Journal scientific and applied research, vol. 27, 2024
International Journal

ISSN 1314-6289 (Print)
ISSN 2815-4622 (Online)

Original Contribution

AN APPROACH OF DOM RECORD MANAGEMENT COMPONENT UI CREATION

Valentin T. Atanasov

COMMUNICATION AND COMPUTER TECHNOLOGIES, FACULTY OF TECHNICAL SCIENCES, KONSTANTIN PRESLAVSKY UNIVERSITY OF SHUMEN, SHUMEN 9712, 115, UNIVERSITETSKA STR., E-MAIL: v.atanasov@shu.bg

ABSTRACT: *This paper presents an approach of Document Object Model database record management component user interface (UI) creation. Mainstream of UI development approaches concentrates in few directions, omitting holistic view on the fundamental database operations. The proposed solution uses method lay on of standard clauses adopted by the ECMA Script edition sixth and above. The program logic for the UI component was built with the intention - The existing functionality of such UI component to be extended and complemented.*

KEY WORDS: *Record set, UI, DOM, JavaScript, Database, HTML.*

1. Introduction

According to neuroscientists data visualization is a vital tool of every discipline, revealing relationships in datasets and communicating information. Such claims involved visualization techniques [1][2]. Moreover, there is an important theoretical basis for optimizing the human visual information process laid on the theory of cognitive fit [3].

As the data visualization is in direct correlation with the user experience it could be stated that user interface could impact cognitive impressions and responses to the system [4]. Such systems should possess human-computer interaction functionality and has broad disciplinary implementations, including educational field in university [5].

When access to database data is needed in Web-based Client-Server application model, often certain set of these accessed data have to be displayed on the client agent's display window or so-called browser window. As stated above, the importance of the way of onscreen data presentation is among the key factors of the user's effectiveness in processing these data.

This paper will present an approach to building a database record management UI component via a method laid on the standard clauses of the ECMA Script edition. (issue sixth and above). As noted in the ECMAScript standard, the ECMAScript language is based mostly on JavaScript (JS) [7]. One of the important reasons of that programming language choice is the fact that JavaScript is used as a programming language over 62% among common developer`s society [8]. Following the previous lines in this study, the main intention is to find a program algorithm for extending and complementing the functionality of the existing program UI models for record management in the Client side of Client-Server model.

2. Related works

In the field of Web-based programming, UI modeling approaches take a crucial place in the development process. Wide adopted practice of building UI for database access and data visualization results in certain solutions. But it should be noted that these solutions are limited in functionality. Mainstream of UI development approaches concentrates in few directions [6],[9],[10],[11] :

- Recordset.
- Navigation;
- Filtering;
- Paging.

As a part of the mentioned above directions is a particular database query resulting in set of data, forming specific information structure. This queried information structure is known as *recordset*. It is first step of user's information processing. The next three directions - *navigation*, *paging*, and *filtering* comprise just part of the database UI domain of operations. This leads to incompleteness of such program models of the UI data visualization and its processing by the user respectively.

As long as these program models satisfies certain logic, there are glimpse of gap in it. There are precisely defined operations, related to database. These are *Create*, *Read*, *Update*, and *Delete* (CRUD) operations [12][13][14]. This appears to be fundamental for whole database processing. The main gap of the existing record management UI solutions is the lack of the holistic approach that could comprises all these operations in one single UI component. In this logic an approach of record management UI component (RMUI) creation will be exposed in the next chapter.

3. An approach of record management component UI creation.

As good practice suggests - in the beginning of the development process of the program components, a set of requirements should be defined. In regard of the journal volume requirements, in the Table 1. only a main set of requirements is presented.

Table 1. Main requirements for record management UI component.

Used language	JavaScript, HTML, CSS, SQL, PHP
Application model	Client-Server with test database
Usage of JS library	Custom developed
Environment	Local server deployed*
Covering database operations	Create, Read, Update, Delete
Testing platforms	Chrome ver. 130, Firefox ver. 132, Opera ver. 114
Operating environment	Microsoft® Windows 7, 8, 8.1,10 and 11
UI component	Based on external JS module
Used model	Document Object Model
Used method for server requests	HTTP request

***Note**: As local server environment an WAMP Server is used [16].

As the UI conception suggests, just a Client side of Client-Server Web application should be regarded. There will be no server code or server logic presented here. Following good practice in the program development process, the conception of RMUI component should be built. On the figure 1 the graphical representation of that conception is shown.

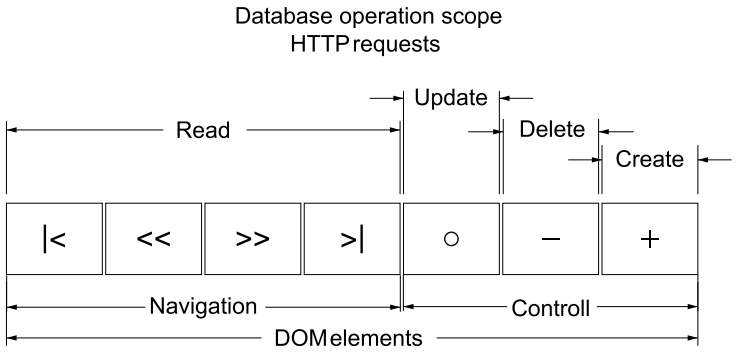

Fig. 1. A graphical representation of the conception of RMUI component.

The main goal of this conception is to encompass all database operations in Web-oriented applications in one single complex component, built by the standard DOM elements. Each of these DOM elements should possess specific functionality. All valid states in that RMUI component are presented in Table 2.

Table 2. Valid states of the record management UI component.

q_0	q_1	q_2	q_3	q_4	q_5	q_6
true	false	false	false	false	false	false
false	true	false	false	false	false	false
false	false	true	false	false	false	false
false	false	false	true	false	false	false
false	false	false	false	true	false	false
false	false	false	false	false	true	false
false	false	false	false	false	false	true

Here $q_i = f(\theta, e_i)$, where $i \in \{0..6\}$
θ - valid database read operation;
e_i- *onclick* event of UI DOM element i.

In fig.2, a particular transition diagram, a specific part of the whole model, of the modeled UI component is presented.

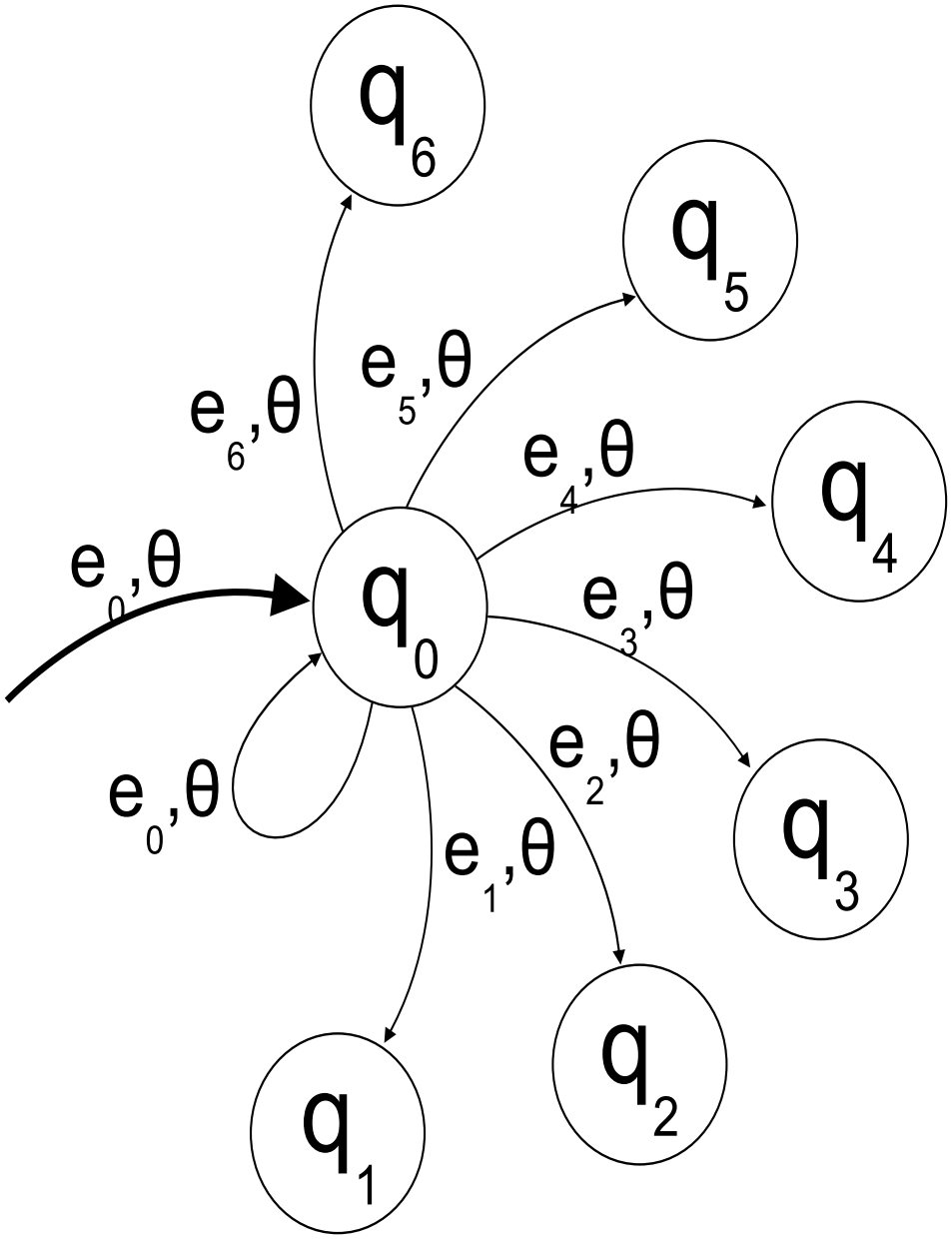

Fig. 2. Particular **t**ransition diagram of the modelled UI component.

Next follows coding cycle planning phase. On the fig. 3. a development process is presented.

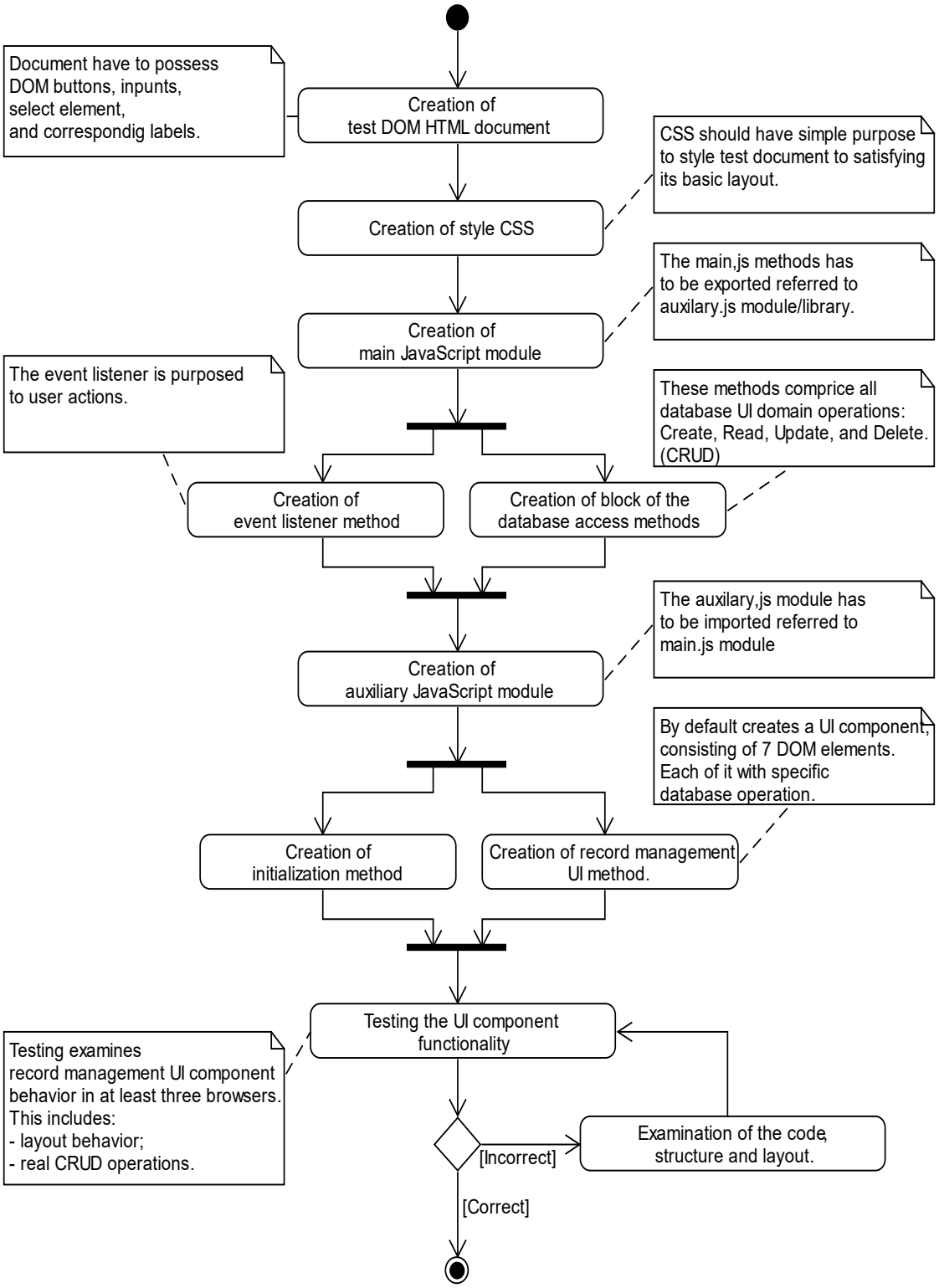

Fig. 3. A development process of record management UI component.

As a language, JavaScript is laid on the standard clauses of the ECMA Script. In the development phase certain JavaScript statement are used to. One

of the key statements are declarations of *import/export* which ensures runtime import/ export evaluation process. For example, the following code (1) imports method for record management component in the main (*main.js*) module.

```
(1) import { createRecordManagementUI } from "./auxilary.js";
```

The method itself is consisted of program sections defining each of those DOM elements (input) with the specific functionality. Below in (2), an excerpt of the that method (createRecordManagementUI) is presented.

```
(2) export function createRecordManagementUI(containerId){
        const inputFirst = document.createElement("input");
        inputFirst.id = navButtonSettingsArr[0][0];
        inputFirst.type = navButtonSettingsArr[0][1];
        inputFirst.value = navButtonSettingsArr[0][2];
        inputFirst.title = navButtonSettingsArr[0][3];
        inputFirst.style.width=navButtonSettingsArr[0][4];
        inputFirst.onclick = navButtonSettingsArr[0][5];
```

The *navButtonSettingsArr* variable provides ability to the user to specify a set of elements traits before the runtime process. It is an array of array customized type, defining specifications as *id*, *type*, *value*, *title*, *width*, and *onclick*. The *title* and *width* are not mandatory, but the developer should consider component layout. The code line (3) gives the developer independence of the bound operation e.g. outside of the CRUD domain (but this should be used with purpose, justified and tested).

```
(3) inputFirst.onclick = navButtonSettingsArr[0][5];
```

The UI component behavior was tested via browsers as requires in Table 1.

4. Conclusion

The proposed approach forms holistic approach for data record visualization, user information acquisition, and its processing as well. The first its importance is comprising of the full database domain operations i.e. CRUD. As a second importance of proposed approach is independence of the record management UI component from the any of used methods by the developer in its main module. As a future task an optimization of the program logic of the auxiliary module is foresees.

Acknowledgments

This scientific article is a part of Web-based learning application research.

References:

[1] Allen, E., A., Erhardt, E., B., Calhoun, V., D., Data visualization in the neurosciences: overcoming the curse of dimensionality, Neuron, Volume 74, Issue 4, 2012, pp: 603–608, ISSN:1097-4199 , doi:10.1016/j.neuron.2012.05.001.

[2] Li, Q., Embodying Data, Springer, 2021, pp: 17-47, ISSN:978-981-15-5069-0 , doi: 10.1007/978-981-15-5069-0_2.

[3] Yi, M., Wang, Y., Tian, X., Xia, H.,User Experience of the Mobile Terminal Customization System: The Influence of Interface Design and Educational Background on Personalized Customization, Sensors (Basel), Volume 21, Issue 7, 2021, EISSN 1424-8220,doi: 10.3390/s21072428.

[4] Song, N., He, X., Kuang, Y., Research hotspots and trends analysis of user experience: Knowledge maps visualization and theoretical framework construction, Frontiers in Psychology, Volume 13, 2022, doi: 10.3389/fpsyg.2022.990663.

[5] Popgeorgiev, A., Ibryamova, E., Ivanova, G., Microlearning-Inspired Approach for Teaching Business Modeling to Computer Engineering Students, In Proceedings of the International Conference on Computer Systems and Technologies, Association for Computing Machinery, 2024, pp. 185-191, ISBN: 979-8-4007-1684-3.

[6] McFarland, D., S., Dreamweaver CS5: The Missing Manual, O'Reilly Media, Inc, 2010, ISBN: 978-1-449-38181-3.

[7] Standard ECMA-262: ECMAScript® 2024 Language Specification, Ecma International, 15th edition, 2024, (https://ecma-international.org/wp-content/uploads/ECMA-262_15th_edition_june_2024.pdf) (last visited at 05.11.2024).

[8] Most used programming languages among developers worldwide as of 2024, Statista Inc., (https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/) (last visited at 05.11.2024).

[9] Display database records, 2024, Dreamweaver, (https://helpx.adobe.com/dreamweaver/using/displaying-database-records.html) (last visited at 05.11.2024).

[10] Working with data in JavaScript, (https://www.lianja.com/doc/index.php/Working_with_data_in_JavaScript) (last visited at 05.11.2024).

[11] Hartman, J., 2024,How to Display SQL Data in HTML, (https://www.htmltables.io/blog/how-to-display-sql-data-in-html) (last visited at 05.11.2024).
[12] MDN Web Docs: CRUD, 2023, (https://developer.mozilla.org/en-US/docs/Glossary/CRUD) (last visited at 05.11.2024).
[13] Microsoft Ignite: CRUD (Create, Read, Update, Delete), 2019, (https://learn.microsoft.com/en-us/iis-administration/api/crud) (last visited at 05.11.2024).
[14] Oracle CRM On Demand JavaScript API Developer's Guide: Methods for CRUD Operations, Oracle Help Center (https://docs.oracle.com/cd/F47663_02/books/OnDemJavaDev/c-Methods-for-CRUD-Operations-akf1025611.html) (last visited at 05.11.2024).
[15] MDN Web Docs: Fetching data from the server, Mozilla Fondation, 2024, (https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Fetching_data) (last visited at 05.11.2024).
[16] WampServer, a Windows web development environment, (https://www.wampserver.com/en/) (last visited at 06.11.2024).