[5] S. A. Voinova, "Tehnicheskaya gerontologiya. Potentsial vliyaniya na effektivnost funktsionirovaniya tehnicheskih ob'ektov / Mater. za VI Mezhdunarodna nauchna praktichna konferentsiya «Naynovite postizheniya na evropeyskata nauka-2010», 17 - 25 yuni, 2010g. Tom 20 «Tehnologii. Fizicheska kultura i sport».- Sofiya: «Byal GRAD-BG» OOD, 2010. - S. 79 - 84.
[6] S. A. Voinova, Sistemniy pidhid do upravlinnya tehnologichnoyu efektivnistyu tehnichnih ob'ektiv / Fizicheskie i kompyuternyie tehnologii.- Trudyi 17-y Mezhdunarodnoy nauchno-tehnicheskoy konferentsii, 20-21 sentyabrya 2011g.- Harkov: HNPK «FED», 2011.- S.105 - 108.
[7] S. A. Voinova, "Pro aktualnist upravlinnya ekologichnoyu efektivnistyu tehnichnih ob'ektiv," *Energetika ta elektrifikatsiya*, no. 1, pp. 64-67, 2012.

UDC 004.4'244

MULTILINGUAL RECODING METHOD DESIGNED FOR SCADA-SYSTEM'S SOFTWARE UPGRADE

S. Velykodniy[1], O. Tymofieieva [2]

[1]National University "Odessa Maritime Academy", Odessa
[2]Odessa State Environmental University, Odessa
E-mail: [1]dotsent 1981@mail.ru, [2]smsl1985@mail.ru

Abstract: The article discusses a new approach to upgrade the software for SCADA-systems. A distinctive feature of this method is the ability to support more than ten most popular programming languages. By applying this method it's possible to automate the process of recoding software components and, by doing so, it saves time programmers have to spend to rewrite the code and reduce the likelihood of structural errors inherited from the previous system.

Key words: reengineering, recoding, software component, Supervisory Control And Data Acquisition (SCADA-system), Computer-Aided Software Engineering (CASE-means).

Introduction

Supervisory Control And Data Acquisition Systems (SCADA-system or SCADA) has been implemented in various fields of human life but the most widespread occurred in industry and on transportation. And SCADA operation for each mode of transportation (by water, air, rail, road) has fundamental differences.

One common feature for all SCADA is that with time its software become outdated under number of factors such as constant update of information systems, changes in programming languages and distributed data processing systems, etc. Inevitable, such trend leads to speed and graphic deterioration, loss of information, communication and time and / or other features decline until complete system failure.

Defining the problem and its relationship with important scientific or practical tasks

Creating SCADA is quite complicated and time consuming process that can be accomplished by highly skilled and well-coordinated team of developers and coordinators, as only design process included more than ten stages already: research, specification development, technical proposal, preliminary design, technical design, mock-up project, coding, compiling, testing, releasing, and quite often further technical and customer support.

SCADA development is an enormous scientific and technical undertaking which implementation requires significant investment. All existing SCADA and those that are in stage of development using current methodology contain the results of many years of research done by thousands of scientists, engineers, designers and developers who participated in developing

design solutions.

According to current world trends SCADA design should be evolving process. There are at least two good reasons why SCADA should evolve with time. First, the development of such a complex project as SCADA takes long time and from economic point of view it's became obvious that whole system should be broken into smaller units which could be implemented as they become ready (further units will be expanded and combined as necessary). Second, steady progress in object's design, technology, computing and mathematics leads to new mathematical models and methods that should replace older and less successful counterparts.

In this regard, one of the most important features of SCADA should be ease of use and expansion capability by means of adding new and/or improved elements. Here is the question that further addresses this problem – should it be a new development or reengineering.

Recent development and publications analysis where this issue has been addressed

Reengineering is a discipline that incorporates restructuring and reorganization of SCADA, where some units could be redeveloped using modern technology, also allowing modification and upgrade data structure [1]. At the same time architecture of the system might remain unaltered.

Subject of SCADA reengineering were studied in details in [2, 3]. Methodology of reengineering were explained in [4].

From commercial point of view reengineering considered the only method to preserve inherited units while maintaining SCADA. In opinion of leading experts [5, 6] it is not wise to create such a system from scratch not just because lack of time but because all initial time and resources spent on its development will be wasted therefore significantly increasing cost and likelihood to introduce structural defects. At the same time reengineering allows SCADA to evolve, incorporating necessary upgrades aiming to boost quality, streamline maintenance and support [7].

Another point to consider while developing new SCADA is to ensure that new system is consolidated and universal enough [8]. This issue could be related to "beginning of development" which means that most of existing SCADA from various fields could not be considered as up to date, simply because technologies that were used at the time systems were designed become obsolete with time. After 3 – 4 years majority of them do not have the capabilities to sustain current demand for speed, load, image rendering and computation, moreover given that existing SCADA might contained millions lines of code, it could take months or even years (!) to do a code conversion.

The ideal system should utilize as many common units as possible but requirement for units being universal could easily interfere with demand for high efficiency. That's hold true for SCADA.

Current value of the article lies within expectations and benefits for reengineering as a method and lay a course for development of a new units using sequential steps which defined how changes to be introduced, when and how restructuring should occur and also mandatory information for recoding of existing components. As well as terms and requirements for language conversion and support data contingency after restructuring process takes place.

There are several reasons hindering agile implementation of this method such as insufficient documentation explicitly outlining redesign of an existing system and preservation of its architecture.

Define research objectives

It is absolutely necessary to minimize man-hours and expenditure in order to keep SCADA cost effective and the major task is to improve its software.

Fundamental idea discussed here is to develop method of reengineering for SCADA by evolving its software which in turn will ease maintenance, improve support and usability and also facilitate process of system upgrade either through newly developed or redesigned units (modules).

The main objective of this article is to design, evaluate and implement method for redesigning software which will allow creating efficient SCADA across variable fields.

Define approach

In order to achieve multilanguage transition for SCADA software it is necessary to define and follow subsequent steps:

1. Choosing appropriate coding language

Choice of the language for this task will depend on technical requirements or current market trends, and very likely come from top programming languages such as C, C++, C#, Java, PHP, Delphi, Python, Visual Basic etc.

2. Choosing applicable CASE-means

Selection of preferred CASE-means will be dictated by end user preferences. From author's viewpoint the most appropriate CASE-means is Enterprise Architect (EA) which support import to / from and can generate code written by all languages mentioned above. From now on the EA will be used as very efficient tool for recoding.

3. Identification of components

It is necessary to identify which component of SCADA requires recoding. Let's take a look at any open component written in C language for example (pic.1). Let's inspect component "pm-dm.c" (pic.1).

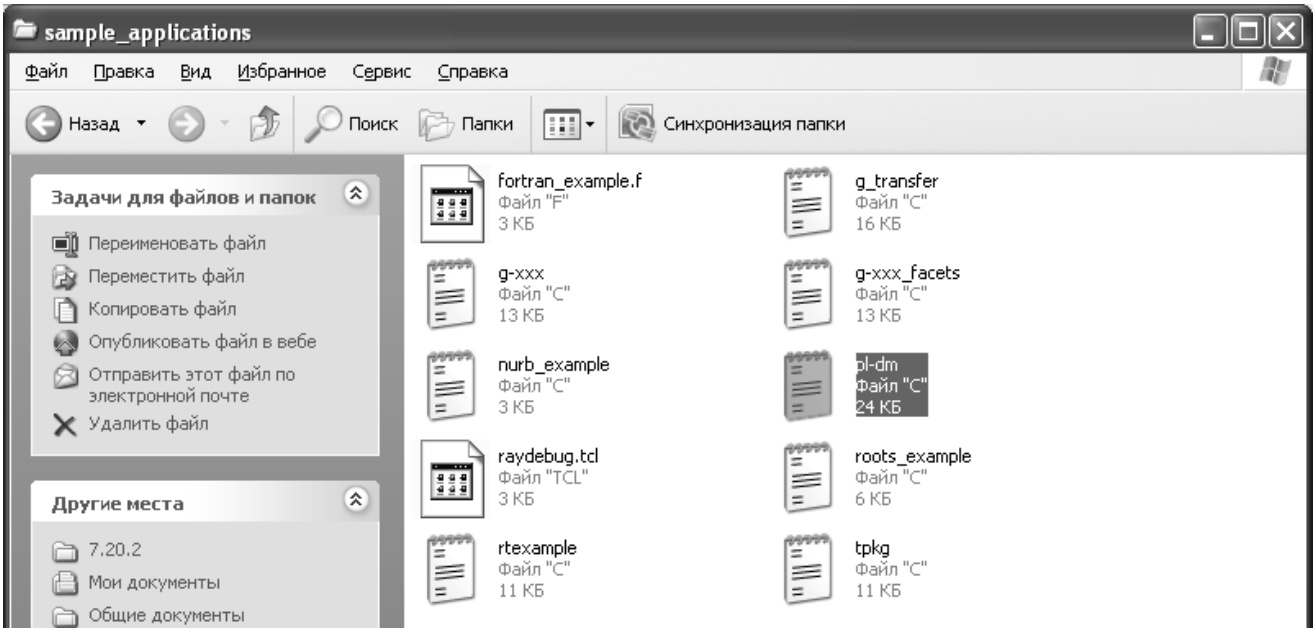

Fig. 1 – Identification of the component written in C language

4. Import of the components

With the help of EA we will be doing import of the selected components. After setting all necessary parameters select following (see pic.2).

a) Tools;

b) Source Code Engineering;

c) Import C files…

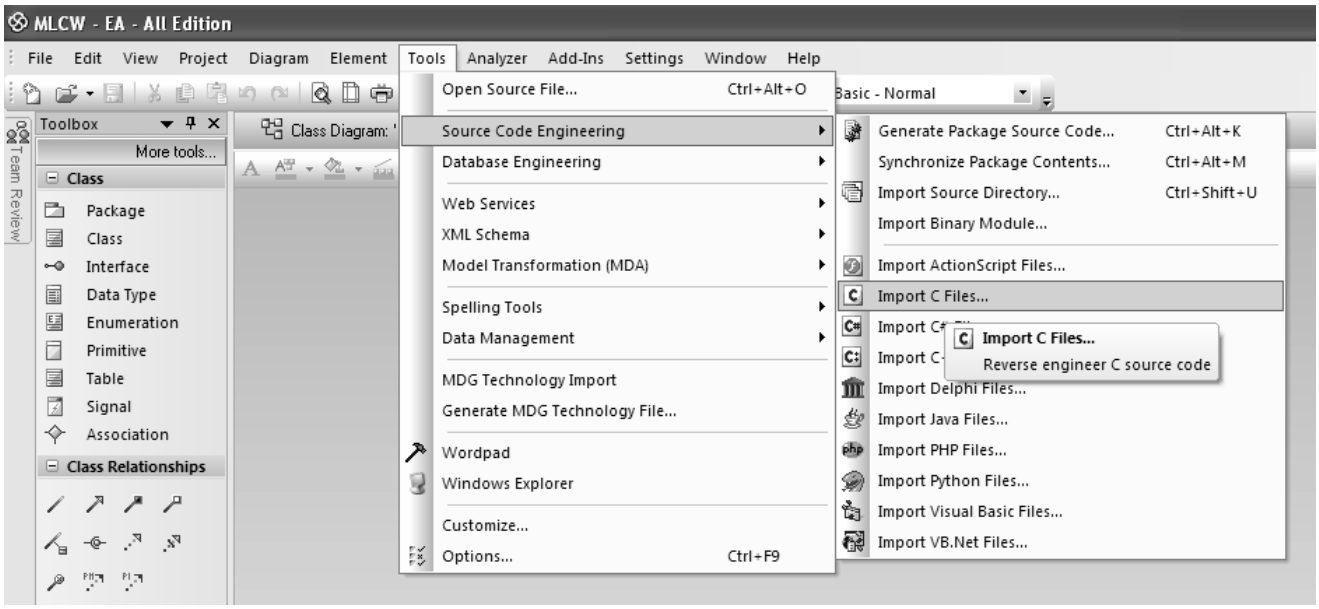

Fig. 2 – Sequence selection for importing components

Any components, written in the language from dropdown list in EA can be imported in the method described above. See pic.2.

In case of correctly performed import procedure and completion of reverse engineering of imported code you will see updated structure of original component (pic.3).

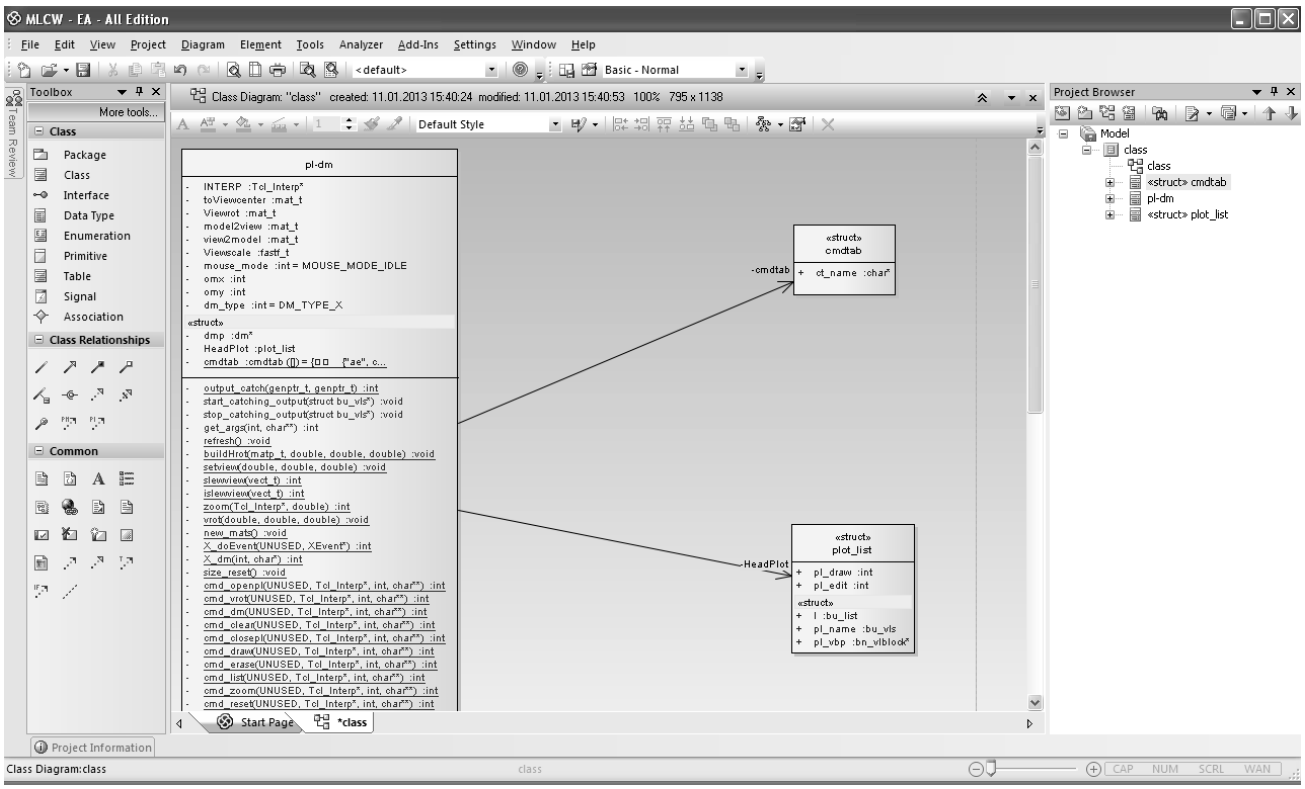

Fig. 3 – Updated structure of imported code

5. Code generation for new structure

Ensure all mandatory parameters setup in EA before generating new code it is necessary to select corresponding structural elementsand then follow the steps: see pic.4.

a) Source Code Engineering;

b) Generate Current Element…

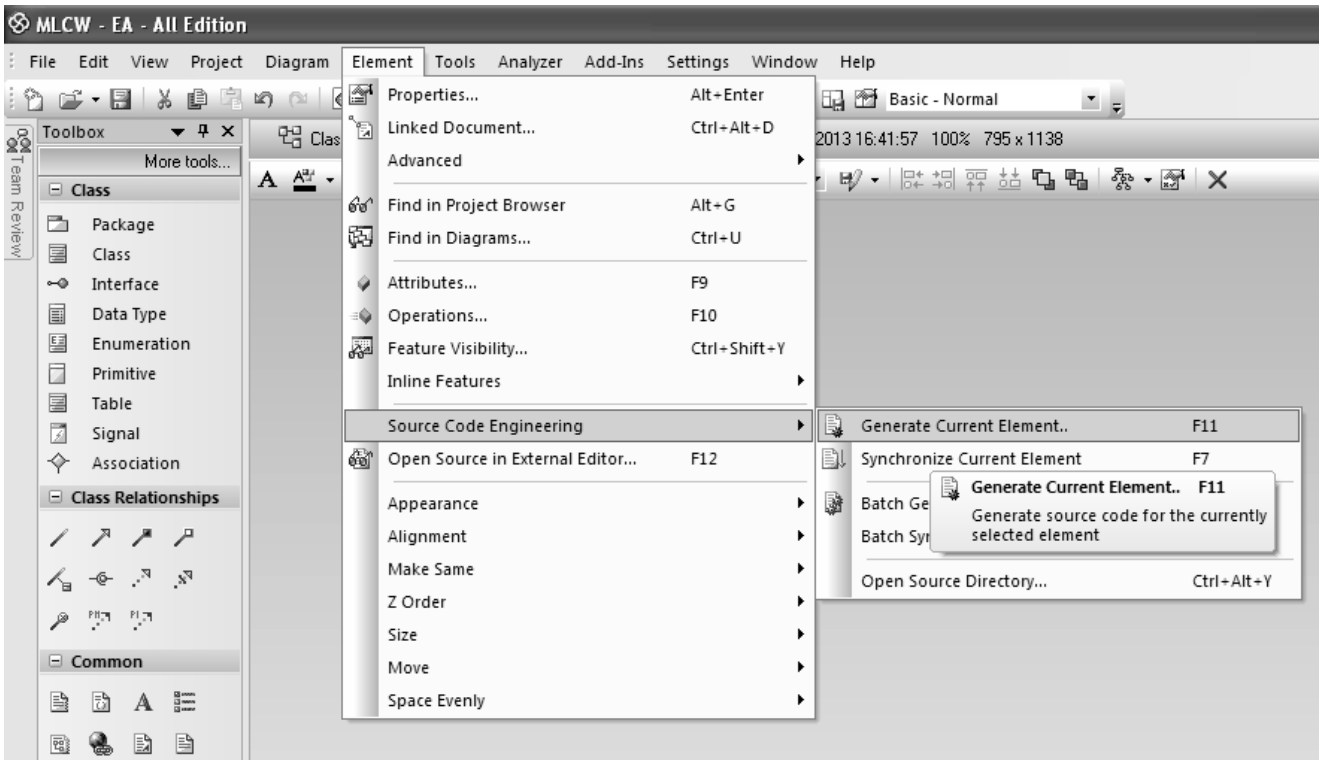

Fig. 4 – Prompt generating new structure

Ensure to specify correct pass in prompt window in order to save new recoding component. For e. g. “Desktop” see pic.5.

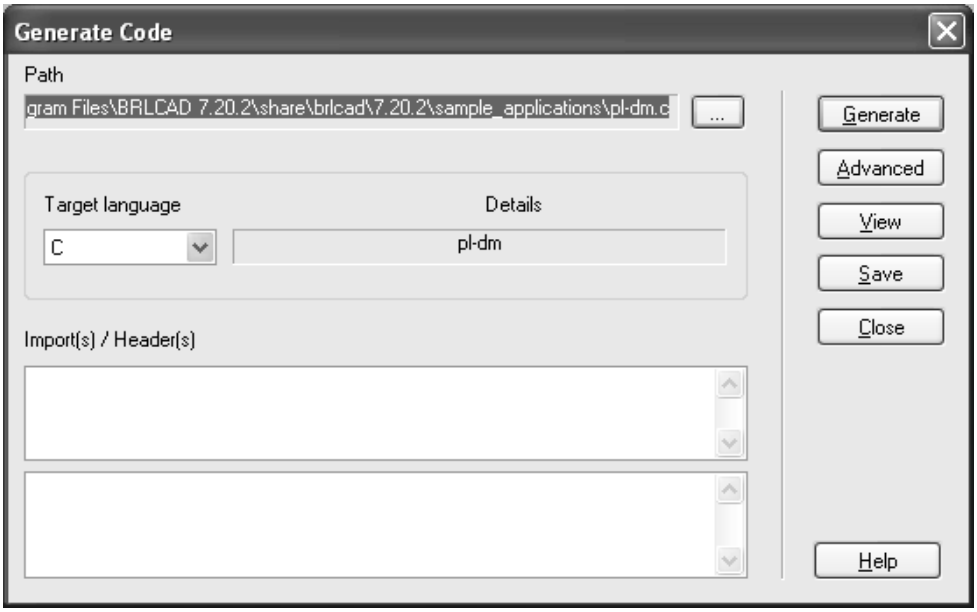

Fig. 5 – Example of code generation prompt

After selecting language for conversion for e.g. PHP press “Generate” button (pic.6).

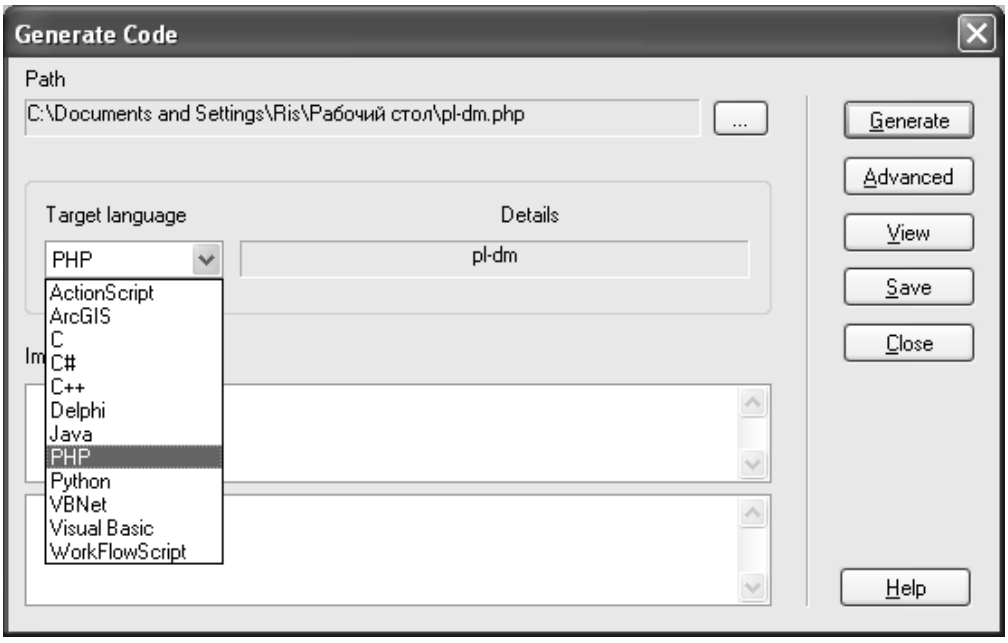

Fig. 6 – Choosing language of recoding

In case of successful operation you will find updated SCADA component written in selected language in specified place (in our case it’s “Desktop”).

6. Editing recoded component

The abovementioned component of SCADA is available for further editing and / or with help of EA or any other editor suitable for selected language, could be integrated into another SCADA.

Conclusion and prospects for the method further development

The article is outlining the new method of multilingual recoding for SCADA software. This method consists of definite steps combined into sequence illustrated above.

Correctly performed recoding should deliver following results:

– reduced number of errors while redesigning SCADA;

– improved cost effectiveness for new system development by reusing components;
– shortening man-hours by cutting down on recoding of repeating operations by reusing already defined components.

In author's opinion implementation of this method will increase efficiency SCADA in various fields.

This method could be used by system architects and programmers already working on existing SCADA reengineering projects.

Future reengineering model development will lay the ground for other support types of redesign SCADA. Reengineering SCADA will overcome the discrepancy between the pace of development of science and technology and design processes, improve technical support SCADA and reduce operating costs.

References

[1] E. M. Lavrischeva i V. N. Grischenko, Sborochnoe programmirovanie. Osnovyi industrii programmnyh produktov. K.: Nauk. dumka, 2009;
[2] S. S. Velykodniy, "Problema reinzhiniringa vidov obespecheniya sistem avtomatizirovannogo proektirovaniya," *Upravlyayuschie sistemy i mashiny*, no. 1, pp.57–61, 76, 2014;
[3] S. S. Velykodniy, "Reinzhiniring sistem monitoringu ta distantsiynogo upravlinnya sudnovimi energetichnimi ustanovkami," v *Avtomatika 2015: Mater. XXII mizhn. konf. z avtom. upravl.*, Odesa, Ukraine, sent. 10 – 11, 2015, Izd. Odesa: ONAFT, 2015, pp. 133–134;
[4] S. S. Velykodniy, "Metodologicheskie osnovy reinzhiniringa sistem avtomatizirovannogo proektirovaniya," *Upravlyayuschie sistemy i mashiny*, no. 2, pp.39–43, 2014;
[5] I. P. Norenkov, *Osnovy avtomatizirovannogo proektirovaniya*. M.: MGTU im. N. E. Baumana, 2009;
[6] A. A. Timchenko, *Osnovi sistemnogo proektuvannya ta sistemnogo analizu skladnih ob'ektiv. Kn. 1. Osnovi SAPR ta sistemnogo* proektuvannya. K.: LibId, 2000;
[7] A. A. Panteleymonov. "Aspekty reinzhenerii prilozheniy s graficheskim interfeysom polzovatelya," *Problemy programmirovaniya*, no. 1-2, pp.53–62, 2001;
[8] M. Fauler, *Refaktoring: uluchshenie sootvetstvuyuschego koda*. SPb.: Simvol-Plyus, 2003.

UDC 681.5.033

THE INTEGRATED LAYOUT DECISIONS FOR AUTOMATIC CONTROL OF PACKING SYSTEMS

A. Volodin[1], L. Kryvoplias-Volodina[2]

[1]National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kiev, Ukraine
[2]National University of Food Technologies, Kiev, Ukraine
ORCID: [2]0000-0001-9906-6381
Researcher ID : [2]L-7826-2016
E-mail: [1]sashavolodin4574741@ukr.net, [2]krivoplyas-volodina@ukr.net

Abstract: On many industrial objects regulators of pressure are used. Following initial data are necessary for proper selection of a regulator and calculation of throughput: a working environment, entrance and target pressure, expense, temperature of a working environment, accuracy of regulation, a class of tightness, diameter of the pipeline, type of accession, constructional materials, external or internal registration of target pressure, protection against excess of pressure.