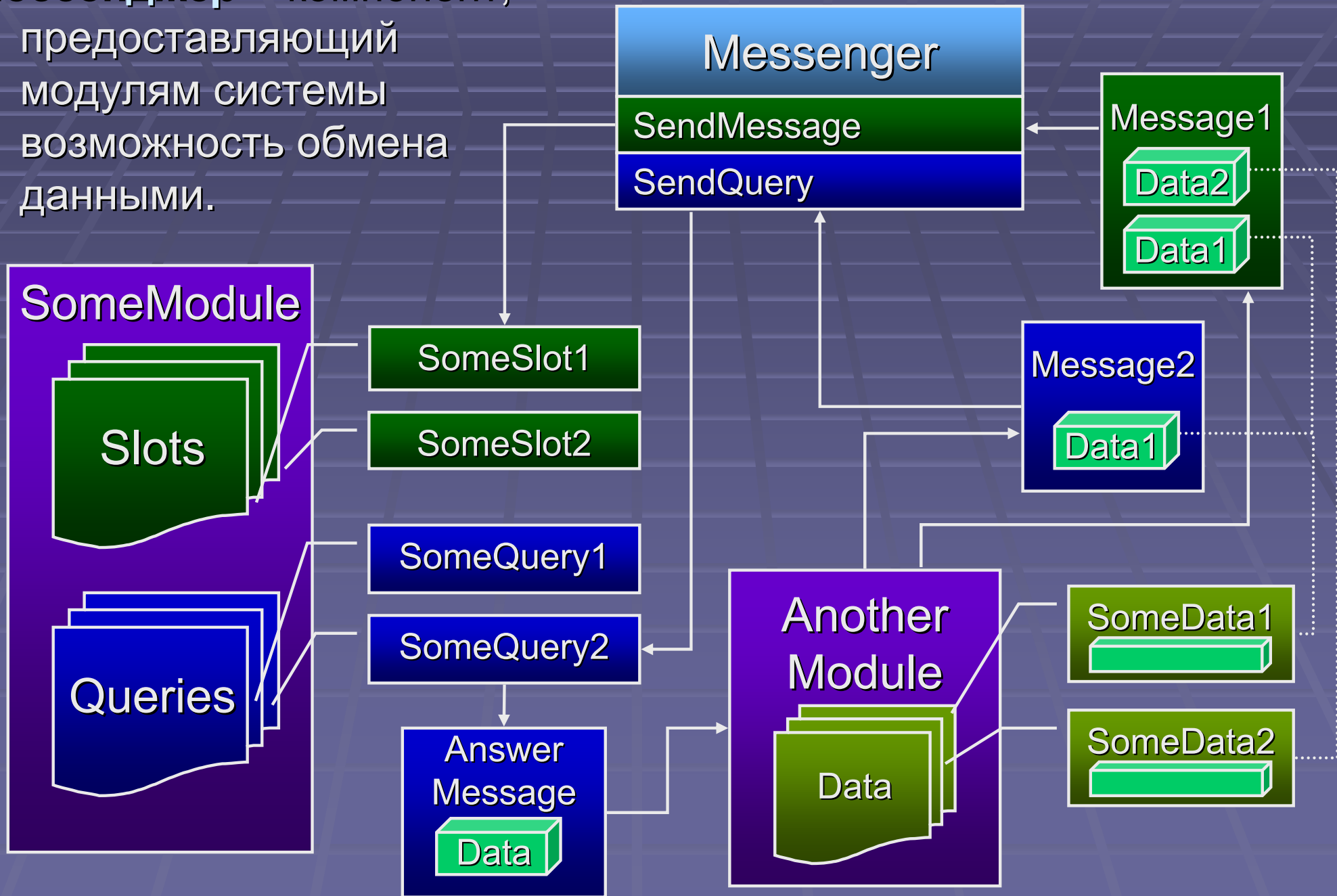
Схема мультиагентной системы



В данном комплексе интеллектуальными агентами являются модули. Для них и достигаются все 3 основные характеристики агентов МАС.

Мессенджер

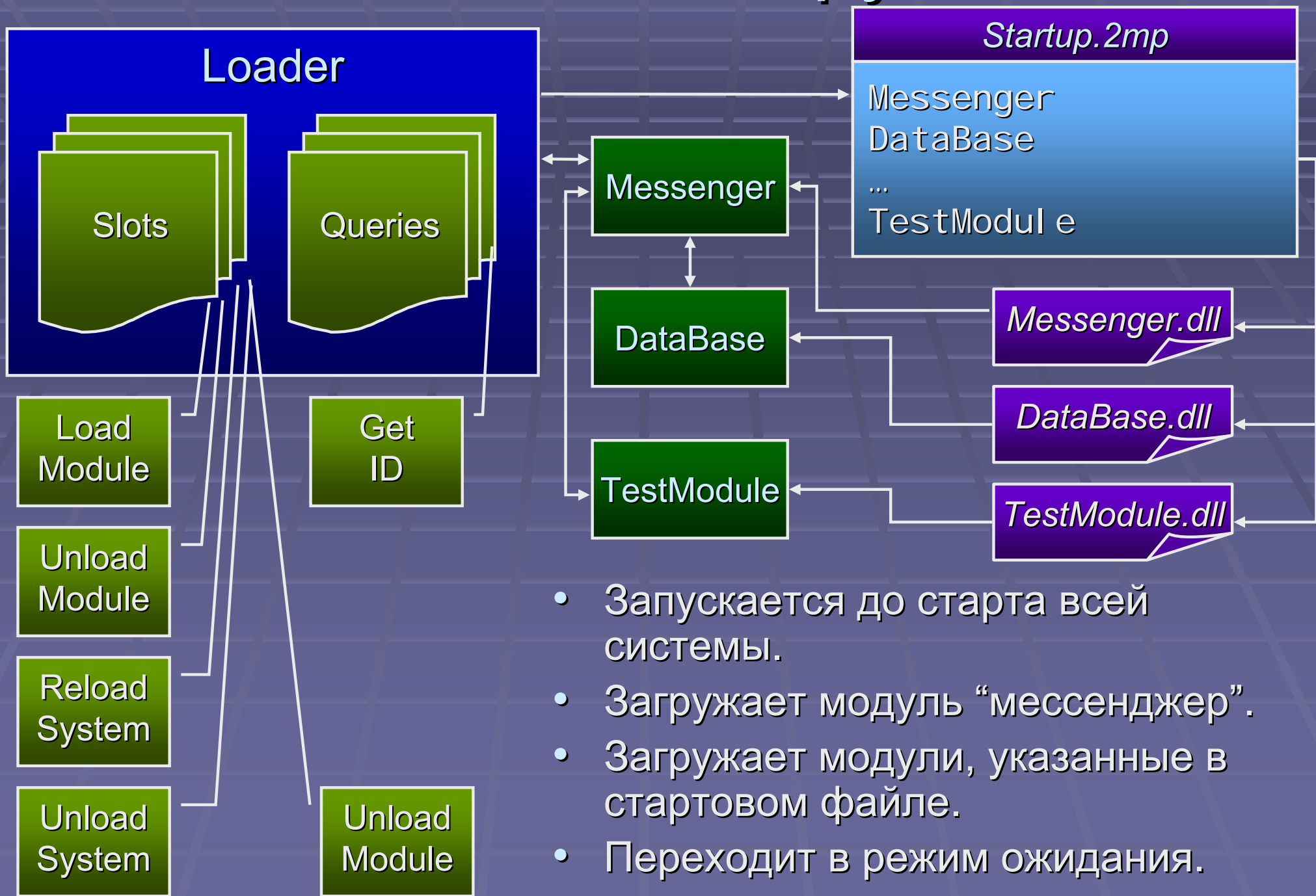
Мессенджер – компонент, предоставляющий модулям системы возможность обмена данными.



Сравнение реализованной системы сообщений с общепринятой

	Общепринятая	Реализованная
Реализация доставки сообщений	Необходимо писать функцию ожидания и переадресации сообщений	Сообщения сразу доставляются функциям-обработчикам
Порядок доставки сообщений	Необходимо вручную работать с очередью сообщений для модуля	Сообщения доставляются слотам и запросам модуля в нужном порядке
Избирательность доставки	Модулю доставляются все сообщения	Модулю доставляются лишь те сообщения, на которые он подписался

Системный загрузчик



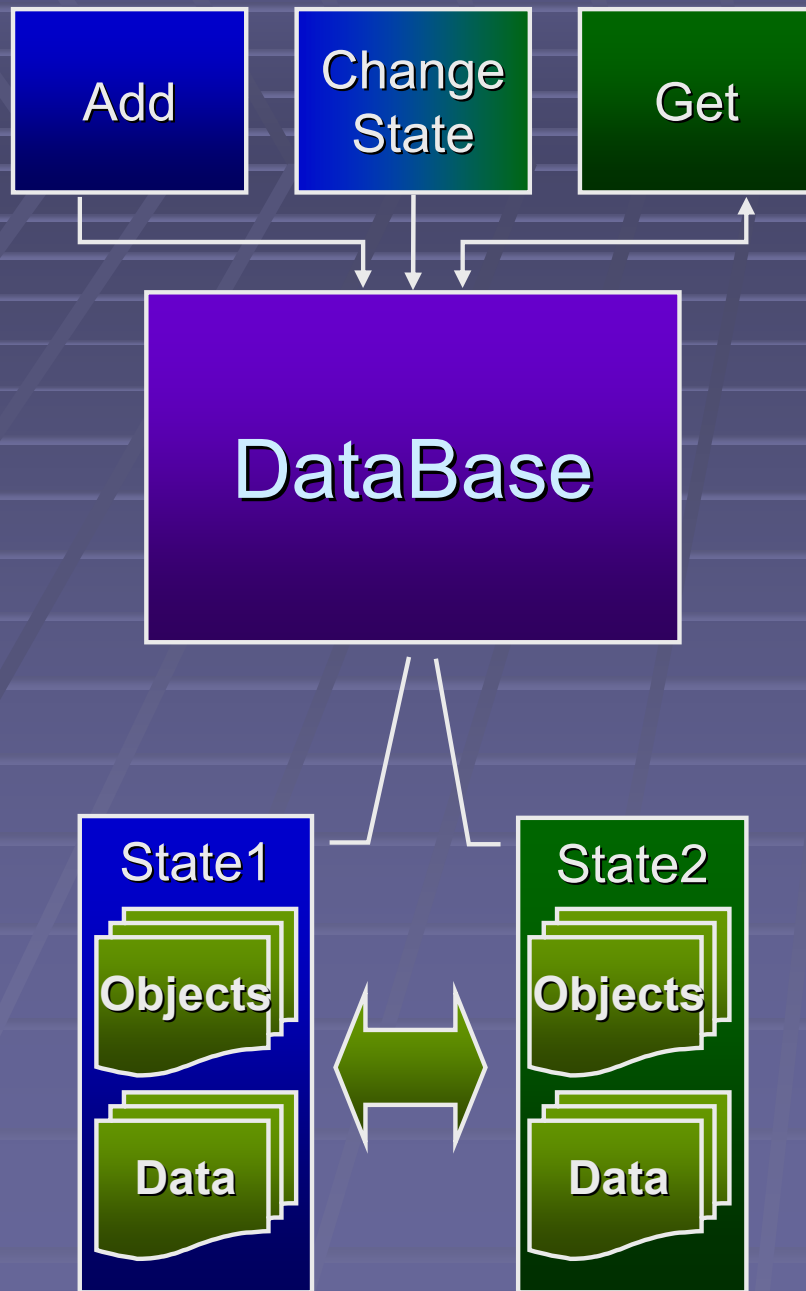
База данных

Позволяет модулям:

- создавать / удалять объекты;
- присоединять данные к объектам и отсоединять их;
- разделять одни и те же данные объекта между собой;
- получать сохранённые данные;
- изменять уже существующие данные.



Состояния базы данных



- Введены для синхронизации работы модулей с разделяемыми данными.
- Все изменения данных записываются в следующее состояние системы.
- Все данные считываются из текущего состояния системы.
- Следующее состояние системы наступает, когда каждый из модулей обработает свои данные в текущем состоянии.
- При наступлении следующего состояния отсылается сообщение о смене состояния БД.

Система отладки

Система отладки

статически линкуется
с модулем в одном из
3 режимов:

- отладка модуля;
- отладка
взаимодействий
модуля;
- отладка закончена.

В зависимости от
режима отладки для
каждого модуля по
завершении работы
системы будет
сгенерирован файл
системного журнала,
чьё содержимое
зависит от режима
отладки.

```
time: 00: 22: 46. 940  
message: functi on begi n  
thread: 3764  
functi on: voi d DataBase: :  
SetMessenger(struct IMessenger *)  
locati on: . \DataBase. cpp  
params: 10941760;
```

```
time: 00: 22: 46. 970  
message: functi on ended  
thread: 3764  
functi on: voi d DataBase: :  
SetMessenger(struct IMessenger *)  
locati on: . \DataBase. cpp  
params: 10941760;
```

*Фрагмент файла, сгенерированного
системой отладки*

Результаты работы

Был создан программный комплекс, облегчающий создание МАС

Планы на будущее

- Добавление в модуль “мессенджер” возможности отправки сообщений на другие машины через сеть, что позволит вынести модули, требующие много ресурсов, на отдельные машины.
- Добавление в загрузчик возможности загрузки модулей, написанных на языках, отличных от C++.
- Создание красивого модуля визуализации работы системы.
- Создание модуля, просчитывающего физику.
- Создание МАС для моделирования поведения роботов, обладающих ИИ, на основе данного ядра.

Спасибо за внимание