UDC 004.423

O.V. Boychenko, DPhil, associate professor,
O.S. Lenkov,
L.V. Okhramovich

METHODOLOGIES OF DESIGN OF ARCHITECTURE SPECIFICATION

Paper explores the possibility of applying the methodology of designing architecture specification in solving the problems of the development of new technologies for the sustainable design of layered software information system for special purposes (ISSP).

By using cause-effect relationships between the tasks of the system based on their mutual dependence on the data, the conditions for obtaining the scheme of tasks free from conflicts generated by the lack of synchronization.

Applying of the methodologies, oriented at the processing at the expense of the method of functional decomposition, as well as of the design method of data structures etc. allows the determination of the conditions of a detailed format of each input element ISSP.

Keywords: *methodology, architecture specification, the method of converting graphs layered design technology, stable software.*

Solving the problem of increasing fault tolerance, reliability, and survivability of special-purpose information systems (ÌSSP), which often operate in conditions of significant amounts of data, uncertainty and limited terms of processing and providing information for making management decisions require the sustainable development of software and systems for the control of unauthorized influence on information resources of the system. To develop methodological fundamentals of processing based on a modular principle, method of functional decomposition methods design using data flow, methods of design based on the use of data structures for the design of sustainable software ÌSSP of his time were specialists such as E. Jodan., M. Halstead, H. Myers, M. Jackson, A. Zbìtniev and others [1–5].

The need to conduct research to address the challenge of raising the stability of software ÌSSP due to the features of the methodology of architecture specification, which is the basis of a new technology of designing of sustainable software automated computer systems. Methodology, oriented to processing, provide particularly important process and structure in the creation of the architecture program. In particular, the most known methodologies focused on processing, based on 5 main methods (Fig. 1).

The main concept of modular programming [1]: each module implements a single independent feature; each module has a single point of entry/exit; module size as possible to minimize; each module can be designed and coded by various members of the crew of programmers and can be tested separately; the whole system is built from modules.

In this approach, proejected system is divided into several parts simultaneously created a variety of programmers. Each module implements a single function. Module size small, so testing is managed and can be carried out thoroughly. After the coding and testing of all modules is their integration and test the whole system. When accompanied by tested and adjusted only to the module that does not work. Obvious benefits in relieving writing and testing programs, reduced the value of their escorts. The module must not change the team another module and usually not obliged to store the history of your calls (although the last property greatly simplifies the trace when the crashes).

Known positive quality modular programs: the ease of ordering and debugging. Functional components of such a program are also establishing separately; It is possible to well structured Setup as "up" and "down"; ease of maintenance and modification; the possibility of sharing modules between the programmers of various skill level according to their level of complexity; the ability to create libraries of the most used programs; simplification of loading into memory a large task that requires the segmentation; the occurrence of many natural control points for monitoring the promotion of programs used to enhance ustalenostì software.

Transmission control module structure occur only on vertical lines connecting the modules in the diagram hierarchy. Any module can activate the slave module and gain control after the completion of his work. A lower layer may not cause a higher stratum. Control module is therefore obliged to return the module that it caused. Making lower-level modules for higher-level modules are not permitted. Excessive number of arguments that are available for the module indicates the separation of functions (the main purpose of this is reducing the number of arguments). In the paper [2] provides a formula for the number of modules in the program:

$$M = {}^{\Pi}\!/_{6} ,$$

where m is the number of modules. The formula is based on the assumption that for a perfect module number of parameters is equal to 6. Functional decomposition is essentially based on a strategy of "divide and drive". Renowned specialist in the field of methodology of programming of Parnassus, which tried to formalize the procedure of functional decomposition in the form of incremental detail, as a criterion for decomposition the system proposed the concept of information hiding. Using this criterion, each module is characterised by subjective decision of the designer.

Only some information about this module requires other modules, the relationships between modules are organized using a well defined interfaces. Another important idea is to design a software system as a set of virtual machines instead of the traditional approach, which uses a flowchart. The advantage of functional decomposition in her life. The drawbacks from the unpredictability and variability. Methods of designing using data stream using the stream of data as the driving force of the design of the program. This is used to display a variety of functions that transform the flow of information in the structure of the program. Structural design based on the concept that put forth in works of Jodan (1979) and Mayers (1978). It is sometimes called composite design or transformation design. The method tries to fight with hiboû, the usual method of functional decomposition, using which you can not control the quality of the decomposition of the function. This approach consists of the concept of structural design, general line composite design and detail project criterion measures, methods of analysis of the project.

The approach consists in displaying data flow problems in the structure of the program using some methods of analysis of the project. Adopted the following procedure: identified data stream and reproduced a graph of data flow; identified, Central and weekend elements; formed by a hierarchical structure of the application that uses these items; optimize the structure of the program formulated at the third step. This approach is usually applied in the absence of clearly defined data structures.

Technology of structural analysis project of the SADT is based on structural analysis, proposed by Ross. SA-graphical language used for intuitive expression of hierarchical and functional relationships between any objects and actions. The structure of the system presented graphically, interfaces between components of the architecture, modular and hierarchically. SADT includes procedures for planning, control, develop and manage configuration, tools, organization of working professionals in the cabin and the connections between them. SADT is successfully applied in various fields. The method is particularly effective at early and later stages of development of the system and less effective in detail. At the same time, allowing each designer to create independent chart, you can get additional complexity in the process of their viewing.

Conversion method of graphs based on the obtaining of system specifications in the form of graph control problems with the count of information communication between them. On the basis of the method is the use of causation between task system based on their mutual dependence according to the data. The main objective is to get the scheme objectives free from the conflicts generated by the lack of synchronization of data [3]. Regarding design, based on the use of data structures, there exist two approaches, developed independently by Jackson and Warner. Both are used as design architecture and implementation detail of the project.

In the methodology of Jackson data structure used as a key element in building good software project. The basic structure of the program is determined by the structure of the data that it manages. The program is seen as the mechanism by which input data are converted into the weekend. Using the incoming and outgoing structure as a basis, trying to get a well structured program. The main advantage of the Jackson methodology is that the quality of the resultant project does not depend on experience designer, each step of the design can be verìfied, a variety of designers, working independently on one and the same problem, get the same result.

However, the methodology is no recommendations, how to structure the data. The basic steps are to such [4]: to identify and show the structure of the input data and the structure of the source data; show the structure of the program, combining images of these structural elements; define discrete operations that make up the program; convert the text of the program.

The methodology of such a methodology for Jackson is that the key to the project is the data structure. However, the procedure of designing more detailed. Four types of views: charts of data, the chart logic completion, a list of instructions, pseudocode. Organization chart data describes the incoming and outgoing data. The chart logic completion serves a logical flow of this data. List of instructions contains commands that are used in the project. Pseudocode is required when describing the design deliverables. Methodology of can be summarised as follows [5]: identify all incoming data system; organize incoming data in hierarchical form; determine the detailed format of each element of the input file and fix the number of their appearances; repeat steps 1–3 for source data; the subsequent details of programs, identifying the types of commands that are contained in the project, in the following order: reading, branching, calculate, entrances, exits, call routines; use the chart type flowcharts to show the logical sequence of instructions using symbols to represent the beginning of the process, the end of the process, branching and attachments; number the items in a logical sequence and open them using the instructions recorded in step 5.

Hierarchy plus input, Processing, output is a method of hierarchical graphs, developed by IBM. Main characteristics: the ability to provide a link between the input/output data and process development; the possibility of decomposing system hierarchically, not attracting extra fine details; the use of three elements: input, processing, output. Process (process) specifies as the central block diagram, she connected with elements that reflect the input and output. The basic procedure of designing using HIPO: start with the highest level of abstraction; identify the input, output, and processing; Connect each element of the input and output of the processing;

documentation for each element of the system, using hipo charts; drill down chart, using steps 1–4. Programming of large software systems is significantly different from the programming of small software systems. Most programming languages do not give means of specification of interaction between modules. The first language that solves this problem, obviously, is considered the language of (relationship between the modules), formally specific modular structure and the links in it. In addition to descriptions of internally modular connections and attributes of the modules, is a project management and a means of support to the process of designing. Another option is similar to language is SDL. It describes the management interfaces of the module and the control system.

A defined place in the hold of language representation graph. The proposed method conversion graphs using the causation between the tasks system based on their mutual dependence according to the view, supported in motorcades. Functional-event-restricted decomposition vested in the foundation technologies of designing software for real-time systems (including automated). It is a generalization of the method of functional decomposition in the direction of real-time controls and enhance stability and survivability of software systems. In general, technology based on FPR-decomposition, integrates a lot of elements of the above approaches, developing the part of them independently and independently. Application of the methodologies for processing, due to the method of functional decomposition, method for designing data structures etc.., allows you to create the conditions for determining the detailed format for each element of the input data for IDMS in the development of new technology stepped design sustainability software.

LIST OF USED SOURCES

1. *Йордан Э.* Структурное проектирование и конструирование программ / Э. Йордан. – М. : Мир, 1979. – 415 с.

2. *Холстед М.Х.* Начало науки о программах / *М.Х. Холстед.* – М. : Финансы и статистика, 1981. – 128 с.

3. *Сбитнев А.И.* Структурная организация и проектирование ПОАСУТП / А.И. Сбитнев. – К. : УСМ, 1982. – № 5. – С. 38–42.

4. *Jakson M. A.* Principles of Program Design / *M. A. Jakson.* – N.Y. : Academic Press. – 1975. – P. 288.

5. A software design system based on a unified design metodology // J. of Information Proc. / O. Shigo et al. – 1980. – Vol 3. – № 3. – PP. 186–196.