

# INTERNET AUTOMATED DESIGN SYSTEM OF CONTROL AUTOMATICS OF SCHEMES

*Nikolay Bogdanov<sup>1</sup>, Diana Semjina<sup>2</sup>*

<sup>1</sup>*Riga Technical University  
Transport and Electronic Department*

<sup>2</sup>*Transport and Telecommunication Institute  
Lomonosova Str. 1, Riga, LV-1019, Latvia  
Phone: +371 29727288. E-mail: meistars86@inbox.lv*

**Keywords:** Internet automated design system, expert system, Adobe Flex 3

There are many advantages of this work. Firstly we can faster design automatics of schemes. Secondly the automated design system by means of programming allows making simulation of control automatics of the scheme and seeing an operating efficiency of automatics. And also all automatics of the schemes can be controlled by Internet.

In a basis of application of the automated design system created with use of artificial intellect technologies, in particular expert systems – to keep designer's knowledge, to work with them and to make decisions. As a result users of Internet automated design system have information and help support based on knowledge of more qualified experts about the object of designing is presented on Fig.1. [1]. Such interaction of designing algorithms with expert system allows automating those procedures, which are solved on the basis of designer's knowledge and cannot be presented in the form of algorithms [2].

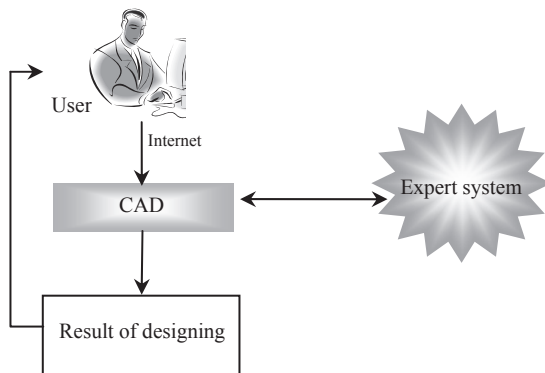


Fig. 1. Automated design system structure

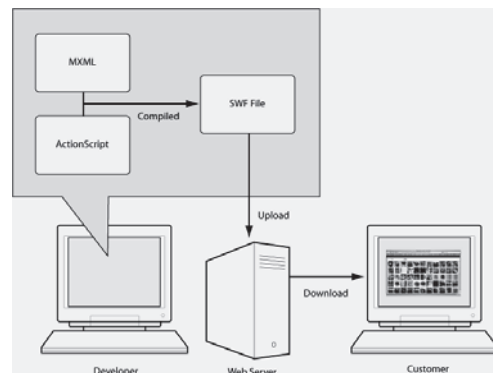


Fig. 2. Adobe Flex3 structure

The major aspect of development of modern automated design systems is their orientation to information integration and interoperability with other systems and software. Unfortunately it imposes certain restrictions of the methodical character, which must be taken into consideration at system engineering [3].

In the designing of control automatics of schemes and control schemes and reports of equipment are indispensable. For this purpose it is necessary to create the automated design system of control schemes and reports of equipment, using Adobe Flex 3, programming language Java, PHP 5 and database Oracle. Adobe Flex 3 is a software development kit released by Adobe Systems for the development and deployment of cross-platform rich Internet applications based on the Adobe Flash platform. Flex applications can be written using Adobe Flex 3 Builder or by using the freely available Flex compiler from Adobe (Fig.2.). Traditional application

programmers found it challenging to adapt to the animation metaphor upon which the Flash Platform has been originally designed. Flex seeks to minimize this problem by providing a workflow and programming model that is familiar to these developers. MXML, an XML-based mark-up language, offers a way to build and lay out graphic user interfaces [4]. MXML is used mainly to declaratively lay out the interface of applications, and can also be used to implement business logic. Interactivity is achieved through the use of ActionScript, the core language of Flash Player that is based on the ECMAScript standard (Fig.3.).

Simple dynamic data-driven applications load external data from a range of sources. The next section of this chapter covers the types of data you’re likely to work with in this sort of application. Dynamic applications also often provide mechanisms for updating the existing data and adding new data. Changing the external data source is enough to update the content in this type of application; there is no need to recompile the SWF file [5].

Adobe Flex 3 data access components use remote procedure calls to interact with server side environment, such as **Java and PHP**, to provide data to Adobe Flex application and send data back to server side resources. In this article we will see how we can use Adobe Flex data access components in our Flex application.

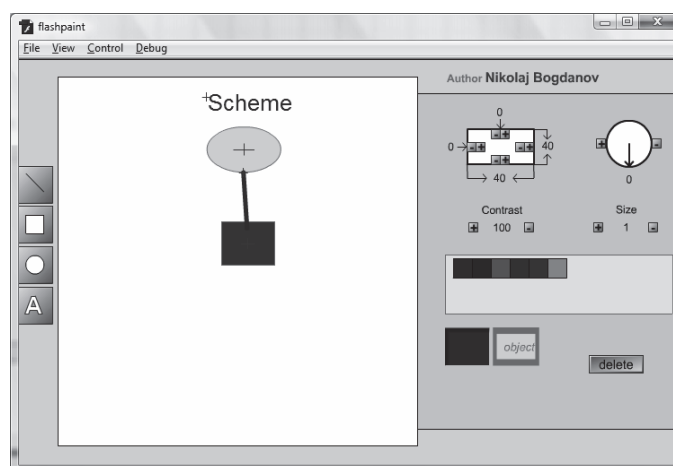


Fig. 3. Flash automated design system

This system of creation of control schemes and reports of equipment will help more quickly and without mistakes to design control schemes and reports of equipment and some users can use this designing system using simultaneously one general database.

## References

1. Kun, Li. *SAPR Basis*. Moscow: Piter, 2004, p.17-96. (In Russian)
2. Norenkov, I.P. *Automated Projecting*. Moscow: MGTU by N.E. Bauman Publishers, 2000, p.10-18. (In Russian)
3. Kandakov, I.A. *Technological Processes of SAPR*. Moscow: Academy, 2007, p. 5-9. (In Russian)
4. Herrington, Jack, Kim, Emily. *Getting started with Flex 3*. Canada: O'Reilly, 2008, p. 23-126.
5. Jacobs, Sas, De Wegghelire, Koen. *Flex for Developer*. New York: Friends of, 2008, p. 9-17.