

План выполнения курсовой работы по теме
«Реализация поддержки асинхронного
программирования для фреймворка DSLab»

Содержание

1	Описание проекта и постановка задачи	3
1.1	Устройство DSLab	3
1.2	Цель	4
2	Актуальность и значимость	4
3	Существующие работы и решения	4
4	Предлагаемые подходы и методы	4
5	Ожидаемые результаты	4
6	План работ	4
	Список литературы	5

Аннотация

DSLlab - программный фреймворк для имитационного моделирования и тестирования распределенных систем.

В проекте используется дискретно-событийный подход описания моделей и приложений, где события обрабатываются в пользовательских функциях (callback-ax). В рамках проекта предстоит добавить возможность управлять событиями асинхронно.

1 Описание проекта и постановка задачи

1.1 Устройство DSLab

В силу широты охвата областей применения фреймворка он организован в виде набора слабо связанных программных модулей, использование которых будет осуществляться через их API. Это даст возможность пользователям фреймворка (исследователям, разработчикам, преподавателям) гибким образом собирать из модулей решения под свои цели, например симуляторы для конкретных типов систем или постановок задач.

Входящие в состав фреймворка модули можно условно разделить на три типа:

1. Базовые, функциональность которых используется остальными модулями (например, реализация дискретно-событийного моделирования)
2. Универсальные, функциональность которых может быть использована в различных предметных областях (например, модели сети);
3. Специализированные, которые заточены под определенную предметную область (например, библиотеки для моделирования облачных инфраструктур, исследования алгоритмов планирования заданий на кластерах или тестирования решений учебных заданий).

Архитектуру DSLab можно схематично представить в виде трех слоев (Рис. 1), включающих модули соответствующего типа. На рисунке также указаны текущие модули и зависимости между ними. Зависимости от dslab-core (от него зависят все имеющиеся универсальные и специализированные модули) не указаны, чтобы не загромождать рисунок. Таким образом, модули могут зависеть от модулей с нижних слоев, но не наоборот.

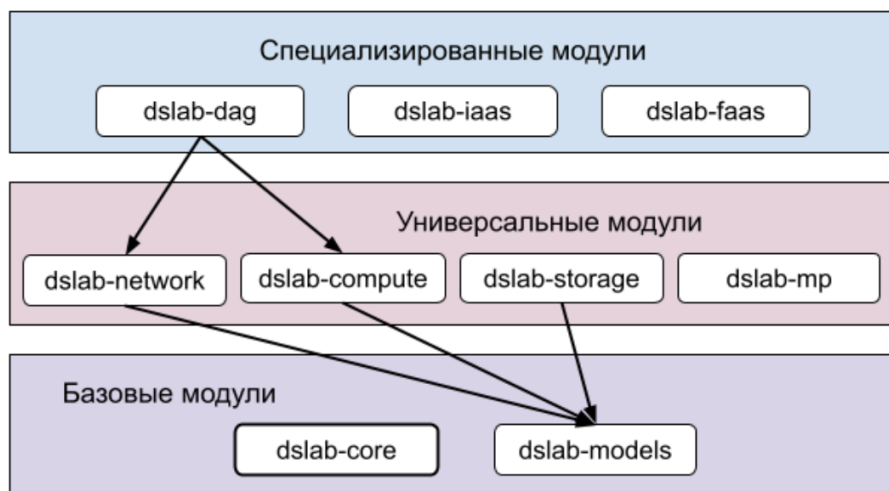


Рис. 1: Архитектура DSLab

Таким образом пользователь при разработке собственной симмуляции может либо использовать уже готовые разработанные компоненты, либо реализовывать свои и произвольно их связывать.

Основным процессом создания симмуляции является создание событий и написание процессов реагирования на них. Каждое событие представляет из себя следующую структуру:

1. идентификатор события
2. идентификатор компонента, создавшего событие
3. идентификатор компонента, которому это событие предназначено доставить
4. внутреннее время, когда событие должно наступить
5. произвольные данные события («полезная нагрузка»)

Таким образом в процессе симмуляции разные компоненты генерируют события друг для друга и с помощью самого базового модуля dslab-core обмениваются ими.

В текущей реализации проект поддерживает одну общую схему реагирования на события – через написание хэндлеров.

1.2 Цель

Добавить возможность использовать асинхронность как альтернативу текущей модели программирования с событиями и callback-ами.

2 Актуальность и значимость

3 Существующие работы и решения

4 Предлагаемые подходы и методы

5 Ожидаемые результаты

Поддержана возможность реализовывать алгоритмы в DSLab используя асинхронность.

В проект добавлены примеры кода, документация по использованию реализованных методов.

6 План работ

Требования к DSLab, документация <https://docs.google.com/document/d/1CprULnQiVSWTXiBkx90CSEi4tgkUhM3R6Ve>