

DOI: <https://doi.org/10.33216/1998-7927-2021-268-4-7-12>

УДК 004.41:004.9

OBJECT ADAPTATION OF DRAG AND DROP TECHNOLOGY FOR WEB-SYSTEM INTERFACE COMPONENTS

Ratov D.V.

ОБ'ЄКТНА АДАПТАЦІЯ ТЕХНОЛОГІЇ DRAG AND DROP ДЛЯ КОМПОНЕНТ ІНТЕРФЕЙСУ WEB-СИСТЕМИ

Ратов Д.В.

Today, in the development of information systems, cloud technologies are often used for remote computing and data processing. There are web technologies, and on their basis, libraries and frameworks have been developed for creating web applications and user interfaces designed for the operation of information systems in browsers. Ready-made JavaScript libraries have been developed to add drag and drop functionality to a web application. However, in some situations, the library may not be available, or there may be overhead or dependencies that the project does not need to use it. In such situations, an alternative solution provides the functionality of APIs available in modern browsers. The article discusses the current state of the methods of the Drag and Drop mechanism and proposes a programmatic way to improve the interface by creating a class for dragging and dropping elements when organizing work in multi-user information web systems. Drag and Drop is a convenient way to improve the interface. Grabbing an element with the mouse and moving it visually simplifies many operations: from copying and moving documents, as in file managers, to placing orders in online store services. The HTML drag and drop API uses the DOM event model to retrieve information about a dragged element and update that element after the drag. Using JavaScript event handlers, it is possible to turn any element of the web system into a drag-and-drop element or drop target. To solve this problem, a JavaScript object was developed with methods that allow you to create a copy of any object and handle all events of this object aimed at organizing the Drag and Drop mechanism. Basic algorithm of Drag and Drop technology based on processing mouse events. The software implementation is considered and the results of the practical use of object adaptation of the Drag and Drop technology for the interface components of the web system - the medical information system MedSystem, in which the application modules have the implementation of the dispatcher and the interactive window interface are presented. In the "Outpatient clinic" module, the Drag and Drop mechanism is used when working with the "Appointment sheet." In the "Hospital" module of the MedSystem medical information system, the Drag and Drop mechanism is used in the "List of doctor's appointments." The results of using object adaptation of Drag and Drop technology have shown that this mechanism organically fits into existing technologies for building web

applications and has sufficient potential to facilitate and automate work in multi-user information systems and web services.

Keywords: *object adaptation, Drag and Drop, information system.*

Introduction. Today, in the development of information systems, cloud technologies are often used for remote computing and data processing [1]. Cloud computing is a distributed data processing process in which computer resources and network capacity are provided to the user as an Internet service [2]. Cloud technology inherently implements the processes of creating cloud applications and organizes work with them, without the introduction of additional software. Typically, for such applications, functionality is created in a web browser environment. Such a software product is a client-server application with a Web interface that provides the user with the ability to access data from any active point, provided that they are connected to the Internet [3].

For effective interaction of the client with remote data without completely reloading the current page, the user interface template [4] is used, which is put into the structure of modules that implement controls, input, sending and receiving data in the form of windowed web-forms with their inherent functionality in the browser context. A web form in work means an independent fragment of the user interface with its own logic of behavior, for the display of which the template objects of the module are used. One of the purposes of such a module is to reuse it. This allows you to define the functionality of objects once and use them in different contexts and information systems.

Today there are web technologies, and libraries and frameworks developed on their basis for creating web applications and user interfaces intended for the operation of information systems in browsers [5, 6, 7,

8]. The processes of standardization of HTML [9], CSS [10] and javascript [11] languages allowed achieving not only a high degree of cross-platform user interfaces, but also a fairly good degree of cross-browser compatibility, so the use of appropriate standards when building Web applications has become the dominant approach.

Purpose of the study. Creation of a JavaScript class for components of a web-system with implementation in the context of the user interface of the Drag and Drop mechanism.

Description of software tools. Drag and drop is a common user action found in many graphical user interfaces. There are JavaScript libraries out of the box for adding drag and drop functionality to a web application. However, in some situations, the library may not be available, or there may be overhead or dependencies that the project does not need to use it. In such situations, an alternative solution provides the functionality of APIs available in modern browsers.

Drag'n'Drop is a convenient way to improve the interface. Capturing an element with a mouse and moving it visually simplifies many operations: from copying and moving documents (as in file managers) to placing orders in online store services.

In modern HTML5 standard there is Drag and Drop section, which introduces special events for Drag'n'Drop transfer, such as dragstart, dragend [12]. They are remarkable in that they make it easy to solve simple problems. For example, transferring a file to a browser, with the ability to access its contents. But there are also limitations: there is no organization of the transfer "only horizontally" or "only vertically", it is impossible to restrict the transfer within a given zone. There are other front-end tasks that are not implemented by such built-in events. Also, mobile devices don't support them well.

Firefox and other Mozilla applications have a number of drag and drop options. This allows the user to press and hold down the mouse button over the element, move it to a different position; by releasing the mouse button, the user can leave the element at the new

position. Throughout the move operation, the semi-transparent representation of the element follows the mouse cursor. The new position of the element can be located in a completely different application. Websites and XUL applications can use this functionality to determine which elements on a page can be moved, as well as to determine which elements can be moved to.

API interface HTML Drag and Drop uses the DOM event model to get information about the element being dragged and update that element after dragging. With JavaScript event handlers, it is possible to turn any element of the web system into a drag-and-drop element or a drag-and-drop target.

Algorithmic and software implementation. In this paper, we have considered creating a class for drag-and-drop elements using the HTML Drag and Drop API with JavaScript without extensions. At the same time, object adaptation and implementation of Drag'n'Drop technology takes place using mouse event handlers.

The basic algorithm of the Drag'n'Drop technology (Fig. 1), based on processing mouse events:

1. When the mousedown event occurs - we prepare the element for movement - create a copy of it and place the copy (avatar) in the same place, but in absolute coordinates (Fig. 2, lines 14-31).

2. When the mousemove event occurs, move the avatar object to new coordinates - under the coordinates of the cursor, by changing left / top at position: absolute and shifting them by half the width / height for centering (Fig. 2, lines 35-37).

3. Tracking the end of the transfer (Fig. 2, line 40), when the mouseup event of the avatar object occurs, we stop the transfer of the avatar element and perform actions related to the end of Drag'n'Drop - we execute the function whose name is passed as the 2nd parameter (Fig. 2, line 50).

The browser for pictures and some other elements has its own ondragstart handler for the Drag'n'Drop event, which is automatically launched and may conflict with the custom one. Therefore, it must be disabled (Fig. 2, line 20).

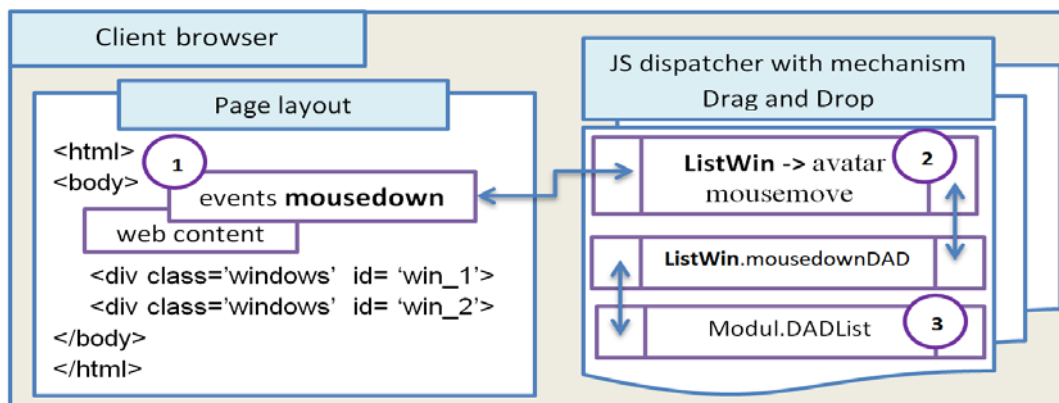


Fig. 1. Scheme of the basic Drag'n'Drop algorithm based on handling mouse events

```

12 ListWin.mousedownDAD = function(object, funcMouseUp, hideClone) {
13   let main = this,
14       avatar = object.cloneNode(true),
15       styleDefault = object.style.border;
16
17   main.moveElement = object;
18   main.avatar = avatar;
19   object.style.border = '4px dotted #c8300e';
20   avatar.ondragstart = function() { return false; }
21
22   avatar.style.position = 'absolute';
23   avatar.style.left = event.clientX + 'px';
24   avatar.style.top = event.clientY + 'px';
25   avatar.style.width = object.offsetWidth + 'px';
26   avatar.style.height = object.offsetHeight + 'px';
27   avatar.style.zIndex = 10000;
28   avatar.style.border = 'solid 2px #c8300e';
29   avatar.style.color = '#0e72a8';
30   avatar.style.background = 'ffffff';
31   avatar.hideClone = (hideClone !== undefined ? hideClone : true);
32
33   document.body.appendChild(avatar);
34
35   document.onmousemove = function(e) {
36     avatar.style.left = e.pageX - avatar.offsetWidth / 2 + 'px';
37     avatar.style.top = e.pageY - avatar.offsetHeight / 2 + 'px';
38   }
39
40   avatar.onmouseup = function() {
41     let downEl = null, downTh = null;
42     avatar.hidden = true;
43     main.downElement = document.elementFromPoint(parseInt(avatar.style.left, 10),
44                                                    parseInt(avatar.style.top, 10));
45     avatar.hidden = false;
46     document.onmousemove = null;
47     avatar.onmouseup = null;
48     if (avatar.hideClone) avatar.parentNode.removeChild(avatar);
49     object.style.border = styleDefault;
50     (new Function(funcMouseUp))();
51   }
52 }

```

Fig. 2. The mousedownDAD method of the ListWin dispatcher, which implements the Drag'n'Drop algorithm

```

1   var ListWin = new ListWin();
2
3    (date accessed: 05/15/2021).
6. jQuery. URL: <http://jquery.com/> (date of access: 05/15/2021).
7. AngularJS - Superheroic JavaScript MVW Framework. URL: <http://angularjs.org/> (date accessed: 05/15/2021).
8. Gamma E., Helm R., Johnson R., Vlissides J. Techniques of object-oriented design. Design patterns. SPb. : Peter, 2001. 368 p.
9. HTML 4.01 Specification. URL: <https://www.w3.org/TR/html401/> (date accessed: 05/15/2021).
10. Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification. URL: <https://www.w3.org/TR/CSS22/> (date accessed: 05/15/2021).
11. ECMAScript Language Specification - ECMA-262 Edition 5.1. URL: <http://www.ecma-international.org/ecma-262/5.1/> (дата звернення: 15.05.2021).
12. Стоян Стефанов. Javascript. Шаблоны. Санкт-Петербург, 2011. 263 с.
13. Ратов Д.В. Модель модуля інтерфейсу користувача інформаційної web-системи. Математичні машини і системи. 2020. № 4. С. 74–81.

#### **Ратов Д.В. Об'єктна адаптація технології drag and drop для компонентів інтерфейсу веб-системи**

*Сьогодні при розробці інформаційних систем хмарні технології часто використовуються для віддалених обчислень та обробки даних. Існують веб-технології, а на їх основі розроблені бібліотеки та фреймворки для створення веб-додатків та користувальницьких інтерфейсів, призначених для роботи інформаційних систем у браузерах. У статті розглядається сучасний стан методів для реалізації механізму перетягування компонент та пропонується програмний спосіб вдосконалення інтерфейсу шляхом створення класу для елементів перетягування при організації роботи в багатокористувацьких інформаційних веб-системах. Drag'n'Drop - це зручний спосіб покращити інтерфейс. Захоплення елемента мишкою та його візуальне переміщення спрощує багато операцій: від копіювання та переміщення документів (як у файлових менеджерах) до розміщення замовлень в послугах Інтернет-магазину. API перетягування HTML використовує модель подій DOM для отримання інформації про перетягнутий елемент та оновлення цього елемента після перетягування. За допомогою обробників подій JavaScript можна перетворити будь-який елемент веб-системи в елемент Drag'n'Drop. Для вирішення цієї проблеми були розроблені об'єктні методи JavaScript, які дозволяють створювати копію об'єкта та обробляти всі події, спрямовані на організацію механізму Drag'n'Drop. Розглянуто програмну реалізацію та представлено результати практичного використання об'єктної адаптації технології Drag and Drop для компонентів інтерфейсу веб-системи - медичної інформаційної системи MedSystem. Результати використання об'єктної адаптації технології Drag and Drop показали, що цей механізм органічно вписується в існуючі технології побудови веб-додатків і має достатній потенціал для полегшення та автоматизації роботи в багатокористувацьких інформаційних системах та веб-сервісах.*

**Ключові слова:** адаптація об'єкта, Drag and Drop, інформаційна система.

**Ратов Денис Валентинович** – к.т.н., доцент кафедри програмування та математики, Східноукраїнський національний університет імені Володимира Даля (м. Северодонецьк), [ratov@snu.edu.ua](mailto:ratov@snu.edu.ua)

#### **Література**

1. Медведев А. Облачные технологии: тенденции развития, примеры исполнения // Современные технологии автоматизации. 2013. № 2. С. 6–9
2. Papadopoulos A., Katsaros D. A-Tree: Distributed Indexing of Multidimensional Data for Cloud Computing Environments. Third IEEE Intl Conf. on Cloud Computing Technology and Science, 2011, pp. 407–414.
3. Крейн Д., Паскарелло Э. Аjax в действии. М.: Изд. дом «Вильямс», 2006.
4. Ratov Denis Architectural paradigm of the interactive interface module in the cloud technology model. Applied Computer Science. 2020. Vol. 16, N 4. P. 48–55.
5. Архитектура MVP. URL: <http://www.gwtproject.org/articles/mvp-architecture.html> (дата звернення: 15.05.2021).
6. jQuery. URL: <http://jquery.com/> (дата звернення: 15.05.2021).
7. AngularJS — Superheroic JavaScript MVW Framework. URL: <http://angularjs.org/> (дата звернення: 15.05.2021).
8. Гамма Э., Хелм Р., Джонсон Р., Влссидес Дж. Приёмы объектно-ориентированного проектирования. Паттерны проектирования. СПб. : Питер, 2001. 368 с.
9. HTML 4.01 Specification. URL: <https://www.w3.org/TR/html401/> (дата звернення: 15.05.2021).
10. Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification. URL: <https://www.w3.org/TR/CSS22/> (дата обращения: 15.05.2021).