

## **Architectural Patterns in Front-End Development**

Oleh Martyn

Lviv Polytechnic National University, Lviv, Ukraine  
olegmartin12@gmail.com

**Abstract.** Software technologies tend to change, but many patterns remain the same. Nowadays in software development we use modern frameworks and libraries, but often it is hard to answer a question – why certain framework was chosen. A large number of design patterns were developed decades ago, at a time when desktop and enterprise application development were popular. They were designed by engineers to reduce the complexity of processing large amounts of information within applications. Browser applications have become increasingly popular today, but a need to process a large amount of data remains relevant. Patterns, designed ten years ago, can greatly simplify development of client side of web applications today.

**Keywords:** Pattern, web application, MVC, MVP, MVVM, FLUX, REDUX, SPA, RIA.

### **1 Introduction**

One of the main aspects of writing quality code is finding patterns and repetitions in this code for optimization purposes. Design patterns can help us with that. Design pattern is reusable solution that can be applied to a common software development problems.

Design patterns have three main advantages:

- Patterns are solid solutions: they provide proven approaches to solve software development problems. They use robust techniques which reflect an experience and understanding of developers who created them;
- Reusable: Patterns usually display a non-standard version that can be customized to suit your needs. This feature makes them quite reliable;
- They are clear and structured: when we look at pattern, we can see a structure and vocabulary of proposed solution, which can help elegantly express complex decisions;
- Patterns should be user for
- Avoiding minor issues that can cause serious problems with application development;
- Summarizing solutions which are documented in a way not connected to a specific problem;

- Reduce overall file size of our code, avoid repetition;

Design patterns are not complete solutions. They only provide a diagram for solution. Patterns do not solve all development tasks, nor do they replace good software developers, but they do support them.

## 2 Problem Formulation

With the advent of RIA (Rich Internet Application), simple, at first glance user interfaces, are supported by thousands of lines of program code. Scale of front-end application development has significantly grown, and in many cases can be more complex than server side.

So developers are facing with a the problem of choosing the right application architecture. Architecture should contain a mechanism that must direct and process data flows in both directions from&to data source. Data source can be a client database, RAM, cache, local and session storages, a server.

## 3 Analysis of Recent Researches Publications

Addy Osmani (2017) mentioned three most important architectural patterns ("Learning JavaScript Design Patterns", 2017, p. 171):

- MVC (Model-View-Controller)

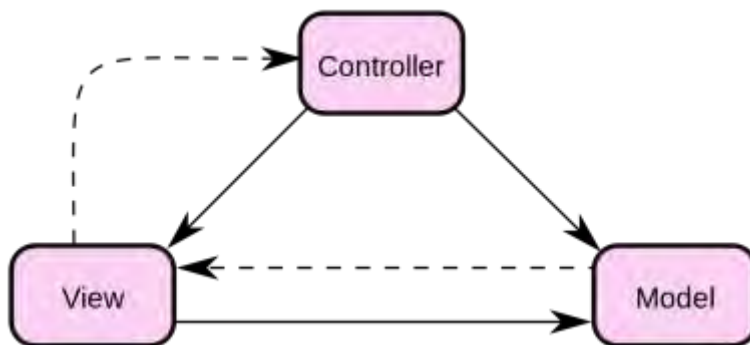


Fig. 1. Diagram of MVC components interaction

Logic is divided into three main components:

- Model - is a major component of this pattern. It contains dynamic data structures and does not depend on View
- View - visual representation of data (graphs, tables, charts, etc.)
- Controller - responsible for converting input data to Model and View commands
- MVP (Model-View-Presenter)

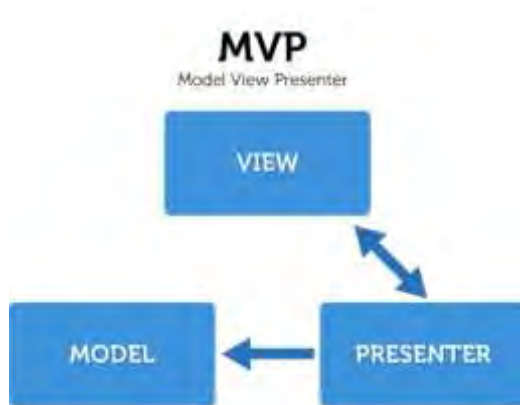


Fig. 2. Diagram of MVP components interaction

MVP template is derived from MVC. Presenter contains business logic for the relevant View component. Unlike MVC, all actions performed through interface are delegated to Presenter component.

- MVVM (Model-View-ViewModel)

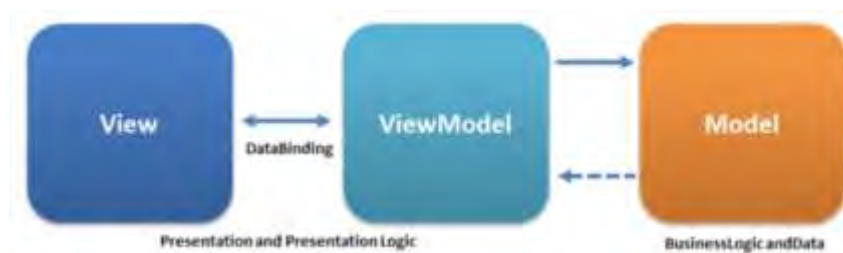


Fig. 3. Diagram of MVVM components interaction

MVVM is a modification of the MVP pattern and it is used to separate models and their representations, which is necessary to make changes separately from one another. For example, developer sets the logic for data manipulation, and designer is working on user interfaces development. ViewModel is a specialized controller designed for data conversion. It converts Model information into View information, passing commands from View to Model.

## 4 Article Purpose

In this article we will look at architectural patterns of front-end development, as well as ways to process a large amount of data when users interact with application interface, advantages and disadvantages of different patterns.

We would consider ways to avoid weak code structuring and Spaghetti code anti-pattern.

## 5 Main Material

At the beginning of the 21st century, jQuery library was the most popular tool for client-side web applications development. This library provides a convenient way to retrieve and manipulate DOM (Document Object Model) elements in your browser.

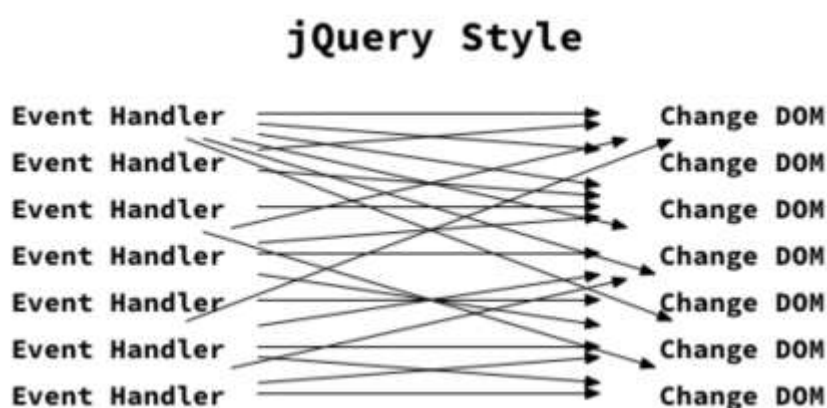


Fig. 4. Diagram of DOM elements and handlers interaction

Using this library, as the number of user interface elements increases (see Figure 4), we may notice that there is a need to handle an increasing number of DOM operations, which cost a lot. This approach also leads to a large number of complex constructions that make code unreadable.

### Two-way data binding, MVVM pattern

In the context of the Single Page Application (SPA) Model is a state of our application, and the View is an HTML template. In this case, each DOM element (or set of elements) is only interested in a particular part of a model (in particular application part). By binding each element to the appropriate part of the model, it can directly listen for changes and update its part of the model. It is called as two-way data binding (see Figure 3). Using this approach, we separate business logic of an application from its presentation layer (interface). This gives us an ability to quickly make changes to the data model, which is perfect for CRUD (create-read-update-delete) applications.

### One-way data binding, EventBus pattern

When event occurs (for example, a user clicks a button), component can send a message to EventBus. Components can also subscribe to EventBus to track specific

events, and EventBus will send messages to all subscribers. In this approach, EventBus is an intermediary, because components do not directly modify data model. This approach is also called a one-way data binding.



Fig. 5. Implementation of one-way data binding in ReactJS

Figure 5 shows that button is a sender that sends the message to EventBus, and header (h1) is a listener that subscribes to changes which will occur in some part of data model. This approach allows us to separate read/write operations and also make our code easy to maintain. Disadvantage of this approach is a lack of a way to manage who and when made changes to data model fragments.

### Flux/Redux pattern

Flux template solves the problem of implicit data change of our application, providing a mechanism for explicit mutation.

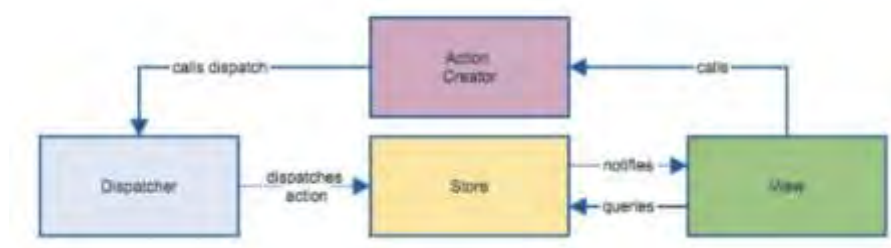


Fig. 6. Diagram of Flux pattern components interaction

The clarity of a state mutation is ensured by:

- Application state can be changed only through the Dispatcher component (see Figure 6)

- In order for the Dispatcher to change state, it is necessary to dispatch an action
- The only way to change application state is to dispatch an action

So using this approach, we will get all the information regarding data changes across entire application. We will know why/when/how changed application state.

## 6 Conclusion

In order to achieve a high level of code quality of front-end application, data model and presentation layer must be separated. We also need to choose an appropriate way to manage data flows between them.

Two-way data binding allows each element to read and update a specific part of data model. One-way communication allows us to separate write and read operations. Using the Flux template, we limit the mutation of the data model, thereby we will keep the entire state of our application under control by tracking of all the actions.

## References

1. Addy Osmani (2017). Learning JavaScript Design Patterns.
2. Graham, Paul (2002). Revenge of the Nerds.
3. Hannemann, Jan (2002). Design pattern implementation in Java and AspectJ.
4. Norvig, Peter (1998). Design Patterns in Dynamic Languages.
5. David Auerbach (2015). Rowhammer security exploit: Why a new security attack is truly terrifying.
6. Flanagan, David (2006). JavaScript: The Definitive Guide.
7. Rehorik, Jan (2008). Why You Should Never Put Sensitive Data in Your JavaScript.
8. Rusyn, B., Vysotska, V., Pohreliuk, L.: Methods of Information Resources Processing in Virtual Library. In: Computational Linguistics and Intelligent Systems, COLINS, 2, 28-39. (2018)
9. Chyrun, L., Vysotska, V., Chyrun, L., Gozhyj, O., Kalinina, I.: SEO Technology for Web Resource Processing. In: Computational Linguistics and Intelligent Systems, COLINS, 2, 40-52. (2018)
10. Demchuk, A., Lozynska, O.: The Typhlocomments Rules for Audiodescription System of the Video Content Formation for People with Visual Impairments. In: Computational Linguistics and Intelligent Systems, COLINS, 2, 53-59. (2018)
11. Lytvyn, V., Vysotska, V., Chyrun, L., Hrendus, M., Naum, O.: Content Analysis of Text-based Information in E-commerce Systems. In: Computational Linguistics and Intelligent Systems, COLINS, 2, 81-94. (2018)
12. Markiv, V., Mykyichuk, M., Markiv, O.: Detection of Gaps in Documentation Concerning Remote-piloted Aviation based on Content Analysis. In: Computational Linguistics and Intelligent Systems, COLINS, 2, 97-107. (2018)
13. Kovalchuk, V., Lytvyn, V., Vysotska, V., Hrendus, M., Naum, O.: The Information System for Identification of Content Set Based on Analysis of Similar Texts. In: Computational Linguistics and Intelligent Systems, COLINS, 2, 122-127. (2018)

14. Veres, O., Kis, Y., Kugivchak, V., Rishniak, I.: Reverse-search System of Similar or Identical Images. In: Computational Linguistics and Intelligent Systems, COLINS, 2, 129-138. (2018)
15. Pelekh, I.: Extracting and Classification the Semi-Structured Data of Web-Systems. In: Computational Linguistics and Intelligent Systems, COLINS, 2, 139-145. (2018)
16. Basyuk, T.: Influence of Readability on Popularization of Internet Resources. In: Computational Linguistics and Intelligent Systems, COLINS, 2, 146-156. (2018)
17. Cherednichenko, O., Vovk, M., Kanishcheva, O., Godlevskiy, M.: Studying Items Similarity for Dependable Buying on Electronic Marketplaces. In: Computational Linguistics and Intelligent Systems, COLINS, CEUR workshop proceedings, Vol-2136, 78-89. (2018)
18. Gozhyj, A., Chyrun, L., Kowalska-Styczen, A., Lozynska, O.: Uniform Method of Operative Content Management in Web Systems. In: Computational Linguistics and Intelligent Systems, COLINS, CEUR workshop proceedings, Vol-2136, 62-77. (2018)
19. Lytvyn, V., Dosyn, D., Emmerich, M., Yevseyeva, I.: Content Formation Method in the Web Systems. In: Computational Linguistics and Intelligent Systems, COLINS, CEUR workshop proceedings, Vol-2136, 42-61. (2018)
20. Vysotska, V., Fernandes, V.B., Emmerich, M.: Web Content Support Method in Electronic Business Systems. In: Computational Linguistics and Intelligent Systems, COLINS, CEUR workshop proceedings, Vol-2136, 20-41. (2018)
21. Висоцька, В.А. Модель і архітектура систем електронної контент-комерції: тези науково-практичної конференції, 16 травня 2007 р. / В.А. Висоцька // Науково-практична конференція “Математичне моделювання складних систем”. Львівський державний інститут новітніх технологій та управління ім. В. Чорновола. – Львів 2007. – Стор.15-16.
22. Берко, А.Ю. Аналіз і класифікація моделей систем електронної комерції / А.Ю. Берко, В.А. Висоцька // Комп’ютерні системи проектування. Теорія і практика. Вісник Національного університету “Львівська політехніка”. – Львів 2007. – № 591. – Стор.103-112.
23. Берко, А.Ю. Методи та засоби оцінки ризиків безпеки інформації в системах електронної контент-комерції / А.Ю. Берко, В.А. Висоцька, І.В. Рішняк // Комп’ютерні системи проектування. Теорія і практика. Вісник Національного університету “Львівська політехніка”. – Львів 2007. – № 591. – Стор.81-86.
24. Чирун Л.В. Інтелектуальний аналіз таблиць прийняття рішень в системах електронної контент-комерції: тези науково-практичної конференції, 16 травня 2007 р. // Л.В. Чирун, В.А. Висоцька, Т.В. Шестакевич // Науково-практична конференція “Математичне моделювання складних систем”. Львівський державний інститут новітніх технологій та управління ім. В. Чорновола. – Львів 2007. – Стор.140-149.
25. Висоцька, В.А. Моделювання системи захисту інформації в електронній комерції на основі оцінювання ризиків інформаційної безпеки: тези науково-практичної конференції, 16 травня 2007 р. / В.А. Висоцька // Науково-практична конференція “Математичне моделювання складних систем”. Львівський державний інститут новітніх технологій та управління ім. В. Чорновола. – Львів 2007. – Стор.17-22.
26. Висоцька, В.А. Аналіз та проектування інформаційних систем електронної контент-комерції: тези науково-практичної конференції, 18 травня 2007 р. / В.А. Висоцька // III міжнародна наукова конференція “Интеллектуальные системы принятия решений и прикладные аспекты информационных технологий (ISDMIT’2007)”. – Євпаторія 2007. – Стор.121-124.

27. Висоцька, В.А. Аналіз інформаційних технологій проектування систем віртуальних підприємств / В.А. Висоцька // Міжвузівський збірник наукових праць “Системні технології”. – Дніпропетровськ 2007. – № 6'(53). – Стор.30-38.
28. Висоцька, В.А. Методи та засоби опрацювання інформаційних ресурсів в системах електронної контент-комерції / В.А. Висоцька // Міжвузівський збірник наукових праць “Системні технології”. – Дніпропетровськ 2007. – № 6'(53). – Стор.39-48.
29. Висоцька, В.А. Переваги та перспективи проектування систем електронної контент-комерції / В.А. Висоцька // Комп'ютерні науки та інформаційні технології. Вісник Національного університету “Львівська політехніка”. – Львів 2007. – № 598. – Стор.161-170.
30. Висоцька, В.А. Проектування та реалізація інтелектуальних інформаційних систем інтернет-аукціону / В.А. Висоцька, А.В. Овчаренко, Л.В. Чирун // Інформаційні системи та мережі. Вісник Національного університету “Львівська політехніка”. – Львів 2008 – № 610. – Стор.70-84.
31. Висоцька, В.А. Models of functioning of electronic commerce systems: тези науково-практичної конференції, 23 травня 2008 р. / В.А. Висоцька // Міжнародна наукова конференція “Інтелектуальні системи прийняття рішення та проблеми обчислювального інтелекту (ISDMIT'2008)”. – Євпаторія 2008. – Том 2 (частина 1). – Стор.45-47.
32. Берко, А.Ю. Архітектура та інформаційні технології систем електронної комерції / А.Ю. Берко, В.А. Висоцька, Л.В. Чирун // Відбір і обробка інформації. Міжвідомчий збірник наукових праць. – Львів 2008 – № 28 (104). – Стор. 118-125.
33. Берко, А.Ю. Алгоритми опрацювання інформаційних ресурсів в системах електронної комерції / А.Ю. Берко, В.А. Висоцька, Л.В. Чирун // Комп'ютерні науки та інформаційні технології. Вісник Національного університету “Львівська політехніка”. – Львів 2008. – № 616. – Стор.128-136.
34. Висоцька, В.А. Математичні моделі інформаційних потоків в системах електронної контент-комерції: тези науково-практичної конференції, 13 листопада 2008 р. / В.А. Висоцька // VI міжнародна науково-практична конференція “Математичне та програмне забезпечення інтелектуальних систем (MPZIS-2008)”. – Дніпропетровськ 2008. – Стор.71-72.
35. Берко, А.Ю. Моделі та методи проектування інформаційних систем контент-електронної комерції / А.Ю. Берко, В.А. Висоцька // Інформаційні системи та мережі. Вісник Національного університету “Львівська політехніка”. – Львів 2008. – № 621. – Стор. 29-45.
36. Висоцька, В.А. Алгоритми та засоби опрацювання інформаційних ресурсів в системах електронної контент-комерції / В.А. Висоцька // Інформаційні системи та мережі. Вісник Національного університету “Львівська політехніка”. – Львів 2008. – № 621. – Стор. 78-96.
37. Висоцька, В.А. Особливості проектування та впровадження систем електронної комерції. / В.А. Висоцька // Комп'ютерні науки та інформаційні технології, Вісник Національного університету “Львівська політехніка”. – Львів 2008. – № 629. – Стор. 34-45.
38. Висоцька, В.А. Особливості проектування та впровадження систем електронної комерції. / В.А. Висоцька // Інформаційні системи та мережі. Вісник Національного університету “Львівська політехніка”. – Львів 2008. – № 631. – Стор. 55-84.
39. Висоцька, В.А. Моделювання систем електронної контент-комерції за допомогою мереж Петрі. / В.А. Висоцька // Комп'ютерні науки та інформаційні технології,



- Вісник Національного університету “Львівська політехніка” . – Львів 2009. – № 638. – Стор. 99-109.
40. Берко А.Ю. Проектування навігаційного графу Web-сторінок бази даних систем електронної комерції. / А.Ю. Берко, В.А. Висоцька // Комп’ютерні науки та інформаційні технології, Вісник Національного університету “Львівська політехніка”. – Львів 2009. – № 638. – Стор. 3-14.
  41. Берко А.Ю. Системи електронної контент-комерції. Монографія / А.Ю. Берко, В.А. Висоцька, В.В. Пасічник // Видавництво національного університету “Львівська політехніка”. – Львів 2009. – 612 с.
  42. Висоцька В.А. Математичні моделі інформаційних потоків у системах електронної контент-комерції. / В.А. Висоцька // Збірник наукових праць Львівського державного інституту новітніх технологій та управління ім. В.Чорновола (серія «фізико-математичні та технічні науки»). – Львів 2009. – Стор.160-167.
  43. Висоцька В.А. Математичні моделі керування контентом в системах електронної комерції: тези науково-практичної конференції, 14 травня 2009 р. / В.А. Висоцька // III міжнародна конференція молодих вчених «Комп’ютерні науки та інженерія(CSE-2009)». – Львів 2009. – Стор. 189-193.
  44. Берко А.Ю. Семантична інтеграція неповних та неточних даних / А.Ю. Берко, В.А. Висоцька, В.В. // Збірник наукових праць «Системи обробки інформації. Безпека та захист інформації в інформаційних системах», Випуск 7 (79). – Харків 2009. – Стор. 93-98.
  45. Берко А.Ю. Інтелектуальна система прийняття маркетингових рішень з поширення електронних видань / А.Ю. Берко, В.А. Висоцька, О.Ю. Михайлов // Комп’ютерні науки та інформаційні технології, Вісник Національного університету “Львівська політехніка”. – Львів 2010. – № 672. – Стор. 7-18.
  46. Висоцька В.А. Математичні моделі інформаційних потоків у системах електронної контент-комерції / В.А. Висоцька // Комп’ютерні науки та інформаційні технології, Вісник Національного університету “Львівська політехніка”. – Львів 2010. – № 672. – Стор. 185-195.
  47. Щербина Ю.М. Науковий напрям та навчальна дисципліна “Математична лінгвістика” / Ю.М. Щербина, Т.В. Шестакевич, В.А. Висоцька // Інформаційні системи та мережі. Вісник Національного університету “Львівська політехніка”. – Львів 2010. – № 673. – Стор. 384-393.
  48. Висоцька В.А. Метод вибору оптимального алгоритму криптографічного захисту інформації / В.А. Висоцька, О.Р. Гарасим // Інформаційні системи та мережі. Вісник Національного університету “Львівська політехніка”. – Львів 2010. – № 673. – Стор. 220-233.
  49. Чирун Л.В. Методи та засоби опрацювання інформаційних ресурсів в системах електронної контент-комерції / Л.В. Чирун, В.А. Висоцька // Наукові праці Чорноморського держ. Університету ім. П. Могили : Комп’ютерні технології. – Львів 2010. – Том 117. –Вип. 104. – Стор. 188-202.
  50. Chyrun Lyubomyr Using content analysis of textual information in electronic commerce: conference proceedings, October 14-16, 2010 / Lyubomyr Chyrun, Victoria Vysotska // The 5th International Scientific and Technical Conference “Computer Sciences and Information Technologies” (CSIT’2010) which will be held October 14-16, 2010 at Lviv Polytechnic National University (Lviv, Ukraine). – Lviv 2010. – Стор.80-82.