

Жоау Э.Р., Матвеев Ю.Н., Михальцов Н.Г. Применение шаблона проектирования MVC в разработке современных веб-систем. // Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей XIX Междунар. научно-техн. конф. – Пенза: ПДЗ, 2019. – С. 140-144.

УДК 004.415.2

ПРИМЕНЕНИЕ ШАБЛОНА ПРОЕКТИРОВАНИЯ MVC В РАЗРАБОТКЕ СОВРЕМЕННЫХ ВЕБ-СИСТЕМ

Э.Р. Жоау, Ю.Н. Матвеев, Н.Г. Михальцов

APPLICATION OF MVC DESIGN PATTERN IN THE DEVELOPMENT OF MODERN WEB-SYSTEMS

E.R. Joao, Yu.N. Matveev, N.G. Mikhaltsov

Аннотация. В статье рассматриваются общие вопросы о применении шаблона проектирования MVC при разработке современных информационных систем. Указываются некоторые особенности его использования в веб-разработке.

Ключевые слова: модель-представление-контроллер, разработка современных информационных систем, веб-разработка.

Abstract. This article discusses general questions about the application of the MVC design pattern in the development of modern information systems. Some features of its use in web development are indicated.

Keywords: Model-view-controller, development of modern information systems, web-development.

В последние годы концепция MVC (Model-View-Controller: модель – представление-контроллер) очень часто упоминается в мире веб-программирования. Бешеная популярность данной структуры в веб-системах сложилась благодаря её включению в две среды разработки, которые стали очень популярными: Struts и Ruby on Rails. Эти две среды разработки наметили пути развития для сотен рабочих сред, созданных позже.

Впервые она была описана в 1979 году, для другого окружения. Тогда не существовало концепции веб-приложения. Тим Бернерс Ли посеял семена World Wide Web в начале 90-х и навсегда изменил мир. Шаблон, который мы используем сегодня, является адаптацией оригинального шаблона к веб-разработке.

MVC – это шаблон проектирования веб-приложений, который включает в себя несколько более мелких шаблонов. При использовании MVC модель данных приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента, благодаря чему модификация одного из них оказывает минимальное воздействие на остальные или не оказывает его вовсе.

Основная цель применения MVC состоит в разделении данных и бизнес-логики от внешнего вида. За счет такого разделения повышается возможность повторного использования программного кода и упрощается

сопровождение (изменения внешнего вида, например, не отражаются на бизнес-логике).

Концепция MVC разделяет данные, представление и обработку действий пользователя на компоненты:

Модель (Model) – представляет собой объектную модель некоей предметной области, включает в себя данные и методы работы с этими данными, реагирует на запросы из контроллера, возвращая данные и/или изменяя своё состояние, при этом модель не содержит в себе информации, как данные можно визуализировать, а также не «общается» с пользователем напрямую.

Представление (View) – отвечает за отображение информации (визуализацию), одни и те же данные могут представляться различными способами, например, коллекцию объектов при помощи разных «вьюх» можно представить как в табличном виде, так и списком.

Контроллер (Controller) – обеспечивает связь между пользователем и системой, использует модель и представление для реализации необходимой реакции на действия пользователя, как правило, на уровне контроллера осуществляется фильтрация полученных данных и авторизация (проверяются права пользователя на выполнение действий или получение информации) [3].

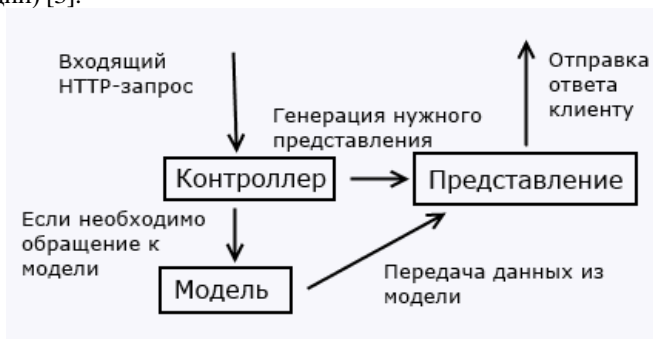


Схема работы по шаблону проектирования MVC

Простые примеры шаблона MVC в реальной веб-разработке:

Ниже представлены простые примеры MVC, которые используются в веб-фреймворках.

Контроллер

Контроллер обрабатывает входящие запросы. В фреймворке это может заключаться в определении конкретных URL, на которые попадает пользователь при переходе по ссылке или при нажатии кнопки.

Переход по ссылке `website(.)com/profile/` ->
возвращает `profilewebsite(.)com/friends/` ->
возвращает `friendswebsite(.)com/friend={userName}/` ->
возвращает профиль конкретного друга

Модель

Модель отвечает за данные, которые хранятся и обрабатываются на сервере.

```
User:- userName- firstName- lastName- friends
```

Представление

Это HTML-шаблон, который возвращает сервер после обработки запроса. Если запрос корректно обрабатывается, вы получаете веб-страницу со списком друзей. Если запрос некорректный, вы попадаете на страницу ошибки 404.

```
<ul>
  <li>Friend 1: {friendList[0].userName}</li>
  <li>Friend 2: {friendList[1].userName}</li>
  <li>Friend 3: {friendList[2].userName}</li>
  ...
</ul>
```

Самое очевидное преимущество, которое мы получаем от использования концепции MVC, – это чёткое разделение логики представления (интерфейса пользователя) и логики приложения.

Поддержка различных типов пользователей, которые используют различные типы устройств, является общей проблемой наших дней. Предоставляемый интерфейс должен различаться, если запрос приходит с персонального компьютера или с мобильного телефона. Модель возвращает одинаковые данные, единственное различие заключается в том, что контроллер выбирает различные виды для вывода данных [1].

Помимо изолирования видов от логики приложения, концепция MVC существенно уменьшает сложность больших приложений. Код получается гораздо более структурированным, и тем самым облегчаются поддержка, тестирование и повторное использование решений.

Сложность и объемы современных веб-систем растут с пугающей скоростью. Любая веб-система профессионального уровня представляет собой сложную, комплексную систему, которую невозможно воплотить в жизнь, не введя в процесс разработки определённый уровень абстракции. MVC позволяет легко превратить тысячи строк кода в стройную и логично устроенную систему, которую легко можно объять взглядом одного разработчика, и лишает разработку и последующее развитие кода множества проблем по интеграции нового кода со старым.

На сегодняшний день паттерн MVC широко применяется при разработке систем в php фреймворках (Symfony, CodeIgniter, Kohana, Yii и т.д.), а также в ASP.NET, Django, Ruby On Rails и множестве других разработок. Сегодня Model-view-controller можно назвать стандартом де-факто в веб-разработке, как, например, использование реляционных баз данных – они

подходят не для всех задач, но для большинства; а в случаях, когда они не подходят, использование нестандартных решений порой вызывает больше проблем (например, с обучением специалистов), нежели не совсем подходящие, но более привычные средства [2].

Библиографический список

1. Рогачев С. Обобщенный Model-View-Controller. URL: <http://rsdn.org/article/patterns/generic-mvc.xml> (дата обращения: 29.09.2019).

2. Дементий Д. Что такое MVC: Рассказываем простыми словами. URL: <https://ru.hexlet.io/blog/posts/chto-takoe-mvc-rasskazyvaem-prostymi-slovami> (дата обращения: 30.09.2019).

3. Model-View-Controller. URL: <https://web-creator.ru/articles/mvc> (дата обращения: 30.09.2019)

Жоау Эдивалду Родригеш

Тверской государственный
технический университет,
г. Тверь, Россия
E-mail: edivaldo.rodriquesjoao@mail.ru

Матвеев Юрий Николаевич

Тверской государственный
технический университет,
г. Тверь, Россия

Михальцов Николай Григорьевич

Военная академия
воздушно-космической обороны
имени Маршала Советского Союза
Г.К. Жукова, г. Тверь, Россия

Joao E.R.

Tver State Technical
University, Tver, Russia

Matveev Yu.N.

Tver State Technical
University, Tver, Russia

Mikhailtsov N.G.

Military Academy of Aerospace
Defense of Marshall of the Soviet
Union G. K. Zhukova, Tver, Russia