

Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных
систем

Биктагиров Булат Дамирович

Реализация функционала визуализации
исходных данных в рамках платформы
Desbordante

Отчёт по учебной практике

Научный руководитель:
ассистент кафедры ИАС Чернышев Г. А.

Санкт-Петербург
2022

Оглавление

1. Введение	3
2. Постановка задачи	4
3. Обзор	5
3.1. Desbordante	5
3.2. Сценарий использования	5
4. Требования	6
5. Используемые технологии	7
5.1. React	7
5.2. SASS	7
6. Реализация	8
6.1. Жизненный цикл	8
6.2. Результат	9
7. Заключение	10
Список литературы	11

1. Введение

В теории реляционных баз данных функциональная зависимость – это связь между двумя наборами атрибутов в отношении из базы данных. Другими словами, наличие функциональной зависимости показывает, что значение атрибута однозначно определяется значением некоторых других атрибутов.

Поиск функциональных зависимостей (ФЗ) имеет большое применение в сфере анализа данных. Они используются для профилирования данных [1], интеграции данных [2] и тд.

В связи с важным значением поиска ФЗ на практике появилось множество инструментов, которые помогают специалистам по обработке данных и ИТ-специалистам в решении этой задачи. Одним из таких инструментов является платформа Desbordante, предоставляющая веб-приложение для работы.

Недостатком ранее разработанного интерфейса данной платформы являлось отсутствие интерактивности, пользователь получал список ФЗ, но возможности посмотреть на исходные данные и удостовериться в корректности найденных ФЗ у него не было.

2. Постановка задачи

Целью данной работы является разработка и внедрение в графический интерфейс платформы Desbordante инструмента для визуализации функциональных зависимостей. Для её выполнения были поставлены следующие задачи:

- провести обзор предметной области;
- определить требуемый функционал;
- разработать требуемый функционал;
- внедрить его в графический интерфейс платформы Desbordante.

3. Обзор

3.1. Desbordante

Данный проект является высокопроизводительной платформой для поиска ФЗ и представляет для пользователей графический интерфейс для работы. Платформа позволяет запускать реализованные алгоритмы поиска ФЗ (такие как FD_Mine [5], Pyro [3] и другие) на исходных наборах данных. На момент начала данной практики проект имел серверную и клиентскую часть, позволяющую работать с платформой через браузер.

3.2. Сценарий использования

Пользовательский интерфейс платформы Desbordante представляет собой одностраничное веб-приложение, написанное с использованием React и TypeScript. Пользователь выбирает доступный датасет или загружает файл формата .csv. Далее выбирает один из доступных алгоритмов поиска (такие как FD_Mine [5], Pyro [3] и другие) и получает список функциональных зависимостей.

4. Требования

Для инструмента визуализации были выдвинуты следующие требования:

- Продемонстрировать часть исходного датасета;
- При выборе некоторой функциональной зависимости отобразить ее на датасете;
- Возможность вертикальной и горизонтальной прокрутки;
- Ограничение на количество строк, в связи с тем, что некоторые датасеты обладают большим количеством строк, что может повлечь за собой неудобства при работе с данным инструментом;
- Скрытие столбцов, которые не задействованы в функциональной зависимости и создают трудности при просмотре

5. Используемые технологии

5.1. React

React – это JavaScript-библиотека для проектирования пользовательских интерфейсов [4]. React разрабатывается и поддерживается Facebook, Instagram и сообществом отдельных разработчиков и корпораций.

5.1.1. Components

Компоненты являются составляющими частями любого React приложения. Компоненты концептуально похожи на обычные JavaScript функции. Они принимают аргументы (props) и возвращают React элемент, то есть то, что должно появиться на веб странице.

5.1.2. Hooks

Состояние – это JavaScript объект, который хранит динамические данные компонента и позволяет компоненту отслеживать изменения между рендерами. Хуки позволяют использовать состояние и другие возможности React без написания классов.

- `useState`, предназначен для инициализации и управления состоянием компонентов;
- `useEffect`, предназначен для перехвата различного рода изменений в компонентах, которые нельзя обработать внутри компонентов, например, получение данных с сервера.

5.2. SASS

Язык сценариев препроцессора, который интерпретируется в Cascading Style Sheets (CSS). По сравнению с CSS имеет ряд преимуществ:

- Наличие вложенных классов и свойств;
- Миксины.

6. Реализация

Поставленная задача свелась к написанию функционального компонента.

6.1. Жизненный цикл

6.1.1. Инициализация

Данный компонент является дочерним, от родительского компонента он получает параметры представленные на листинге 1

Листинг 1: Props

```
1 interface Props {  
2   taskId: string;  
3   selectedDependency: coloredDependency | undefined;  
4 }
```

- taskId – необходимый id задания, с помощью него компонент инициализирует свое состояние путем GET запроса на сервер
- selectedDependency – выбранная зависимость, которую требуется визуализировать на исходных данных

Исходные данные хранятся в состоянии компонента (листинг 2).

Листинг 2: Состояние компонента

```
1 const [table, setTable] = useState<string[] []>([]);
```

Инициализация состояния происходит с помощью хука useEffect, представленный на листинге 3

Листинг 3: useEffect

```
1 useEffect(() => {  
2   axios  
3     .get(`${serverURL}/getSnippet?taskId={taskId}`)  
4     .then((res) => setTable(res.data));  
5 }, [taskId]);
```


6.1.2. Обновление

При выборе зависимости происходит обновление компонента, с помощью вспомогательных методов и применения CSS к отдельным React элементам происходит рендеринг преобразованного состояния компонента.

6.2. Результат

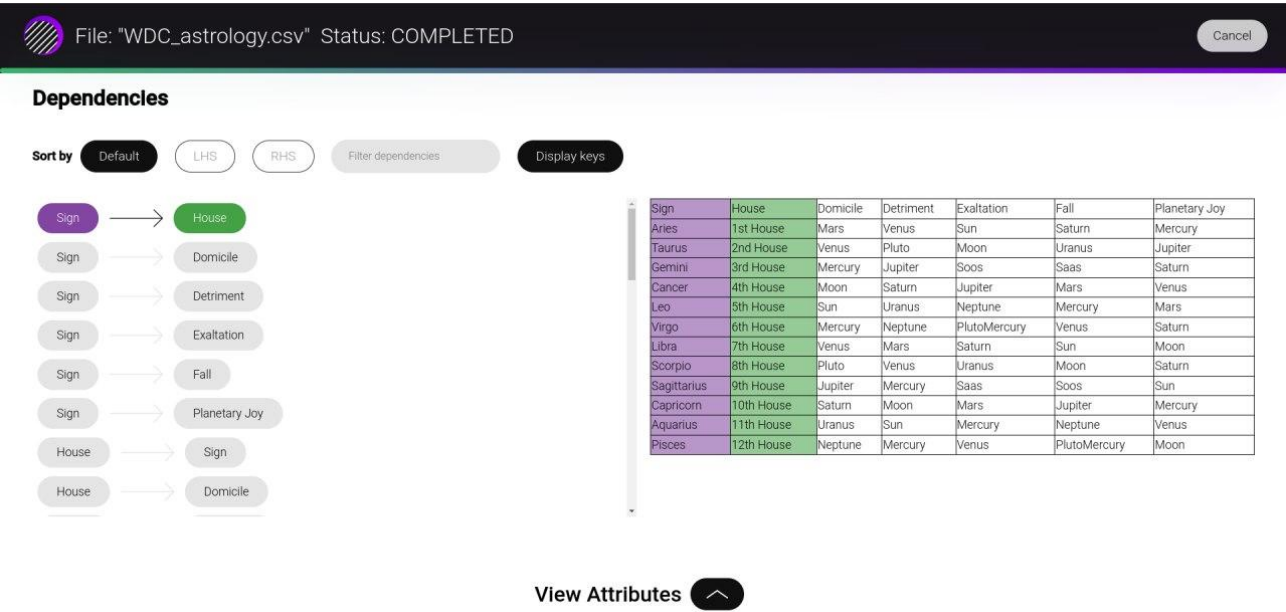


Рис. 1: Результат

На Рисунке 1 представлен измененный графический интерфейс платформы Desbordante. Слева у нас находится список ФЗ, справа исходные данные. Пользователь может выбрать ФЗ и удостовериться в ее присутствии на исходных данных.

7. Заключение

Результатом работы стало добавление возможности интерактивного просмотра исходных данных в интерфейсе платформы Desbordante. Были выполнены следующие задачи;

- проведен обзор предметной области;
- определен требуемый функционал;
- разработан требуемый функционал;
- внедрен в интерфейс платформы Desbordante.

Код является open source и доступен по ссылке¹ (ветка: web-app, имя пользователя: skrinker)

¹<https://github.com/vs9h/Desbordante>

Список литературы

- [1] Data Profiling / Ziawasch Abedjan, Lukasz Golab, Felix Naumann, Thorsten Papenbrock. — First edition. — Morgan & Claypool Publishers, 2018. — . — Vol. 10 of Synthesis Lectures on Data Management.
- [2] Ilyas Ihab F., Chu Xu. Data Cleaning. — New York, NY, USA : Association for Computing Machinery, 2019. — ISBN: [978-1-4503-7152-0](#).
- [3] Kruse Sebastian, Naumann Felix. Efficient Discovery of Approximate Dependencies // [Proc. VLDB Endow.](#) — 2018. — mar. — Vol. 11, no. 7. — P. 759–772. — URL: <https://doi.org/10.14778/3192965.3192968>.
- [4] React Documentation. — URL: <https://reactjs.org/docs/getting-started.html> (online; accessed: 2021-10-29).
- [5] Yao Hong, Hamilton Howard, Butz Cory. [FD_Mine: Discovering Functional Dependencies in a Database Using Equivalences](#). — 2002. — 01. — P. 729–732.