

# **Доклад к ВКРБ на тему:**

## **«Программная система моделирования искусственной жизни с использованием цифровых автоматов»**

Ссылка на видео:

### **Слайд 1**

Здравствуйте, уважаемые члены государственной экзаменационной комиссии! Представляю Вашему вниманию программную систему моделирования искусственной жизни с использованием цифровых автоматов.

Целью работы было создание программной системы, благодаря которой широкий круг пользователей мог бы проводить процесс эволюционного моделирования, где для реализации участника эволюционного процесса, «так называемого» агента, используется модель цифрового автомата.

Для достижения поставленной цели необходимо было решить следующие задачи:

- исследовать существующих подходы к эволюционному моделированию;
- разработать модель агента на основе цифрового автомата;
- спроектировать архитектуру программной системы;
- обеспечить расширяемость и масштабируемость системы;
- разработать программные модули и произвести их интеграцию друг с другом;
- обеспечить работоспособность системы и разработать технологию тестирования.

Актуальность разработки обусловлена высокой популярностью генетических алгоритмов при решении прикладных задач при реализации эволюционных процессов. В связи с этим было принято решение о создании системы, благодаря которой пользователи без специальных навыков в программировании могли бы проводить моделирование эволюционных

процессов, настраивая среду и оценивая получаемые результаты. Эти результаты могли бы быть использованы для их собственных исследований.

## **Слайд 2**

В рамках исследовательской части была построена модель предметной области. Сущность «Сеанс» является определяющей и отвечает за запуск процесса моделирования и его окончание. Сущности «Генетический алгоритм» и «Метод отбора» определяют, по каким законам будет происходить модификация генетического кода агентов и их отсев. «Хранилище генов» хранит в себе «Генетический код» каждого «Агента», который изменяется в процессе работы «Генетического алгоритма». Сущность «Агент» обладает «Предками» и «Родителями», родственная связь которых формирует «Генетическое древо». Сущность «Поле визуализации» представляет пользователю информацию в графическом и текстовом виде о ходе сеанса моделирования.

## **Слайд 3**

Для построения модели агента была взята модель конечного автомата. В процессе превращения модели конечного автомата в программную модель агента каскад регистров был преобразован в блок параметров, который включает в себя параметры агента и параметры окружения. Комбинационная схема была преобразована в блок с ячейками памяти под команды агента. За счет случайных изменений номера команды в ячейке осуществляется эволюционный процесс.

Параметры агента и параметры окружения агента определяются настройками сеанса моделирования перед его запуском. Количество ячеек агента может быть также изменено, но слишком маленькое количество ячеек может привести к тому, что эволюционный процесс будет протекать слишком медленно, или вовсе остановится на раннем этапе. Слишком большое количество ячеек наоборот приведет к излишним нагрузкам на систему, что также может замедлить процесс моделирования.

## **Слайд 4**

Так как преследовалась цель охватить широкий круг пользователей, было принято решение разработать программную систему как интернет-приложение.

Для взаимодействия пользователя с системой используется React-приложение, раздаваемой с сервера Node.js. Для обработки данных используется сервер Golang, а для хранения информации реляционная база данных PostgreSQL.

Также использовался микросервисный подход, в рамках которого из сервера Golang было выделено несколько более мелких серверов, каждый из которых несет отдельную функцию. Такое разбиение позволяет упростить процесс разработки, а также использовать другие языки программирования и технологии, если в этом есть необходимость.

## **Слайд 5**

Для разработки системы и проектирования интерфейса была построена диаграмма вариантов использования. В рамках системы существует одно действующее лицо – пользователь.

Пользователь в первую очередь может зарегистрироваться, авторизоваться, просмотреть список своих сеансов моделирования и запустить новые сеансы моделирования, а также наблюдать за их процессом.

Пользователь может взаимодействовать с другими пользователями, просматривая их сеансы моделирования и отправляя им текстовые сообщения. В системе также предусмотрен личный профиль пользователя, который он может настраивать.

## **Слайд 6**

Интерфейс системы разбит на три зоны, каждая из которых несет собственный набор функционала.

Для удобства навигации было принято решение использовать в интерфейсе навигационную панель с корневыми экранами для сокращения времени перехода между экранами системы. Для отображение текущего

состояния пользователя и некоторых часто используемых функциональных элементов внедрена шапка сайта. Для отображения самого экрана используется зона контента.

### **Слайд 7**

На основе диаграммы вариантов использования был разработан интерфейс системы. Основной целью при проектировании интерфейса было достижение баланса между количеством функциональных элементов на экране и количеством самих экранов.

Было принято решение, что для перемещения к любому из экранов пользователь должен совершать не более 3-х переходов, что и было достигнуто за счёт использования «так называемого» навигационного меню, с помощью которого можно мгновенно перейти на один из корневых экранов из любого другого экрана.

### **Слайд 8**

Для хранения данных используется реляционная база данных PostgreSQL. Таблицы можно условно разделить на две группы: информация пользователей и информация сеансов моделирования.

В каждой группе присутствует служебная таблица, никак не связанная с другими. Она служит для сбора информации о системе и не влияет на общий ход её работы.

Таблица сессий показывает какие пользователи на данный момент авторизованы в системе. Таблица сообщений хранит данные о сообщениях, которыми обмениваются пользователи. Таблица пользователей хранит информацию о пользователях и их профилях.

Таблица сессий моделирования хранит общую информацию о конфигурации сеанса моделирования. Шаблон моделирования хранит информацию о возможных параметрах для каждого шаблона моделирования. Таблица агентов хранит информацию об агенте, а таблица отношения хранит информацию о родителях и потомках каждого агента.

Для сокращения количества хранимых данных вместо хранения информации о каждом дискретном моменте сеанса моделирования используются снимки, создаваемые периодически в процессе моделирования, остановки или окончания сеанса.

### **Слайд 9**

Для упрощения развертывания и поддержания системы было принято решение использовать технологию контейнеризации Docker и систему оркестровки контейнеров Kubernetes. Docker позволяет запускать компоненты системы в изолированной среде, а Kubernetes – запускать их, управлять ими и перезапускать по необходимости. Также Kubernetes предоставляет удобные средства для администрирования, такие как ресурсы service, deployment, vault и другие. Для взаимодействия кластера с внешним миром перед ним ставится прокси сервер Nginx, принимающий запросы из сети Интернет, в том числе и по сетевому протоколу HTTPS.

### **Слайд 10**

Диаграмма размещения показывает, как расположены программные компоненты системы. Из диаграммы видно, что система оркестровки контейнерами Kubernetes позволяет размещать большинство компонентов системы в своем кластере. Его можно представить в виде цельной физической машины благодаря абстракции Kubernetes. Перед кластером располагается прокси-сервер Nginx, который переправляет запросы, приходящие из сети Интернет, напрямую в кластер Kubernetes. Само графическое приложение отдается с сервера Node.js пользователю в его браузер и выполняется уже на его физической машине

### **Слайд 11**

В результате работы над ВКРБ были изучены существующие подходы к эволюционному моделированию, а также проведена классификация генетических алгоритмов, построена программная модель агента на основе цифрового автомата, спроектировано и разработано веб-приложение по

трёхзвенной архитектуре с использованием микросервисного подхода, обеспечена расширяемость и масштабируемость системы, разработаны программные модули и произведена их интеграция с друг другом, разработана технология тестирования для программной системы с использованием функционального, модульного и UX подхода.

В процессе разработки было проведено комплексное тестирование, которые показало некоторые недостатки и ошибки программного продукта, которые были по итогу устранены.

Система разработана с учётом возможного расширения, что позволяет сказать, что работа может иметь развитие в будущем.

Конец доклада. Спасибо за внимание!