

Method For Complex Web Applications Design

Rustam A. Valiev, Lenar A. Galiullin, Irina S. Dmitrieva, Aleksey N. Ilyukhin

*Kazan Federal University, Russian Federation, the Republic of Tatarstan,
the city of Kazan, Kremlevskaya ul, 18.*

Abstract

This article deals with peculiarities of design that are taken into consideration during practical development of complex information systems, made on the basis of web technologies, in particular during development of web applications. This article covers the majority of development aspects from the position of different methodologies of web applications design and lists design peculiarities differences in comparison with programme means that do not use web technologies.

Web application architecture design according to block and modular principle is suggested. The main programme block will be responsible for users' authorization, session provision, request of enclosed modules, connection to database and similar service functions. Administrative module will allow managing sets of modules, features of web applications, change informational content. It is also suggested that services of web application are designed according to modular principle as the ones that have their own managing modules and modules of data processing (working modules). Peculiarity of the suggested methodology is that the modules are made in the form of separate files that execute their own specific function and work strictly with their own tables in database.

It is suggested to use traditional approach when creating a web application. It is connected with presence of a big number of different relational database control systems supported by web servers.

Initially on the stage of database design datalogical models are being developed. They are oriented on data storage and processing environment.

The difference of the suggested graphic interface is that navigation in the web application is managed by the user who may execute previously permitted actions on random following his way in navigating the interface.

The method described in the article is used in web applications development for automated informational systems of diesel engines trials.

Keywords: web applications, client-server technology, heterogeneous data storages, structural design, informational architecture, web programming, scripting languages, and web interface.

1. Introduction

Web applications are complex programme systems that have the following peculiarities:

- High interactivity;
- Client-server technology;
- Usage of data storages that have different structures;
- Open architecture.

High interactivity is the direct communication of the user with information computation system that is characterized as a request or dialogue. Web applications, to start with, are meant for data storage access of the end customer through user interface. Due to this fact web applications are to have simple, convenient and clear interface, oriented on the usage of the users that have different level of computer literacy. Difference from the traditional graphic interface is that navigation in web application is managed by the user who can execute preliminary granted actions on random following his own way of navigating documentation.

2. Selection of web application architecture.

Term “client-server” means the technology of contact between user’s browser and web server. Client requests’ processing is made on the server. The client gets a created document that contains the processing result of data inputted by client. Technologies from the side of server may vary. They depend only on the software and general web application architecture used by server.

Usage of heterogeneous data storages is the basis of web application as an automated informational system. Usage of data storages allows excluding reciprocal influence of data and code of web application. Open architecture of web application allows the designer to add and change modules in the ready system without the necessity to re-make core programme code. Openness of the system improves its flexibility and usability for creation of applications of different functionality and purpose.

Applicability of structural design method in the case of web application calls for more thorough approach as due to the specific of the developed programme product one has to select both methods of structuring – functional oriented method (successive division of the task or the whole problem into separate, simple enough parts that have functional distinctness) and method of data structuring. Term

“informational architecture” is introduced in web programming in respect to the methods of data structuring.

Informational architecture — the process of identification, organization and systematization of web application content. In the first place this is true for status hypertext data models. In the technology “client-server” and for web application of general purpose creation where informational architecture is determined by the end user of web application this term has limited usage.

During web application development the designer has the following tasks:

- Selection of generic solutions that correspond to the functions most frequently used in these systems;
- Creation of small informational systems with the capability of their catenation and integration in some bigger web applications according to functional pattern;
- Unification approach creation, unified data formats creation and support for the possibility to link between different informational systems of different manufacturers and also development and unification of data formats in informational systems of one manufacturer;
- To use standards for metadescription of informational resources during development of web application for the specific areas of use;
- To foresee possibility to keep heterogeneous metadata inside one information storage for web application with possibility to use by different informational systems where it is not advisable to use all resources of web application;
- To foresee possibility to increase data storages on the basis of modular principle as well as functional modules that correspond to them;
- To foresee application cross-platform and cross-browser compatibility of its interface.

The most popular services of web application are usually selected as a cluster of typical solutions.

As web applications are built according to modular principle the scheme of connections is usually made in the following way:

- The main programme block responsible for users’ authorization, session provision, request of enclosed modules, connection to database and similar service functions;
- Administrative block that allows to manage module sets, web application properties, change informational content;
- The services themselves that are built according to module principle and that have their own control modules and data processing modules (working modules). Web application architecture peculiarity is that the blocks are made

in the form of separate files that execute preliminary indicated function and operate strictly with their own tables in database.

3. Databases control system selection

The issue of database selection that is the basis of web application is the main thing for web application logic in general. At present despite the development of object-oriented and object-relational databases, two approaches for data storages creation received the main development.

1. Usage of relational databases.

Traditional approach which despite the time that passed since relational databases conception development is still the standard for the majority of informational systems. The weak point of databases relational management systems is decomposition of the complex object during adding it to database tables. Usage of object-oriented database is justified only during the selection of object approach to informational system design. Besides means for limited work with objects are present in the majority of relational databases.

2. Usage of XML-technologies.

This approach is used mainly during creation of distributed data storage environments that have different data scheme and use XML-technologies for processing and transformation of structured documents.

During creation of web application traditional approach is used that is connected to a big number of different relational database management systems supported by web servers.

On database design stage initially datalogical models are created that are oriented on data storage and processing environment. Datalogical models have logical and physical levels of representation. Physical level corresponds to data storage organization in computer memory. Data logical level as far as database management systems are concerned is realized in the form:

- Database conceptual model – data integrated structures under database management systems;
- External data models – subset of data structures for applications' realization.

4. Web programming language selection

Used web programming languages is another peculiarity of web application creation. As a rule all web programming is based upon scripting languages.

The main characteristic of scripting languages – dynamic nature that allows interpreting data as programming code and vice versa and also it is easy to learn them.

There are four classes of such languages: command-scripting languages; applied scripting languages; formatting languages; universal scripting languages.

The first class comprises numerous CGI commands interpreters, stack processing languages and the languages for system command frameworks creation. Such languages are orientated not to interactive but to package processing mode.

The peculiarity of second-class scripting languages is their orientation to client part of software. They are used mainly for client part realization. These languages include VBScript, JavaScript and their extensions.

Building in a special code (in the form of separate formatting commands – tags) directly in regular texts is a special feature of formatting languages (or tagged ones). Tags are used for separation of information structure from its content for formatting and setting dynamic behavior for the interactive objects built in the document. HTML language (Hypertext Markup Language) is the main language of information on the web.

Metalanguage XML (eXtensible Markup Language) is the language of transportation and intermediate data storage during data exchange between heterogeneous and distributed systems. On its basis one can carry out complex transformations of documents and textual information and also keep data of relational and hierarchical structure in unified form including the data of settings and components programming. In 2001 a revision of HTML appeared that was named XHTML (Extensible Hypertext Markup Language, where requirements of XML were taken into account.

Universal scripting languages are usually associated with the languages of web programming. The majority of these languages were created before appearance of HTML and the Internet for OS UNIX – such languages as Perl, Tcl and Python. After CGI technology creation Perl became the main processor of client forms. The most famous scripting language that appeared after the Internet creation was PHP (PHP: Hypertext Preprocessor). After its fifth version, the language became object-oriented.

Thus all web applications are realized with the help of scripting languages. On the basis of applied scripting languages and markup languages interfaces and programmes of client side are built. On the basis of universal scripting languages server side programmes are built.

Due to the nature of these languages certain problems appear as was mentioned above, these languages have dynamic nature and interpretation of “data-code” may be an advantage or a disadvantage during modules designing. Scripting languages are mainly typeless and can hinder testing of modules created on them. Also scripting languages are mainly interpretive that is a visible code is executed which improves symbol debugging but it decreases speed and requires web server for automated debugging.

5. Web applications' interface designing

Main differences of web application interface from the interfaces of other programmes are the following:

- Interface is always graphical and uses hypertext;
- Interface is created not as a separate programme module, but as data interpreted by browser;
- Interface must be developed according to the corresponding standards of data provision;
- Layout and efficiency of the interface should not depend upon the user platform and browser;
- Layout of the interface should not depend upon user information presentation hardware.

Web application interface must be graphical. From both engineering and esthetic points of view expectations here are high. As web application is software, the main rules for interface creation may be formulated in the following way:

- Interface should not be overloaded with graphical information;
- Interface for web application should not be designed according to typical solutions for system interfaces;
- Ideally interface does not use technologies that are related to the programmes of client side, such as JavaScript/VBScript, as this factor increases load upon client's computer and is not always correctly shown on different platforms and in various browsers.

The very nature of scripting languages says that programme product created in these languages is an intermediary between the user and the resources of operational system, graphic environment, service machine, not directly interacting with them.

Such work method allows not to depend upon client's programme environment, but at the same time it does not allow to work with databases in real time, receiving direct response.

Standards for data layout are developed on the stage of incoming and outgoing data determination and they are not only data formats but also input-output forms, generated reports, etc., that is they are the external level of interaction with end user. On data work elements design stage norms of representation and limitation and also checking of inputted data are set. Also it is worth to mention another thing that is directly connected to graphical information displaying – coding. Text coding is to be strictly set in the obvious form.

Outward appearance of interface should not depend upon user's platform and browser.

Web interface should work equally or with slight differences in all popular browsers and on all platforms. Due to this during development it is necessary to take

into consideration correspondence of browsers to the standards of W3C information representation. Many browsers do not support standards in full. Correspondence or lack of it along with the viewpoint of independent specialists with detailed description of representation limitations are usually given on browsers home pages. Interface should also be cross-platform that is it should not contain the client side technologies that are supported by only one operational system or browser. Also interface should not ask user to install additional components for correct work and results displaying.

Outward appearance should not depend on user's hardware. Mainly it concerns monitors and their resolution and also colour rendition.

Interface should be correctly displayed with different screen resolution. All control elements should be clearly visible in screen working area. There should be no empty fields and strict borderlines. Elastic design has been popular recently where sizes of all elements of the page are given not in absolute but in relative measure units. In the result working area automatically changes its size depending upon screen resolution.

6. Summary

Taking into consideration the abovementioned points we can summarize that the number of factors that are taken into account during web application development exceeds considerably the number of factors used during development of conventional software due to development complexity and web application realization and affects almost all aspects of modern informational technologies that are connected not only to software but also to hardware peculiarities. At present time during web application interface design previously created and new software is used. Development dynamics shows that this realm of modern informational technologies evolves and develops rapidly.

7. Conclusions

The described method is used during web applications development for automated informational system of diesel engines tests.

Conflict of interest

The author confirms that the given data does not contain any conflict of interest.

References

- [1] Hong, C.J., Luong, L.K., Chark, S.Y. Building automation through web interface (2012) 2012 IEEE Conference on Sustainable Utilization and Development in Engineering and Technology, STUDENT 2012 - Conference Booklet, art. no. 6408423, pp. 299-304.
- [2] Mengalli, N.M., Camas, N.P.V. Interface for interaction and knowledge building on the web: A look at the educational curriculum and the social network of the systematic learning group (2013) Project Management Approaches for Online Learning Design, pp. 291-314.
- [3] Tsay, J.-J., Tsay, C.-W., Lin, S.-H. Web query interface parsing for building web-based metasearch systems (2011) Proceedings - 2011 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT 2011, 3, art. no. 6040876, pp. 347-350.
- [4] Van Kleek, M., Karger, D.R., Schraefe, M.C. Watching through the web: Building personal activity and context-aware interfaces using web activity streams (2009) CEUR Workshop Proceedings, 512, pp. 36-39.
- [5] Chavarriaga, E., Macías, J.A. A model-driven approach to building modern Semantic Web-Based User Interfaces (2009) Advances in Engineering Software, 40 (12), pp. 1329-1334.
- [6] Martinez, O., Botella, F., Fernández-Caballero, A., González, P. Building e-commerce web applications: Agent- and Ontology-based Interface Adaptivity (2005) WEBIST 2005 - 1st International Conference on Web Information Systems and Technologies, Proceedings, pp. 351-354.
- [7] Ba, C., Carrero, M.A., Ferrari, M.H., Musicante, M.A. PEWS: A new language for building web service interfaces (2005) Journal of Universal Computer Science, 11 (7), pp. 1215-1233.
- [8] Chang, G., Hsieh, J.-W., Calixto, P. Components for building desktop-application-like interface in web applications (2005) Lecture Notes in Computer Science, 3399, pp. 960-971.
- [9] Zubkov, E.V., Galiullin, L.A., 2011. Hybrid neural network for the adjustment of fuzzy systems when simulating tests of internal combustion engines. Russian Engineering Research, 31 (5), pp. 439-443.
- [10] Biktimirov, R.L., Valiev, R.A., Galiullin, L.A., Zubkov, E.V., Iljuhin, A.N. Automated test system of diesel engines based on fuzzy neural network (2014) Research Journal of Applied Sciences, 9 (12), pp. 1059-1063.