

Использование модели приложения для восстановления контекста и пользовательских сценариев

Александр Киракосян

Научный руководитель: Амосов Федор Альбертович
НИУ ВШЭ - Санкт-Петербург

8 июня 2020 г.

- **Пользовательский сценарий** – некоторая последовательность действий, совершенных пользователем в приложении
- Эффективность каждого сценария описывается набором различных **метрик** – вычисляемых численных значений.
- На основе метрик принимаются продуктовые решения.

Изначальная задача и способы ее решения

Необходимо посчитать набор различных метрик для разных пользовательских сценариев.

Способы решения:

- Yandex Metric, Google Analytics
Предоставляют место для хранения данных и способ транспортировки, автоматически подсчитывают простые метрики
- BigQuery
Работает вместе с Google Analytics. Позволяет дополнительно описывать свои метрики.
- Аналитика сообщает набор требуемых событий для логирования, разработка пишет код с поддержкой данных событий, после чего аналитика считает метрики и оценивает новую функциональность.

Частые проблемы

- Игнорирование истории пользовательских действий и обработка каждого действия отдельно
- Отсутствие информации о пользовательском контексте при подсчете метрик

Итог: невозможно считать сложные контексто-зависимые метрики
Возможный способ решения: дополнительно восстанавливать контекст при подсчете метрик

Модель приложения – абстрактное представление бизнес-логики приложения

- Хранит состояние и множество возможных действий
- Изменяет состояние в соответствии с необходимыми действиями

Особенности модели:

- Упрощает логику работы приложения
- Не зависит от платформы
- Трудоемка в написании

Цель: Проверить применимость модели приложения для подсчета контекстозависимых метрик

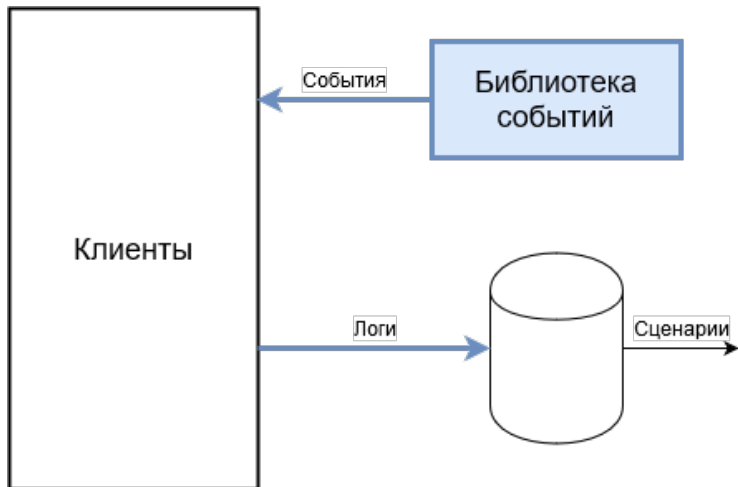
Задачи:

- Создать инструмент для сбора пользовательских сессий
- Реализовать использующий модель модуль для восстановления контекста пользователя по сессии
- Реализовать платформу для подсчета контекстозависимых метрик
- Протестировать разработанные инструменты

В качестве приложения был использован проект Яндекс Почты

- Несколько независимых клиентских приложений (Android, iOS, Desktop)
- Частично реализована модель

Сбор пользовательских сценариев



Единая библиотека клиентских событий для всех платформ

Сбор пользовательских сценариев

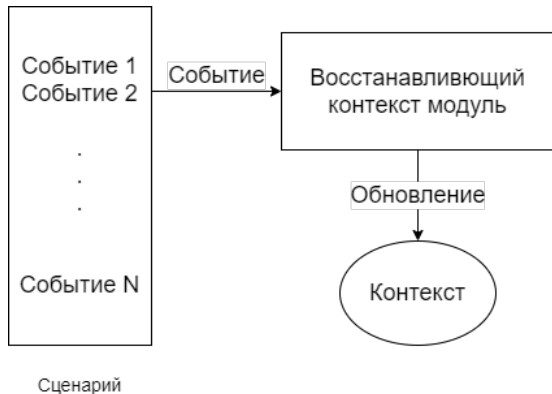
- Каждое событие описывается при помощи набора характерных аргументов
- События дополняются вспомогательной информацией

```
[  
  {  
    "event_name": "START_WITH_MESSAGE_LIST",  
    "timestamp": 1577553104420,  
    "start_event": true  
  },  
  {  
    "event_name": "LIST_MESSAGE_OPEN",  
    "timestamp": 1577553129578,  
    "mid": 171136785840080620,  
    "order": 0  
  },  
  {  
    "event_name": "MESSAGE_VIEW_DELETE",  
    "timestamp": 1577553156869  
  }  
]
```

Восстановление контекста

Модуль должен быть способен предоставлять доступ к контексту на каждом этапе сценария

- Хранит структуру, описывающую текущий контекст (модель)
- Контекст последовательно обновляется с течением сценария



Процесс состоит из инициализации и обновления модели

1. Инициализация состояния модели. Способы инициализации:

- Инициализация значением по умолчанию
- Инициализация первым событием
- Последовательное обновление состояния контекста в процессе работы

Используется последний вариант, как наиболее полный

2. Обновление контекста

- Делается через применение действий модели
- Необходимо разбиение последовательности событий на действия модели
- Не всегда возможно напрямую, из-за чего требуется уточнение в каждом конкретном случае

Подсчет метрик

Реализуется система классов, описывающих метрики и основной метод для их вычисления.

```
export interface Evaluation<T, C> {  
    acceptEvent(event: Event, context: C): void  
    result(): T  
}
```

- Для подсчета метрики в нее последовательно передаются все события сценария
- В качестве второго аргумента передается текущее состояние контекста

1. Юнит-тесты

- Валидация внутреннего устройства клиентской библиотеки
- Валидация восстановления
- Валидация метрик

2. Модельные тесты

- Валидация корректности записи на клиенте

3. Метричные тесты

- Валидация корректности работы цельного процесса

- Написана клиентская библиотека событий и внедрена в Android и iOS, содержит 64 базовых события
- Реализована платформа для восстановления контекста по пользовательским сессиям, использующая модель приложения. Контекст содержит обновляемую в процессе обработки информацию о текущем состоянии приложения
- Платформа применена для подсчета набора метрик различных сценариев. Посчитаны 12 примитивных и 6 новых, контекстозависимых метрик
- Суммарно реализовано 60 тестов на метрики и 14 тестов на клиентское логирование, которые позволили исправить часть ошибок в процессе разработки.

- Использование JS фреймворков и WebView
- Использование React Native или Flutter
 - Возможно встраивание в исходный код
 - Описывает бизнес-логику, а не интерфейсы
- Использование Kotlin Native
- **Использование трансляции**

Необходимо сгруппировать все события по пользователям и отсортировать по времени.

- Используется sql скрипт
- События группируются по пользователям и сортируются по времени
- Про каждую сессию хранятся различные данные, по которым в дальнейшем будут строиться срезы (версия системы, набор включенных экспериментов и прочее)

- Для восстановления недостаточно информации о сценарии
 - При подсчете используется вся сессия
- Многократное восстановление контекста замедляет работу
 - Подсчет всех метрик для одной сессии одновременно
 - Параллельный запуск

Используются оба подхода

- Модельные тесты
 - Сценарии генерируются из модели
 - Каждый сценарий запускается на клиенте
 - По собранным записям восстанавливаются действия модели