



UDC 519.85:004.42

OVERVIEW OF PROBLEMS IN TEXT-DATA PROCESSING AND CREATING A CLIENT-SERVER APPLICATION DEVOTED TO REFERENCES MANAGERS

A.Ivanova¹, K.Kharash², O.Olshevska³, Y. Bortsova⁴^{1,2,3,4}Odessa National Academy of Food Technologies, Odessa, UkraineORCID: ³0000-0002-4512-3915, ⁴0000-0001-6712-8357Scopus ID: ³57192687506E-mail: ivanovaanast29@gmail.com, xkharash@gmail.com, olshevska.olga@gmail.com, bortsova.07@gmail.com

Copyright © 2018 by author and the journal "Automation technologies and business - processes.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>

DOI: 10.15673/atbp.v10i2.971

Applications that provide online services are becoming more popular every day and a part of desktop apps respectively decrease. The search of information, processing documents, creating images and even user games find their place on the Internet. All these things are built on basis of the client-server data transferring model which nowadays is main for most of modern applications starting directly from the web segment ending with mobile devices.

The problem should be talk about is low level of using online applications culture in favor of familiar desktop versions of data processing applications. In particular it's about text-processing tools. Despite the part of web applications increase daily, the survey indicates that most part of researchers still prefer offline redactors for their work. Unfortunately this approach does not ensure user's data from corruption, it doesn't provide an opportunity to cooperate for joint work for researchers and bears lots of other risks which can be reduced by using online tools.

Considering all the above the main topic of the article is the use of client-server model in the development of referative managers and some work-processes automatization. The client-server architecture, actual problems of science works processing sphere and some tools for text-processing are considered in the article.

ONAFТ Reference Manager project based on client-server architecture and developed to eliminate existing problems of the sphere will be mentioned as well.

Key words: ORM, client-server, HTML, CSS, python, MVC

Щодня все більш популярними стають додатки, здатні надавати користувачеві послуги в режимі онлайн, а доля додатків на ПК, відповідно, зменшується. Пошук інформації, оброблення документів, створення зображень та навіть користувацькі ігри знаходять своє місце в мережі Інтернет. Все це побудовано на основі клієнт-серверної моделі передачі даних. Така модель, наразі, є базовою в архітектурі більшості сучасних додатків, починаючи від веб-сегмента і закінчуючи мобільними пристроями.

Проблема, про яку варто сказати - це низька культура використання веб-засобів для роботи і надання переваги звичним десктоп-додаткам для оброблення інформації. Конкретно, мова йтиме про засоби обробки текстових документів. Незважаючи на щоденне збільшення долі додатків у мережі, згідно з опитуванням, більшість дослідників все ще надає перевагу офлайн-редакторам. Нажаль, такий підхід не гарантує збереження даних у разі збою, не дає можливості дослідникам кооперуватися та несе з собою ще багато ризиків, зменшити які дозволяють редактори онлайн.

Враховуючи все сказане вище, головною темою цієї статті є використання клієнт-серверної моделі в створенні реферативних менеджерів і автоматизації виробничих процесів. У цій статті розглядається архітектура клієнт-сервер, актуальні проблеми в галузі оформлення наукових доробків та проводиться огляд деяких засобів для автоматизації форматування текстових документів.

Також, у статті буде розглянуто проект ONAFТ Reference Manager, побудований на основі клієнт-серверної архітектури і розроблений, безпосередньо, для усунення існуючих проблем галузі.



Ключові слова: ORM, client-server, HTML, CSS, python, MVC

1. Introduction

Informational technologies playing the role in execution of scientific works with the help of electronic sources. Thankful to informational technologies saving, processing and representing of information is fully automated. There are some advances automated IT provide in processing scientific works. These are electronic sources with built-in text redactors, the opportunity to find scientific publications in global web, documents' export into preferred format and free access to scientific works.

The automated scientific works-processing system is an electronic source that permits the sciencemetrics-specialists open access to some tools and applications to process the documentation in their sphere. Main steps in developing the platform for documents processing are determining structure aims and State Standard for preparing and formatting documents, creating the tool-kit, creating a pattern for publications edition and developing of user-friendly interface. Besides there are steps in developing web-application at all that can not be avoided. These are analysis of problem sphere, creating the site prototype, developing the main page and if necessary the other pages design, programming the functionality, testing the product in different environments and browsers. In detail Figure 1 represents these steps.

Main references systems of that kind are Mendeley, Citation Machine, MikTEX and Edanz Author-Path. And almost all of them can not be used in Ukraine for several reasons. Main reason is the orientation of that systems on English-speaking users and West State Standards, which are not applicable in Ukraine.

To solve the problem the decision is to create some local ukrainian source which can provide essential functions and avoid all the problems existing in analogous software.

The client-server model is, at the moment, the main architectural solution when creating any web application. Such an architectural solution, however, complicates the development process. The software developer needs to consider more features of the work and interaction of different components among themselves.

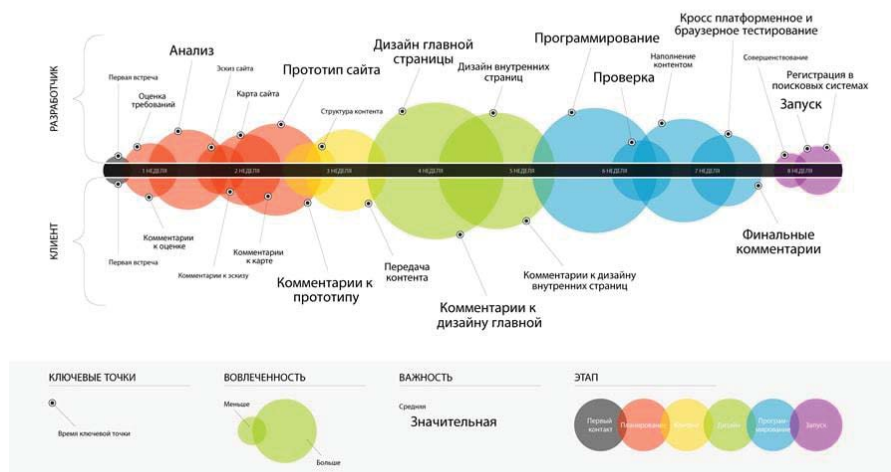


Figure 1. Web-application development steps [8]

The main idea of the client-server architecture is to divide the application into indivisible parts in order to distribute capacities between several hardware units to increase the total capacity of the application - it was also implemented in the above-mentioned project. The external API-interface transfers information to the client-independent application via REST-requests. The application backend by it's turn provides such an API which is a kind of layer between the database and external aggregation systems. To provide the necessary level of security in the developed application, a certain bundle is used: the organization of sessions between the client and the server and the system of web tokens that confirm the current session.

2. Formulation of problem

The main goal of this work is to create software based on a client-server model that will provide an opportunity for users to make scientific works according to the state standards of Ukraine, while avoiding the existing shortcomings of analogous programs.

To achieve the goal the following tasks were set and solved:

- analyzing problems of the subject area
- analyzing existing analogues



- analyzing development tools
- developing a software product that corresponds all the set requirements

3. Materials and Methods

To study the problem, a systematic approach was applied, an object-oriented approach was used to build the information model of the system, a relational approach was used to create the database using ORM technologies.

To create a web application, the following server-side technologies were used: PostgreSQL - a free open source database server which is the best solution for creating relational databases; PyCharm - as a development environment for the Python language - was used to develop the server part of the application; Php Storm - as a development environment for the TypeScript language used to develop the client part of the application. The business logic of the application is built using the Python-framework Flask. The client part was developed with a set of tools for creating a high-quality modern Internet resource. The process of creating a web application is rather time-consuming. To create the project technologies such as the bundle of HTML5, CSS3, the Angular 5 and Bootstrap frameworks, as well as the JQuery library were chosen.

4. Main part

The ONAFT Reference Manager project was implemented as a RESTful API. The main idea of this approach is to build a distributed information system by publishing abstract resources, issuing identifiers to them, identifying a number of simple operations over them, and linking these resources with links. The essence of this approach is that the transfer of the application state occurs via the HTTP protocol using methods, the main ones of which are: GET, POST, PUT, DELETE. Such requests coming from the client part uniquely tell the server what the user wants to do.

4.1 Server side tools

Flask Framework is a platform for Python that provides the ability to develop web applications in Python. This is a microframework, that is, an application framework that allows to implement basic functions. If it is necessary to expand the application, third-party libraries can be used as if they are supported by the framework itself.

Flask is licensed under the terms of the BSD (Berkeley Software Distribution License) license, is a free open source software which may be freely altered by the community.

The Python language, with its appearance, quickly spread to the software developers community. This is due to its versatility, including Python is suitable for writing Web servers. However, for a long time, many Python-based web applications users were limited in choosing web servers that could be used with applications. Usually, developers are limited to support one-way connection to a web server. Some applications could use the CGI (Common Gateway Interface) or Fast CGI protocol, others were tied to the Apache mod_python module, and some applications were generally able to support an API that is specific to one server only. Such situation slowed down the spread of web applications written in Python. For this reason the Web Server Gateway Interface WSGI appeared.

The main goal of the protocol development was the choice of frameworks for the development of web-applications from the selection of web-servers. This allows developers to focus on their tasks and distinguishes WSGI from other communication protocols with the server.

In terms of WSGI, the web-application is divided into two parts: a server (gateway) and a direct application (a framework for creating applications). To access the application, the server part uses an object call (function, method, class, or instance of the class with the `__call__` method). Also, WSGI allows you to create intermediary applications that are an application to a web server and server for a web application. Such intermediaries can be used to pre-process requests for the application or to further process it replies. Thus, the server invokes the application for each HTTP-request assigned to it. [1]

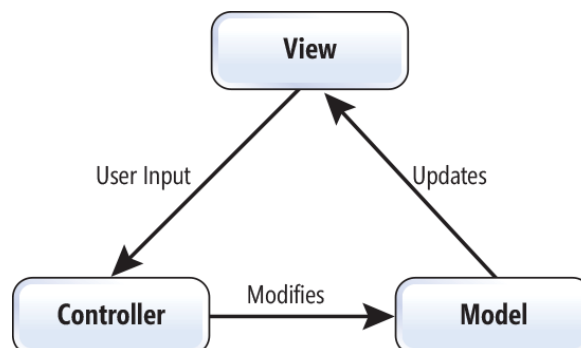


Figure 2. MVC programming-pattern.[9]



There is a Werkzeug Python library to support the WSGI protocol. This library is easy to use, since it provides the ability to test the created application in a web browser window, simplifies the use of request objects (request / response objects), automatically restarts the server, with changes in the code of the application.

The server for this application is implemented using the MVC (Model-View-Controller) model. That is all application management is divided into three parts: model, presentation, controller.

- Model - objects that represent data and react to class-controller commands by changing their state.
- View - Interface and display of data according to model state changes.
- Controller - business logic of the product; user interpreter.

The main purpose of using this approach is to differentiate the business logic of the application and its visual display. Using this pattern increases the versatility and transparency of the code.

An example of a user account table is an illustration of the work of this pattern. To begin with, create a data model that is associated with the representation of the essence in the database used. Here you can specify the name of the entity, the fields and their data type, assign the primary and secondary keys, and set the values of some of the default settings.

The main feature of the project is an ORM technology. The technology is the most widespread and the most rational solution for organizing databases for work inside applications at the moment.



Figure 3. ORM working principles

ORM is a programming technology that links the essence of a database to the concepts of object-oriented programming or transfers the work of a database from the usual sql-queries to the principles of object-oriented programming languages. Thus, the database entities when developing are ordinary classes in programming, which makes it much easier to work with them. ORM creates a virtual object layer between the database and the application, which allows to work with the entities of the database as with objects in the classical sense. When using this technology there is no need to make requests, account the peculiarities of the dialects or various database management systems. The code becomes universal, and it does not matter which storage for an application data will be used. If it is necessary to make some changes, that can be made quickly using only the programming skills, and further conversion of the code into the sql-requests completely falls on the ORM client. Also if necessary, the database can be quickly transferred to another DBMS without loss of performance and without any significant changes to the software code.

Of course, it's worth noting that the relational model is much more detailed than the object model, but this is the advantage of the object-oriented approach. That can be explained by the fact that several tables can be used to store one object in the relational database which the object model allows to avoid.

Relational database management systems do not have inheritance - a completely native paradigm for programming languages. The objects can be compared not only by comparing the values of their separated fields, like the purely relational approach, but the comparison operator may be applied to objects directly.

Using the ORM technology in this application eliminates the developer's need to work with sql-requests directly which can lead to errors in the code.

In the given project the SQLAlchemy - an ORM-framework provides the database-add-on connection.

SQLAlchemy is a library for the Python programming language, designed to solve object-relational mappings. This library allows to establish the association between user-created Python-classes and the tables in the database. The entities of these classes (objects) associate with the rows in the corresponding tables. The specified library includes a system that explicitly synchronizes all changes in the state between objects and their corresponding table rows, and, in addition, expresses queries to the database according to the rules created by the user classes and the links between them. [2] SQLAlchemy is implemented under the MIT license and is a free open source software.

A feature of the SQLAlchemy functioning approach is that the library represents each layer of database interaction as a multifunction API, which, for its part, divides tasks into two categories - Core and ORM. The Core implements interaction with the database through the Python Database API (DBAPI), executing sql requests and managing the data schema. Object-relational mapping is an adjunct over the kernel and is one of many possible layers of object abstraction. The described approach is intended to draw the most benefit from each area of operation. The library allows to reduce the number of steps of



performing trivial tasks, and at the same time maximize database performance, as a result, it is possible to execute thousands of requests in one call of the Python method. [3]

4.2 Client side technologies.

HTML is a hypertext markup language for creating web pages. This language is interpreted by the browser and displayed as a document in a convenient and understandable by human eye form. HTML is an integral part and basis of virtually any web page. It is HTML that gives the content a certain sense, and it is realized by specially targeted tags.

Tags are special markers that browser interprets in a special way. Each system tag is responsible for a certain direction of design and organization of contents for further interpretation by browser. That is, the title tag is responsible for designating the title, paragraph for paragraph, the content of the tag for attaching the multimedia, interpreted by the browser, as a picture, or video, or audio track, and so on. Consequently, each HTML tag intended to mark text information provides some kind of meaning to the given information.

The browser builds the structure of the generated HTML-document and displays it according to the instructions given to the file, named cascading style sheets and scripts. If the text markup of the document is built correctly, the browser window displays an HTML page that contains all the written items.

Each HTML document follows a certain structure for correct interpretation by the browser. All basic rules that are followed by the HTML language contain in a file type of the document that represents the XML document. The document defines what tags, attributes, and their values correspond to a specific version of an HTML document.

For many years, HTML has evolved and still undergoes some changes. Nowadays the most current version is called HTML5. First of all, HTML5 is a hypertext markup language, but unlike HTML and XHTML, is acquired a large amount of features and fixed the main disadvantages.

The main benefits of the HTML5 language are the ability to make the code cleaner, and the document structure is much better. The primary container of the previous version is completely replaced by the semantic elements represented by HTML5. (Figure 2) In addition, the new version of the development of multimedia, now it is possible to listen to music directly in the browser, with the ease of working with photos and images, with no help of third-party programs. The features of the new version contain new field-types and automatic contents of the field checking. [4]

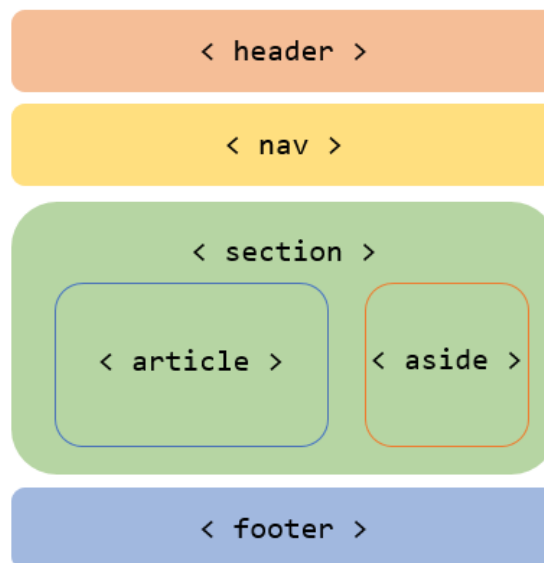


Figure 4. The semantic model of creating an HTML-page.

Typically, web pages are created in conjunction with hypertext markup language and cascading style sheets. Using the appropriate combination improves the appearance of the web page and provides necessary interactivity within.

CSS (Cascading Style Sheets) are used to describe the appearance of a document written in a markup language. Typically, CSS is used to create and modify the style of elements of a web page written in HTML and XHTML languages.

The main purpose of creating the CSS was separating the logical structure of the web page description and the appearance. Hence, the HTML language describes the logical structure of the web page, and CSS describes the appearance of this structure. Such separation provides much more flexible appearance of the document control and avoids the code parts duplication. CSS makes setting the web page, its individual elements in dimensions, colors, fields and indents, and other changes in appearance possible.



It is important to understand that a CSS file is a plain text file that contains special instructions for describing the appearance of an item and its positioning on a page. All stylistic rules are represented in the form of "property: value" and applied to the selectors. [5]

CSS3 is the latest version of the cascading style sheet and expands and adds new features to the appearance of the web page. The main benefits are the combination of background images and colors, adding shadows and transparency to blocks, new features for editing text etc. HTML and CSS is the basis of every web-site. Regardless the server or client part development direction, this kind of connection understanding and using is mandatory. The given technology provides the opportunity in creating modern, adaptive and interactive web-sources.

Associated technologies such as JQuery library and the Angular 5 framework are used to manipulate the elements of a web page and provide visual effects to them.

JQuery is a library that specializes in JavaScript and HTML interactions. The library is designed to work with HTML documents and separates behavior from the structure of a web page. At present, the given technology is very common in different circles of specialists. The given library is a free software licensed by MIT. The jQuery syntax is designed to make navigation more convenient by choosing DOM elements, creating animations, handling events, and developing AJAX applications. JQuery also provides developers with the ability to create plugins at the top of the JavaScript library. Using these objects, developers can create abstractions for low-level interaction and create animations for high-level effects. It facilitates the powerful and dynamic web pages creation.

The main task of jQuery is to provide a lightweight and flexible tool for browser-addressing DOM objects using selectors. The principle of jQuery is to use the class (function), which, when appealing, returns itself. So, it provides the ability to build a consistent chain of methods. [6]

Angular is a framework from Google for creating Internet resources, primarily aimed at creating a one-page online resource. By that cause it is the successor to the other AngularJS framework. Worth to be said that Angular is not a new version of AngularJS, but a completely new framework. The framework has such a function as a two-way link, which provides dynamic change the data in one place of the client part when changing the model data in the other part, templates, routing, and the like. Unlike AngularJS, Angular uses the TypeScript programming language. [7]

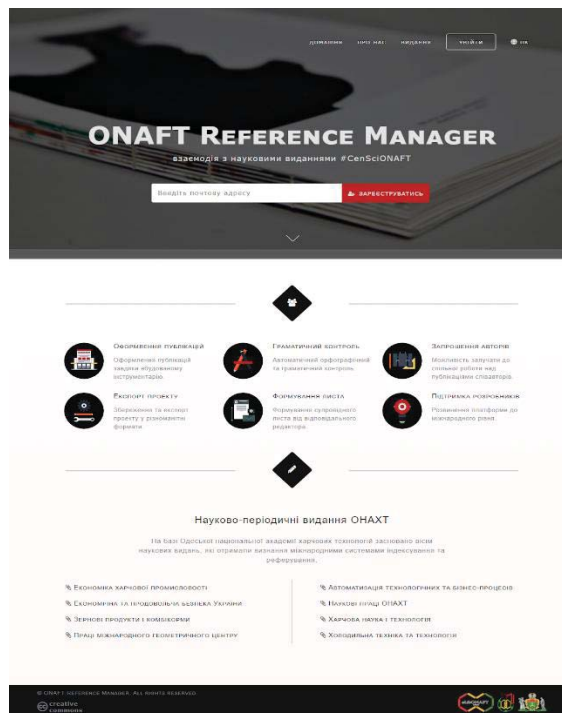


Figure 5. ONAFT Reference manager homepage.

5. Practical part

All the principles described in this article have been applied to the development of the ONAFT Reference Manager project. The resource at the moment is fully Ukrainian-speaking.

The first version of the application was released in 2018. In the future, it is planned to translate the resource into English and Russian and add a list of standards for processing documentation. It is also planned to realize the possibility of authorization through other scientific resources, such as Scopus, Researcher, Orch.



The main feature of the project is ability to switch on or off the headers while making a publication if necessary. Every publisher as every state has its own standards for any kind of publication. Reference Manager ONAFT provides the Ukrainian standards for scientific articles.

Also the resource provides saving the formatted document in a popular format, searching within document, inserting indexes and some special symbols. All this is illustrated on Fig. 6.

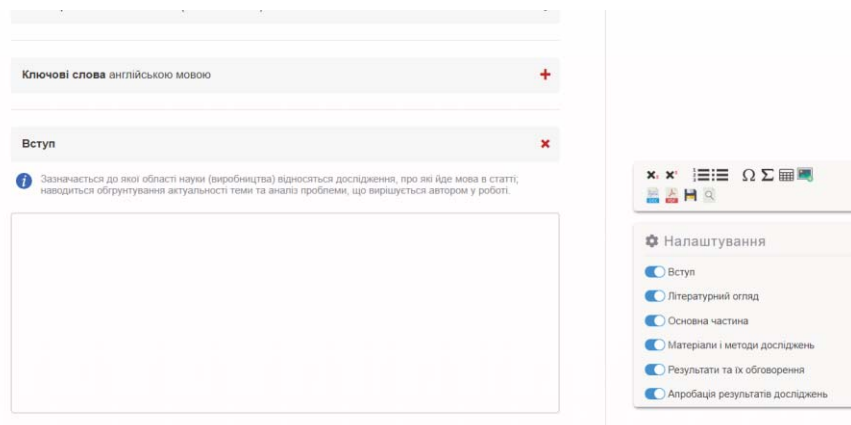


Figure 6. Headers selection example.

Results

The result of the development was a fully functional web application, primarily aimed at scientists of Odessa National Academy of Food Technologies and Ukrainian state standards. First and foremost, the project is designed to solve problems in the design of scientific publications, minimize routine activities during the preparation of scientific papers for publication and reduce the mistakes in process.

References

1. "WSGI - протокол зв'язи Web-сервера с Python приложением." [Online]. Available: <http://hllabs.org/development/python/wsgi.html>. [Accessed: 21-Jun-2018].
2. "SQLAlchemy 1.2 Documentation," *Object Relational Tutorial - SQLAlchemy 1.2 Documentation*. [Online]. Available: <https://docs.sqlalchemy.org/en/latest/orm/tutorial.html>. [Accessed: 21-Jun-2018].
3. "SQLAlchemy," *The Architecture of Open Source Applications (Volume 2): SQLAlchemy*. [Online]. Available: <http://aosabook.org/en/sqlalchemy.html>. [Accessed: 21-Jun-2018].
4. "HTML, CSS, PHP, JavaScript, SQL - что и зачем?," *На главную*. [Online]. Available: <http://codeharmony.ru/materials/125>. [Accessed: 21-Jun-2018].
5. "SEO Direction," *Язык HTML 5 - преимущества и недостатки*. [Online]. Available: <https://seodirection.ru/html5/>. [Accessed: 21-Jun-2018].
6. "Робота #7," #7. *JavaScript та бібліотека jQuery UI*. [Online]. Available: <http://www.victoria.lviv.ua/library/students/wd/work7.html>. [Accessed: 21-Jun-2018].
7. "Введение в Angular." [Online]. Available: <https://metanit.com/web/angular2/1.1.php>. [Accessed: 21-Jun-2018].
8. Figure 1. Web-application development steps. [<http://webworking.by/web-technology/infografika/infografika-na-temu-etapy-sozdanie-sayta/>]
9. Figure 2. MVC programming pattern [<https://www.pinterest.com/pin/284289795201685213/>]

Список використаних джерел

1. WSGI - протокол зв'язи Web-сервера с Python приложением [Електронний ресурс] - Режим доступу до ресурсу: <http://hllabs.org/development/python/wsgi.html>.
2. Object Relational Tutorial [Електронний ресурс] - Режим доступу до ресурсу: <https://docs.sqlalchemy.org/en/latest/orm/tutorial.html>.
3. The Architecture of Open Source Applications (Volume 2) SQLAlchemy [Електронний ресурс] - Режим доступу до ресурсу: <http://aosabook.org/en/sqlalchemy.html>.
4. Науменко Д. HTML, CSS, PHP, JavaScript, SQL - что и зачем? [Електронний ресурс] / Дмитрий Науменко - Режим доступу до ресурсу: <http://codeharmony.ru/materials/125>.
5. Язык HTML 5 - преимущества и недостатки [Електронний ресурс] - Режим доступу до ресурсу: <https://seodirection.ru/html5/>.



6. JavaScript та бібліотека jQuery UI [Електронний ресурс] - Режим доступу до ресурсу: <http://www.victoria.lviv.ua/library/students/wd/work7.html>.
7. Введение в Angular [Електронний ресурс] - Режим доступу до ресурсу: <https://metanit.com/web/angular2/1.1.php>
8. Figure 1. Web-application development steps. [<http://webworking.by/web-technology/infografika/infografika-na-temu-etapy-sozdanie-sayta/>]
9. Figure 2. MVC programming pattern [<https://www.pinterest.com/pin/284289795201685213/>]

УДК 664.1: 65.011.56

МОДЕРНІЗАЦІЯ СТРУКТУРИ СИСТЕМИ КЕРУВАННЯ ВИПАРНОЇ СТАНЦІЇ БУРЯКОЦУКРОВОГО ЗАВОДУ НА БАЗІ МІКРОПРОЦЕСОРНИХ ТЕХНІЧНИХ ЗАСОБІВ І ПРОГРАМ УКРАЇНСЬКОГО ВИРОБНИЦТВА

Скаковський Ю. М.¹, Бабков А. В.², Мандро О. Ю.³

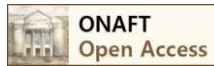
^{1,2,3}Одеська національна академія харчових технологій, Одеса

ORCID: ¹[0000-0003-4888-4469](https://orcid.org/0000-0003-4888-4469); ²[0000-0002-9167-7804](https://orcid.org/0000-0002-9167-7804)

Copyright © 2018 by author and the journal "Automation technologies and business - processes.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>



DOI: 10.15673/atbp.v10i2.972

Анотація: Розглядаються технічні рішення з модернізації структури комплексу технічних засобів керуючої системи випарної станції цукрового заводу. Система керування побудована як автоматизоване робоче місце (АРМ) оператора на базі комп'ютера, мережі мікропроцесорних контролерів та регуляторів, датчиків та виконавчих механізмів. Мережа мікропроцесорних контролерів та регуляторів реалізована на базі перетворювача MODBUS RTU – USB типу БПІ-452. АРМ включає також щит оператора для реалізації дистанційного режиму керування. Фронтальна сторона щита складається з двох панелей, де розташовані мікропроцесорні контролери та регулятори, а також прилади, що показують, та органи дистанційного керування (задавачі). На внутрішніх панелях щита розташована група мікропроцесорних регуляторів та приладів системи живлення. Система керування спроектована з використанням мікропроцесорних контролерів, приладів і програм Українського виробництва, датчиків та виконавчих механізмів імпортного виробництва. Імітаційне моделювання проводилося на спеціалізованому стенді із застосуванням промислових контролерів і програм. Результати досліджень застосовані для модернізації керуючої системи цукрового заводу.

Abstract: Technical solutions for modernization of technical facilities structure for control system of sugar plant evaporator station are considered. Control system is built as an automated workstation (AWS) of the operator based on the computer, microprocessor controllers and regulators network, sensors and actuators. The network of microprocessor controllers and regulators is launched on the basis of MODBUS RTU-USB converter BIP-452 type. Workstation also includes operator panel to implement remote control mode. Shield's front side consists of two panels within located microprocessor controllers and regulators and remote controls (reference-input element). Group of microprocessor regulators and power supply devices are located on the internal panels. The control system is designed using microprocessor controllers, devices and programs of Ukrainian production, sensors and actuators of imported production. Simulation modeling was conducted on specialized stand with the use of industrial controllers and programs. The results of research were used to modernize the control system of the sugar factory.