TOOLS FOR VISUALIZATION AND CONSTRUCTION OF REAL-TIME SPACECRAFT'S CONTROL

Andrey Tyugashev, Anton Nasekin, Elena Sopchenko
S.P. Korolyov Samara State Aerospace University (National Reseach University)

The paper presents a research in the field of visual programming of spacecraft control programs. Means of visual simulation of the design and verification of macro can improve the efficiency of professionals in the design, development and maintenance of software onboard the spacecraft. The use of these funds will reduce the number of errors in the macro.

Nowadays it is impossible to imagine spacecraft without computers. Computers are being used from design stage throw lifting to Space, operation and support till from the end of spacecraft's service. Now, all satellites including micro and nano ones equipped with onboard control computer system which combines several onboard computers integrated into network.

The control functions is being executed by special sort of software – onboard software (so-called «mission critical flight control software»).

Onboard software consists of a large number of programs. The most important parts are programs that perform the functions of the operating system, service software, «functional programs» that control the operation of concrete onboard system, and «integrated macro control» software.

Modern satellites controlled by both ground-based and onboard control complexes. Practically, in case of an onboard complex whole logic embedded in the software.

One of the main objectives in creating the spacecraft is to provide high-quality, correct and robust software that can support satellites operation not only in normal mode, but in the case of a failure (as far as possible or correct it), or in the worst case, contact ground control personnel and provide them with an opportunity to fix critical bugs by their means.

Thus, the error in onboard program can lead to serious financial losses, and it is not the worst-case scenario.

One of the causes of errors is the complexity both of onboard software as well as of its development process. At the initial stage of developing, program logic's designers create and transmit all the information about the control logic to programmers. Misunderstanding or inaccuracy in these processes can lead to errors.

System we proposing is intended to use visual technology to minimize errors in the programs for spacecraft's onboard computer. In the ideal case, it will allow to develop «integrated macro control» part of onboard software by system designers, excluding programmers. As a minimum, we present the program in a more understandable form to avoid misunderstanding in collective.

We use graphical notation DRAGON (Friendly Russian Algorithmic Language that provides Visibility) visual representation was proposed to use.

«Macro autonomous integral control software» is the 'coordinator' of 'conductor' of all programs and processes implemented by onboard equipment [1]. Macro-program consists of groups of 'logical sequences'. Each logical sequence has a condition (analog of 'guard condition'). If this condition (it can be very complex with many clauses connected by 'OR','AND', 'NOT' logical operators) is truth, some commands should be executed in specified time moments.

We are trying to develop a set of programming tools that will allow to support a 'macro-programs' design and verification.

These tools must be integrated with the existing onboard software development tools and interact with an existing database [2].

One of the most important task is to visualize a group of logical sequences in the Dragon graphical notation. It is also necessary to provide the ability to edit the scheme and correctly save of all changes to the database.

We use DRAGON notation and specially developed program tool to solve specified problem. Graphic language consists of the following elements:

- graphic primitives and lines (arrows);
- text labels located inside or outside of graphic primitives.

'Graphical alphabet' and 'graphical syntax' is being used. Graphic elements in this notation are known as the 'icons'. The arrows in the diagrams are used very rarely. In addition, there is a ban on the use of line crossings. In addition, there are icons in the notation which standardize combination of icons, so-called 'macro-icons', which usage is similar to how words combine letters. These features made the language more concise and thus more intuitive and understandable in comparison with 'traditional' flowcharts. All graphic elements are shown in Fig. 1.

	Icon	Name of Icon
1		Title
2		End
3		Action
4		Question
5		Choice
6		Case
7		Headline
8		Address
9		Insertion
10		Shelf
11		Formal parameters
12		Begin of FOR loop
13		End of FOR loop

	Icon	Name of Icon
14		Output
15		Input
16		Pause
17		Period
18		Start timer
19		Synchronizer
20		Realtime parallel process
21		Comment
22	XXX XXX XXX	Right comment
23	XXX XXX XXX	Left comment
24		Loop arrow
25		Silhouette arrow
26		Connector
27		Concurrent process

Fig. 1 – Graphic alphabet of DRAGON language

At a moment, we have developed a prototype of 'visualization' and 'graphical construction' tool that allows:

- connect to existing databases with description of satellite's 'integrated macro control' software.
- visualize the structure of 'macro-program' in DRAGON notation.
- edit the scheme and save the result in the database.

The example of developed tool's screenshot with DRAGON representation of the 'macro-program' structure is presented in Figure 2.

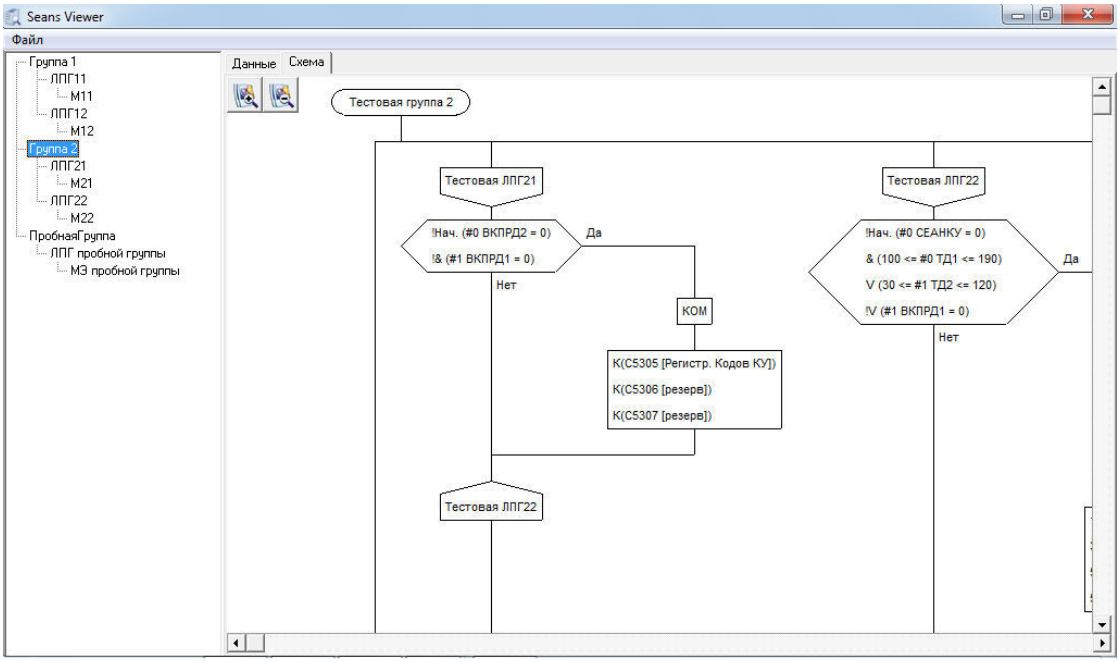

Fig. 2 – Visualized macro-program

As a result of this work, we will have a software product that allows visualizing and constructing a group of logical sequences of 'macro-programs'. The tool provide separate editor for the each element in the group, with customization of the properties, and thus affect the operation of Macro.

Due to the fact that a human better perceives graphical information, the method may accelerate the development of new macros, to reduce number of errors in the design stage and the time needed to 'discover' (find) the errors.

It should be emphasized that the proposing method only requires user to know DRAGON notation and overall principles of satellite's «macro integral control software». The tool has an intuitive interface and can be used by not a professional programmer but by 'system logic designer'. So, due to a clearly defined subject area we have the opportunity to avoid the use of any of the common programming languages and apply the model of visual programming.

References

1. Kochura, E.V. Razrabotka makroprogramm integral'nogo upravleniya kosmicheskimi apparatami [Development of integral control macro-program]. Vestnik SibGAU [Herald of Siberian SAU], 2011, vol. 1, pp. 105-107. (in Russian).

2. Esyunin, V.V. Osnovnye printsipy postroeniya ispytatel'nogo stenda bortovogo kompleksa upravleniya dlya perspektivnykh izdelii OAO «ISS» [Basic principles of construction the test stand for onboard control complex]. Vestnik SibGAU [Herald of Siberian SAU], 2010, vol. 2, pp. 93-96. (in Russian).