УДК 004.514

Проблематика построения интерфейсов под быстроменяющиеся требования

© А. С. Карякина, В. С. Кедрин
Иркутский государственный университет,
г. Иркутск, Российская Федерация

Аннотация. В статье рассматриваются технологии быстрого проектирования каналов для обмена информацией с корпоративной системой на базе платформы 1С: Предприятие 8.3, с помощью формирования динамически генерируемых интерфейсов [1]. Предлагается концепция модифицируемой компонентной архитектуры взаимодействия web-приложения с коммерческими системами хранения и обработки данных, позволяющей реализовывать динамическое проектирование интерфейса web-приложения, а также автоматизировать управление данными непосредственно из корпоративного решения на базе платформы 1С: Предприятие 8.3 [2]. Сформулированы принципы работы такого приложения, а также разработаны системные технологии организации web-приложения, способного интегрироваться в тиражные программные решения, позволяющие управлять механизмами проектирования динамического интерфейса и передачей данных от решения к web-приложению. Реализованное приложение для проектирования интерфейсов позволяет изменять компоненты web-форм, а также определять реквизиты, отображаемые пользователю интерфейсом. Приводится описание принципов динамического взаимодействия web-приложения с компонентой управления данными через интерфейс. Определены компоненты модуля взаимодействия интерфейсов с web-приложением. Сформулирована концептуальная универсальная схема управления генерацией динамических интерфейсов, позволяющая создавать практически любые виды web-форм, в том числе обработку поступающих данных с web-приложения с помощью REST-обмена [7] в формате JSON. Приведено сравнение форматов обмена.

Ключевые слова: IT-технологии в образовании, системная инженерия, сервер web-приложений Node.Js [4], фреймворк VueJS [3], формат данных JSON, web-клиент, корпоративные системы, технологии автоматизации

The problem of building interfaces for rapidly changing requirements

© Anna S. Karyakina, Victor S. Kedrin
Irkutsk State University,
Irkutsk, Russian Federation

Abstract. The article discusses the technologies for rapid channel design for information exchange with a corporate system based on the 1C: Enterprise 8.3 platform, using the formation of dynamically generated interfaces [1]. The article proposes the concept of a modifiable component architecture for the interaction of a web application with commercial data storage and processing systems, which allows implementing dynamic design of the web application interface, as well as automating data management directly from a corporate solution based on the 1C: Enterprise 8.3 platform [2]. The article formulates the principles of operation of such an application, and also develops system technologies for organizing a web application that can be integrated into circulation software solutions that allow you to control the mechanisms for designing a dynamic interface and transferring data from a solution to a web application. The implemented application for designing web application interfaces allows you to dynamically change the components of web forms, as well as determine the data displayed in the interface to the user. The article describes the principles of dynamic interaction between a web application and a data management component through an interface, defines the components of the interface interaction module with a web application, formulates a conceptual universal scheme for managing the generation of dynamic interfaces that allows you to create almost any kind of web forms, including processing incoming data from a web application using REST-exchange [7] in JSON format, as well as a comparison of exchange formats.

Keywords: IT technologies in education, systems engineering, Node.Js web application server, VueJS framework, JSON data format, web client, corporate systems, automation technologies

В связи с событиями, происходящими в мире в последнее время, изменяется подход к работе в организациях, способы взаимодействия людей, происходит переход общения в онлайн-формат. В результате подавляющее большинство коммерческих организаций столкнулось со сложностями проведения базовых бизнес-процессов: наем сотруд-

ников на работу (прием в учебное заведение), внутрикорпоративное взаимодействие. Трудности вызваны:

- Наличием нестандартных документов (различные варианты анкет, транспортные накладные, справки для работников, внутренние приказы/документы и т. д.).
- Потребностью в обеспечении эффективной коммуникации между сотрудниками (совещания, планерки, конференции и т. д.).

Большинство названных бизнес-процессов может быть переведено в онлайн-формат. Однако разрабатывать необходимый удобный интерфейс отдельно для каждого процесса дорого и трудоемко, что противоречит ориентации бизнеса на получение прибыли.

В таких случаях наиболее эффективен динамический интерфейс: он относительно легко настраивается, быстро изменяется под нужды компании и окупается быстрее.

Создание интерфейсов при традиционном подходе длительно и включает следующие этапы:

- Определение источника информации и описание способа ее получения.
- Проектирование необходимых элементов интерфейса на основании информации, полученной на предыдущем этапе.
- Разработка дизайн-макета и прототипа интерфейса с учетом всех элементов.
- Тестирование прототипа на правильность работы и заполнения.
- Исправление найденных недочетов и ошибок.
- Перенесение на рабочий сервер.

На все этапы приходится затрачивать большое количество времени, и в случае внесения правок в получаемую или выводимую информацию приходится повторять все этапы разработки.

Ускорить разработку нового интерфейса или изменение существующего возможно с помощью автоматически генерируемого «метаинтерфейса» [5, 6], который позволяет быстро создавать и изменять большое количество однотипных вариантов, оптимизируя вывод информации.

Быстрое изменение интерфейса происходит благодаря автоматической генерации дизайна по заранее подготовленным шаблонам отдельных компонентов (рис. 2.) [10]. Для обеспечения такой возможности производится стандартизация элементов интерфейса, описываются правила их вывода и заполнения данными. Для отрисовки интерфейса все его спецификации должны быть определены с сервера в бэкенд-логике приложения. Достаточно информации о содержании выводимого блока интерфейса, его классе, месте в структуре, а также в случае необходимости правилах обработки содержания.

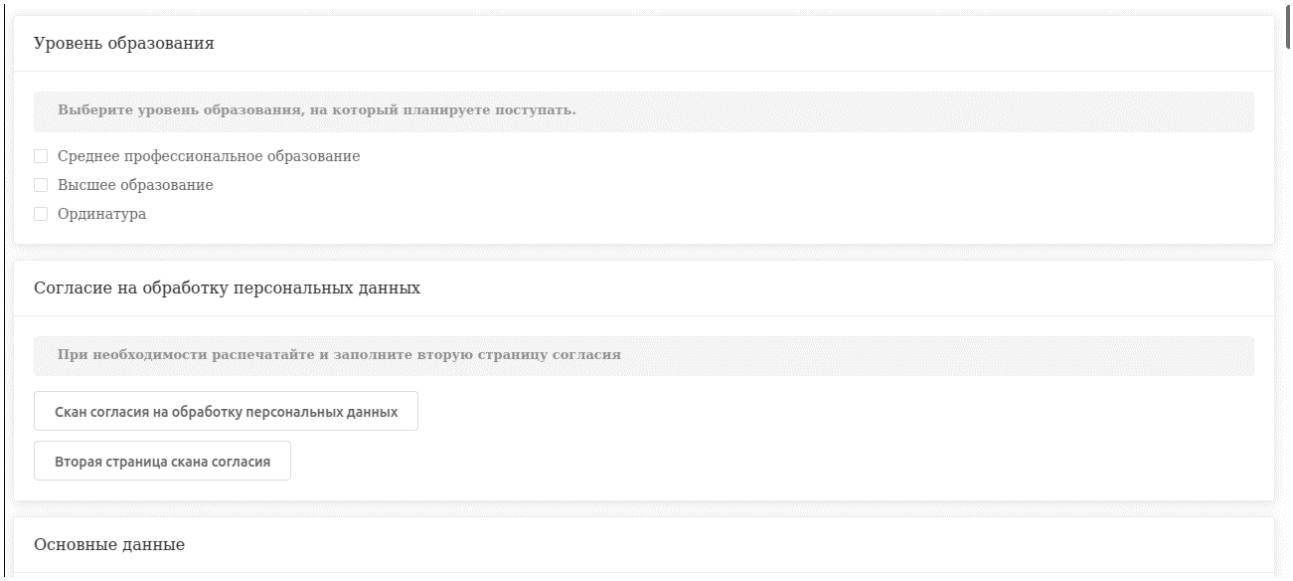

Рис. 1. Пример динамического интерфейса

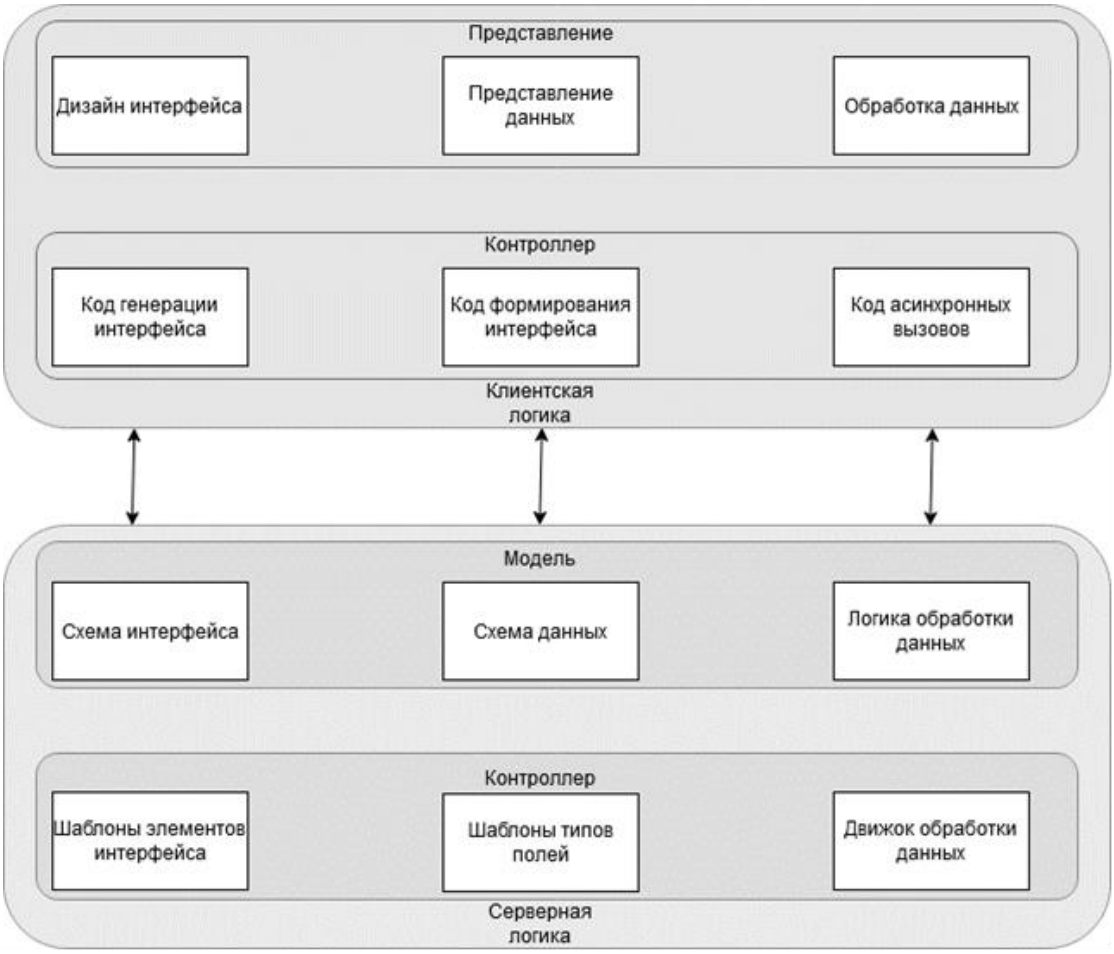

Рис. 2. Архитектура клиентской и серверной части динамического интерфейса

Кроме того, генерацию однотипных интерфейсов можно упростить еще на стадии формирования, задав нужный шаблон для отображения таких элементов.

Оптимизацию вывода информации можно рассмотреть на следующем примере. Представлен массив данных, каждый элемент которого имеет максимум 50 параметров для отображения пользователю. В обычном интерфейсе постоянно будут присутствовать все 50 параметров в незаполненном или скрытом виде, что сделает его размер и скорость разработки неудовлетворительными. В случае генерации интерфейса через его спецификацию мы сможем заранее указать отображаемые и неотображаемые параметры. Это сократит объем передаваемой информации, ускорив работу веб-интерфейса и сайта.

Рассмотрим отличия системы генерации динамических интерфейсов от классической:

1. Разработка займет больше времени, чем при классическом подходе.

2. Установка и настройка будут проходить быстрее, чем при классическом подходе.

3. Динамические интерфейсы рассчитаны на внедрение в массовый сегмент, так как в индивидуальной интеграции практически не окупаются (в данном случае подразумевается генерация целой страницы, а не отдельных её элементов).

Сходство же динамических и классических интерфейсов в том, что количество поддерживаемых элементов ограничено.

Для записи информации об интерфейсе и содержащихся в нем данных лучше всего подходит формат JSON: он читаем, прост в освоении и позволяет передавать любые данные (от текстовой строки до бинарных файлов). Также JSON поддерживает вложенность, что позволяет задавать место размещения элементов, к примеру, группировать по их типу или организовывать массивы для вывода повторяющейся информации (рис. 3).

```
{
        "name":"Stronger",
        "position":"Admin",
        "tasks":[
                {
                        "done":"true",
                        "title":"make refactoring"
                },
                {
                        "done":"false",
                        "title":"write documentation"
                },
                {
                        "done":"true",
                        "title":"remove logs"
                }
        ]
}
```

Рис. 3. Пример JSON

Таблица. Сравнение JSON и XML

JSON	XML
Формат описания объектов в текстовом виде, основанный на JavaScript	Упрощенная версия SGML, позволяющая хранить структурированные данные в человекочитаемом виде
Поддерживает текстовые и числовые типы данных, а также массивы	Не имеет прямой поддержки массивов
Поддерживает только кодировку UTF8	Поддерживает разные кодировки
Не требует наличия начального и конечного тега	Начальный и конечный теги обязательны
Не имеет поддержки нативных объектов	Поддерживает объекты через атрибуты и элементы
Не поддерживает пространство имен	Поддерживает пространство имен

XML [9] также позволяет работать с динамическими интерфейсами, но, учитывая современную тенденцию к использованию JavaScript при фронтенд-разработке, JSON [8] имеет преимущество, так как не требует дополнительных преобразований, а потому не нагружает систему пользователя. Сравнение этих двух форматов представлено в таблице.

В заключение хотелось бы отметить, что динамические интерфейсы не лишены недостатков, хотя являются эффективным инструментом для быстрого изменения дизайна сайта, позволяя создавать множество разнообразных страниц без изменения кода (или при минимальном).

Список источников

1. Кедрин В. С. Системные технологии формирования контура управления данными личного кабинета, поступающего на базе платформы «1С: Предприятие 8.3» // Информатика и образование. 2021. № 2 (321). С. 12–23.
2. Кедрин В. С., Артамонов А. Н. Личный кабинет абитуриента в рамках платформы «1С: Предприятие 8.3» // Новые информационные технологии в образовании: сборник научных трудов 21-й Международной научно-практической конференции (г. Москва, 2–3 февраля 2021 г.). М.: Изд-во: ООО «1С-Паблишинг», 2021. С. 31–34.
3. Macrae C. Vue.js: Up and Running: Building Accessible and Performant Web Apps. 6th Edition ed. Sebastopol: O'Reilly Media, 2018.
4. Волобой Ю. Д. Установка NodeJS // OpenJS Foundation [Электронный ресурс]. URL: https://nodejs.org /ru/download/package-manager/ (21.03.2022).
5. Хэнчетт Э. Vue.js в действии. СПб.: Питер, 2019. 304 с.
6. Эван Ю. Vue.JS. // VueJS Team [Электронный ресурс]. URL: https://vuejs.org// (28.04.2022).

7. Jim Webber, Savas Parastatidis, Ian Robinson REST in Practice. Hypermedia and Systems Architecture. Newcastle: O'Reilly Media, 2010. 448 с.
8. Tom Marrs JSON at Work Practical Data Integration for the Web. Denver: O'Reilly Media, 2017. 374 с.
9. Фрэнк Бумфрей, Оливия Диренцо, Джон Дакет, Джо Грэф, Дэйв Холэндер. XML. Новые перспективы WWW. М.: ДМК Пресс, 2022. 672 с.
10. Stoyan Stefanov. JavaScript Patterns: Build Better Applications with Coding and Design Patterns. Sebastopol: O'Reilly Media, 2010. 236 с.

Информация об авторах / Information about the Authors

Анна Сергеевна Карякина,
магистрант группы 0261-ДМ,
Институт математики и информационных технологий,
Иркутский государственный университет,
664003, Иркутск, бульвар Гагарина, 20,
Российская Федерация,
gsd.dfgdsf@mail.ru

Виктор Сергеевич Кедрин,
доцент кафедры вычислительной
математики и оптимизации,
Институт математики и информационных технологий,
Иркутский государственный университет,
664003, Иркутск, бульвар Гагарина, 20,
Российская Федерация,
kedrinvs@mail.ru

Anna S. Karyakina,
Student,
Institute of Mathematics and Information Technologies,
Irkutsk State University,
20 Gagarin Blvd., Irkutsk 664074,
Russian Federation,
gsd.dfgdsf@mail.ru

Viktor S. Kedrin,
Cand. Sci. (Art History),
Associate Professor, Computational
Mathematics and Optimization Department,
Institute of Mathematics and Information Technologies,
Irkutsk State University,
20 Gagarin Blvd., Irkutsk 664074,
Russian Federation,
kedrinvs@mail.ru