

UDC 681.3

SYSTEM ENVIRONMENT «BRAINSTORM» FOR SUPPORT OF THE INHERITED SOFTWARE

V.P. Zimin, E.A. Khomyakov

Tomsk Polytechnic University

E-mail: zimin@ido.tpu.ru

The problem of support of the inherited software, arising at accompaniment of existing applied programs packages is analyzed. Solution of the given problem with the help of system environment «BrainStorm v. 1.0» under MS Windows which allows to create and to accompany software packages which were carried out under MS-DOS is proposed. The basic mechanisms of system maintenance realized in «BrainStorm» are described. Results of testing of the given tool means are given.

For the last some tens years, owing to intensive development of information technologies and mathematical simulation, were created a plenty of the universal and specialized problem-oriented packages of applied programs (PAP). Creation of such packages is an actual problem as they actually are the form of support and transfer of knowledge and technologies. But objective and subjective difficulties of development, creation and support of such program complexes leads to a number of problems, one of which is the problem of the inherited software (SW). The essence of the given problem is, that by virtue of number of the reasons the developed programs cease to be used, therefore new generations of researchers has to spend again significant resources to create software packages with similar functions. These reasons can be classified as subjective and objective. The subjective reasons are caused, first of all, by an insufficient level of support of the software, in connection with disintegration of developer collectives, partial or full loss of initial texts of programs. The objective reasons of occurrence inherited ON are caused by the methodology of PAP creation. Let's consider objective component of the inherited SW problem in more detail.

The methodology of development of the problem-oriented program complexes recommends to divide the software of packages on functional and system parts [1, 2]. The Functional part of the PAP contains program realization of computing algorithms for a researched subject domain, and the system part contains program support of various specifications of interaction and services: the interface and technology of the user dialogue with the package, interprogram interfaces, the common data processing algorithms of (including visualization of the data) etc. One of the important characteristics of the SW is its life cycle (LC) which represents set of stages (time intervals) of development, operation and support of the created software. At early stages of development both of software and hardware of the PAP as a whole, and functional and system maintenance of a PAP the LC time interval of the last actually coincided. This circumstance, in particular, was explained by weak development of PC software: input of data from punched cards and punched tapes, text-oriented dialogue without additional checks of the entered data, output as a numbers, at the best – output of diagrams onto the paper carrier. With appearance of personal computers (PC) the situation in development of system maintenance has essentially changed: the full-text and graphic interface of the user, dialogue in

the form of menu, ability of the distributed, remote data processing and other technologies of creation, operation and support of SW has appeared. Change of technologies of creation of the system SW occurred and occurs promptly, for the short time interval. For example, the expanded interpretation of the Moore law, its extension on system maintenance of computers is proposed in [3]: change of operational system each 3 years, change of interfaces each 6 years. In the same time functional filling of packages which basically is defined by computing algorithms, varies much more slowly. Though new technologies of data processing have appeared and develop here again. It is pertinent to note such new direction of development of functional maintenance of PAP as algorithms of parallel high-efficiency calculations on computing systems. In our country multiprocessor computing systems are less distributed by virtue of their dearness though in last years this direction of SW development is stimulated by occurrence of rather cheap PC clusters.

By virtue of different velocity of change (updating) of system and functional filling of PAP the objective problem of support of inherited SW is shown. On the one hand, the system SW ceases to be supported in a due measure at a level of operational system, and on the other hand, functional package SW to be modernized constantly for synchronization with the life cycle of system filling of a PAP. And solution of the given problem depends essentially on that system information technology which is necessary to support, to obtain new functional qualities of program complexes. For example, solution of the inherited SW problem was considered in [4] from the point of view of support of the remote access to computing resources. Solution of the given problem from the point of view of support of the functional maintenance developed for operation in operational system MS-DOS, under OS MS Windows, and putting to users of the appropriate services is considered in the present article. Solution of the problem is based on the idea to use at creation of system maintenance of PAP a minimal set of mechanisms (technologies) of integration of functional filling of the packages which eventually do not change, or vary a little with time.

With the purpose to define minimal requirements to tool means of support of inherited SW, let's analyze some tendencies in methodology of creation of PAP, as well as state of system and functional maintenance the packages developed before. At solution of complex scientific and engineering-technical problems value of

mathematical simulation essentially increases. However analytical abilities of the last are limited, actually, by the class of linear problems. Solution of actual problems of a science and engineering demands application of a PC and technology of computing experiment (CE) and, as consequence, development and operation of a PAP-SW means of CE [1]. Similar problems arise and at creation and designing of complex samples of engineering and technology when it is required to carry out mathematical carrying out on all stages of their designing and test. It is noted [5], that in the PAP to be developed three levels of subject domain description have to be presented:

- 1) Stage of development cycle of the technical project and technical propositions. At this stage operative calculations basically are carried out by the simplified models.
- 2) Stage of draft designing. The approved models of an average level of complexity, but taking into account distribution of the object (process) of designing are used.
- 3) Stage of realization SR, obtaining of new fundamental knowledge of work of designs. Creation of new complex mathematical models, their approbation.

At solution of both scientific, and engineering – design problems it is required to use technology CE, providing, first of all, requirements of multimodality and multi-variant at realization of computing works. In the subsequent works [6, 7] the list of stages of technology of PAP creation actually repeats the CE stages, but at the same time the more detailed interpretation of each stages, generalization of experience of concrete PAP creation occurs.

The analysis of available packages and means of their building has allowed to reveal the following laws in their development [1, 2, 5-8].

- 1) The organization of input /output and intermediate data of functional units:
 - Various type files;
 - Databanks;
 - Various data basis DB (which are files with special structure);
 - Storehouses of the data;
 - Other, for example, storage of the not structured data in various formats.
- 2) Forming of logic units of functional providing of packages:
 - units in the form of executed or interpretive programs with arbitrary;
 - units in the form of executed or interpretive programs with the specialized interface;
 - procedures modules -.
- 3) Association of logic units of a PAP functional maintenance:
 - with help of the specialized language.
- 4) The organization of the user work with help of various a package elements:

- input/output data for functional filling;
- logic units of functional filling at their start;
- histories of carrying out of researches of a various level (scientific, design, estimated);
- interface for organization of the user dialogue with the package (a text – oriented graphic interface; formation of tasks, packages etc.).

Thus, from the point of view of the user at creation of tool means for creation and support of an inherited SW it is necessary to keep technology of computing experiment, paying the basic attention to the organization of interaction of functional modules and input/output data.

From the most general positions the functional module can be presented as the program processing by the given algorithm an input data and forming by certain rules the output data. Taking into account these circumstances, the carried out analysis of system maintenance of the existing PAP has allowed to formulate functional requirements which be have to be satisfied by the tool software to be created:

- 1) Operative classification (splitting into classes) sets of functional modules of the PAP.
- 2) Work with different data sets, with different databanks.
- 3) Fast start of functional modules PPP, use of the input/output files predetermined by the user which describe a configuration of input/output data of functional modules.
- 4) Creation of executed (linear) chains of functional modules.
- 5) Carrying out of edited journal of results of start of functional modules and chains of modules.
- 6) Realization of the modern interface of the system environment (Window-interface).
- 7) Visualization of input/output data of functional modules on a plane.

The common component practically in all listed above requirements to the system environment is the task of the effective organization of data exchange. The analysis of the general mechanisms of operational systems such as MS-DOS, Win16 and Win32 has allowed to choose the most simple and reliable model of data exchange: use of the control system of files and a command line. Both mechanisms of interaction of program and the data are available practically in all operational systems, and their use does not demand the additional analysis and structuring of streams of the data. Let's note, that both Win16, and Win32 have much more abilities on the organizations of interaction of programs, than OS MS-DOS.

Let's consider in more detail realization the above specified requirements within the framework of system environment BrainStorm which base window is presented on Fig. 1.

For the purposes of operative classification of functional modules by an user the required hierarchical structure of directories (*categories*), with the corresponding names are organized. With help of the drag&drop

technology from the directory «All modules» (where all set of functional modules is placed physically) filling of the category directories is carried out, using the mechanism of references to modules.

Multi-variant of calculate experiment is displayed as plurality of input/output data of functional modules. The task to save the alternative numerical experiments is solved by two ways. The first way consists in duplication of file names of the input/output data, but in variation of their expansions. The most simple way of coding is – the number of experiment variant coincides with the three-valued number obtained from figures of expansion of the file name. The second way is organization of databanks with help of a system environment (Fig. 1, «Databanks»). Databanks, sets of the input/output data of functional modules, are organized by the user. At work with the concrete databank it is required to define it as «Current». At a plenty of alternative calculations the second way is preferable.

To integrate the functional module written in language Turbo Pascal 7.0, the following technology is used in the system environment. The input and output data of the module are united in input files (inputs.txt) and output ones (outputs.txt) (Fig. 1, the directory «Input/output») which are obtained the module at start from a system environment. In addition, the mode of execution of the module is indicated: *batch* or *interactive* (a key is *batch*). In a batch mode all dialogue in the module is suppressed. For example, start of the module pcc03m_p.exe in the environment of system environment BrainStorm in a batch mode will have the form:

```
pcc03m_p.exe --bs-i=inputs.txt
--bs-o=outputs.txt --bs-batch.
```

Structures of text files of input and output of the module are identical and consist of the lines containing the description of the form «key=value». For example,

```
/ptcsd=..\banks\base\Ptcsd.001
/krekcd=..\banks\base\krekcd.011
/vndvcd=..\banks\base\VNDVCD.001
/dugjvcd=..\banks\base\Dugjvcd.050
```

The given file of input contains in the lines the names of files which are chosen from the basic databank (..\banks\base\) and which contain, finally, the input/output data of the functional module. As such data simple (text) variables, can be specified, for example:

```
Progress=1
Iteration=150
```

Such variables are interpreted in the module to be executed. To define variables as input data in the module body the function `GetParam (param:string):string` is used, for output data – procedure `PutParam (param, value:string)` is used. To suppress dialogue in the batch mode of execution of the module the global logic variable `BatchMode` is used. Let's note, that the output file is created by the module (procedure `SaveOutputFile;`), that is the indicator of its successful execution. In the system environment both functional modules (for example, pcc03m_p.exe.inputs), and the contents of files of input (pcc03m_p.exe.inputs) and output (proba.exe.output) are accompanied by the description. For example, structure of the input file pcc03m_p.exe.inputs of the module pcc03m_p.exe is the following:

```
/ptcsd=Dependence of cesium pressure
on temperature of the tank with cesium
/krekcd=Output works of the emitter
and the collector for electrode pair
```

...

The window of start of the module pcc03m_p.exe is presented on Fig. 2. The batch mode of execution of

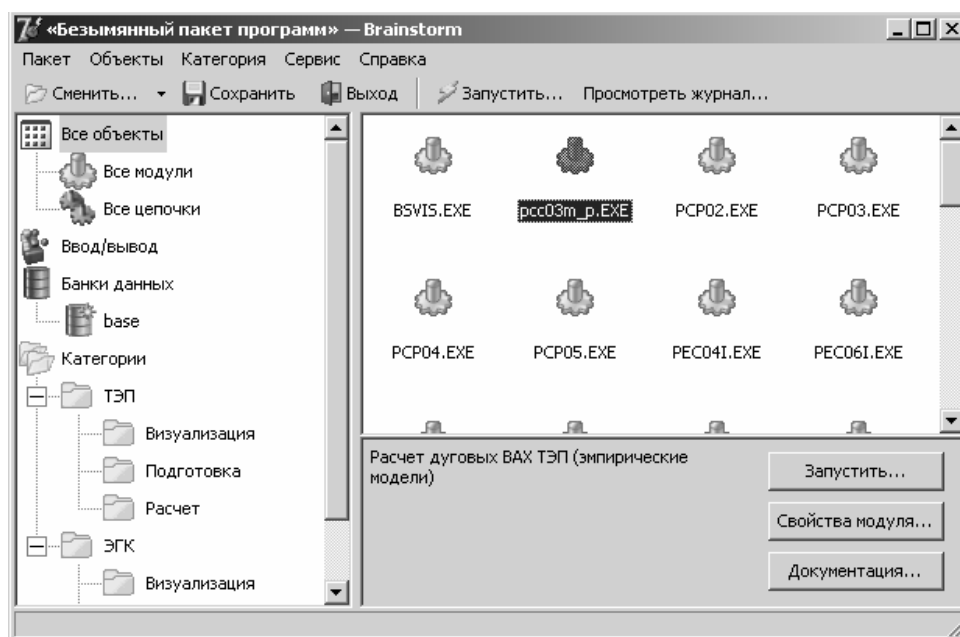


Fig. 1. A base window of the system environment BrainStorm with the loaded PAP

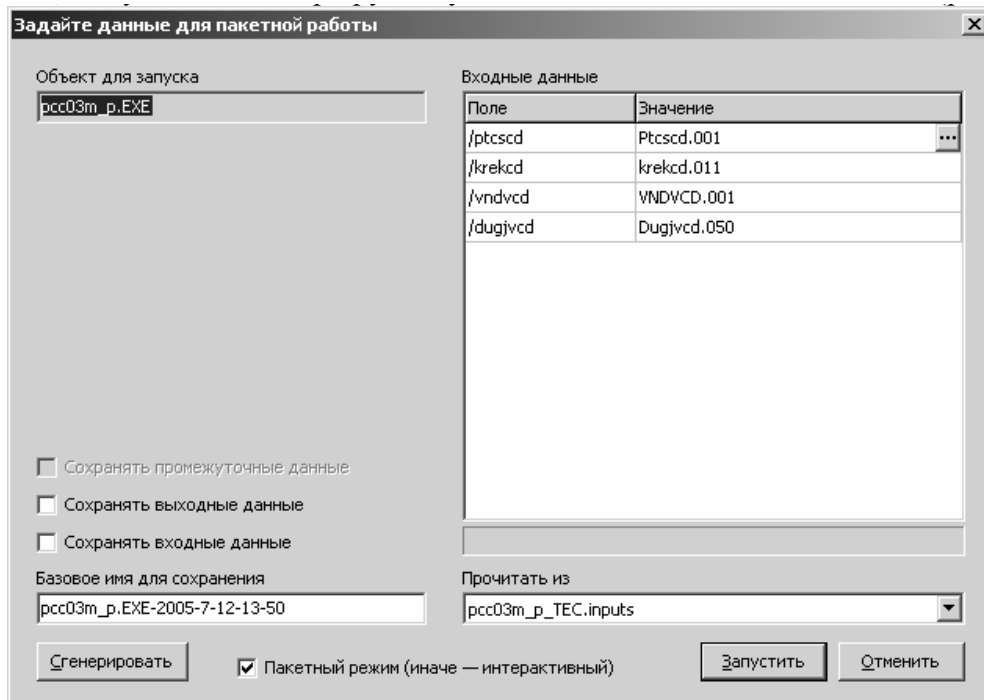


Fig. 2. A window of start of the functional module

the module is realized and the concrete names of input file are specified. The route is not presented, since the route of the current databank – `..\banks\base\` (Fig. 1) is established.

The system environment BrainStorm allows to form from set of functional modules in an interactive mode structure of linear chains of modules. The user specifies a desirable sequence of execution modules in the batch mode and monitors on the basis of description files of the of the module and its files of input/output (Fig. 3). At this «*relay race*» of the data is formed which shows which data (identifiers of data) will be had by the user before execution of the concrete module of a chain and what – after execution.

Each start of the module or chain of modules is recorded in the journal of the system environment. *The journal* is an edited text file, in which the system environment specifies the name of executed object (the module or a chain of modules), the content of files of input/output and result of execution. The user can add his comments in journal.

Modules of visualization form *files of visualization* with *XML-structure* which are interpreted by the program of visualization `bsvis.exe`.

Thus, use of service of a control system by files and the mechanism of command line allows, on the one hand, to modify minimally the code of the programs written under MS-DOS and included in the PAP, and on the other hand – to obtain the interface under OS MS Windows to the user of the created package supported by system environment BrainStorm.

The first version of system environment BrainStorm is written about use of system of visual programming

Delphi 7. The nucleus of the environment is formed by the independent module Tpackage which controls the PAP. Control of a package is carried out with help of object methods: control of functional modules, control of categories (for classification of modules), control of chains of modules, control of files of input/output, control of databanks, control of logging and etc. After preparation of functional modules for integration in the PAP, the special file structure which as its filling is displayed in the interface of a system environment is created with help of BrainStorm.

Testing of system environment BrainStorm v. 1.0 was carried out by two ways.

First, functions on supports of the PAP with help of modeling system (a system environment) KOPTES [8] which was written in language Object Pascal were realized. Except for the interface under OS MS Windows, functions of work with functional modules, with input/output data of modules, functions of visualization of the data are essentially expanded. The functions have appeared, which allow to create and carry out chains of functional modules. Testing of system environment BrainStorm has completely confirmed that execution of functions is adequately to functions of modelling system KOPTES.

Second, software package KIAE_2004 for modeling processes in thermoemission systems [9] was created. 31 functional modules written in language Turbo Pascal v. 7.0 are included in the package. The databank of package KIAE_2004 contains more than 100 input/output files of modules. The structure and the contents of the data at integration in a package did not change, except for functional modules of visualization.

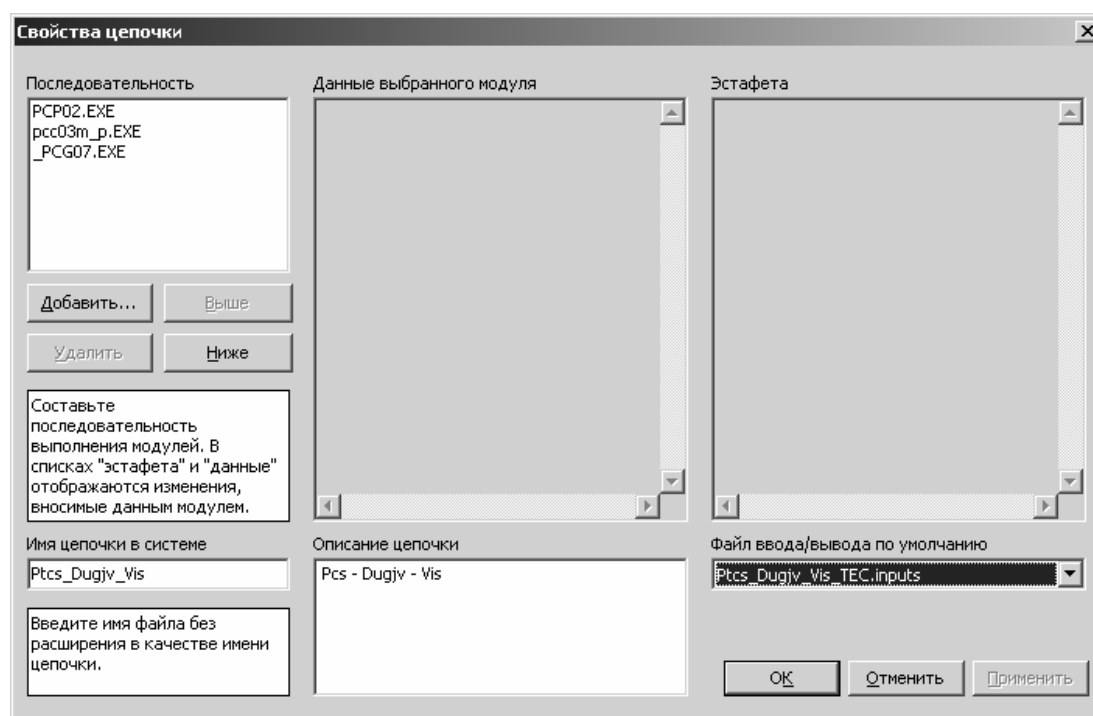


Fig. 3. A window for formation of a chain of modules and the control of its properties

At testing of the package KIAE_2004 the results of modeling being identical to the data [8], and also the new results of modeling stated in [9] were obtained.

In the conclusion it is possible to draw the following conclusions:

- To solve the problem of support of the inherited software it is necessary to use the basic mechanisms of effective data exchange, being characteristic for operational systems such as MS-DOS, as well as Win16 and Win32: file control systems and command line. Both mechanisms of interaction of programs and data are available practically in all operational systems, and their use does not demand the

additional analysis and structurization of data streams.

- Mechanisms of effective data exchange in the PAP are realized in system environments BrainStorm v. 1.0 under MS Windows and checked up at creation of two packages of applied programs.
- The given system environment can be applied at a stage of initial formation (prototyping) of the PAP, to write which functional modules one uses the modern programming systems when the fact of data transmission between PAP modules itself but not efficiency of module communications is more important.

Literature

1. Samarsky A.A. The applied program package as means of maintenance of complex physical calculations. [in Russian] // Prospects of system and theoretical programming: Proc. Allrus. Symp.; I.V. Potosin (ed.). – Novosibirsk, 1979. – P. 5–13.
2. Janenko N.N., Karnachuk V.I., Kononov A.N. Problem of mathematical technology. [in Russian] // Numerical methods of the continuous medium mechanics. – Novosibirsk, 1977. – V. 8. – Issue. 3. – P. 129–157.
3. Ilyin N.P. Expanded interpretation of the Moore law. [in Russian] // Information technologies. – 2005. – No 5. – P. 70–71.
4. Massel L.V., Gornov A.J., Podkamenny D.V. Creation of computing resources in Internet on the basis of the inherited software. [in Russian] // Joint release of magazines «Computing technologies». – V. 7 and «Vestnik KazNU». – No 4(32). – Proc. Intern. Conf. «Computing technologies and mathematical modelling in science, engineering and education». – P. 3. – Novosibirsk-Almaty. – 2002. – P. 247–252.
5. On manufacturing techniques of engineering – physical APP. A package «Arcan». [in Russian] / S.S. Bondarchuk, A.B. Vorozhtsov, A.S. Zhukov et al. // Languages and parallel computers. – M.: Nauka, 1990. – P. 79–90.
6. Potapchuck G.A. The tool means to support process of creation of program systems of discrete optimization. [in Russian] // Cybernetics and system analysis. – 1992. – No 4. – P. 125–133.
7. Parasyuck I.N. Principle of development of modern program systems of data analysis. [in Russian] / Cybernetics and system analysis. – 1993. – No 3. – P. 155–162.
8. Babushkin Ju.V., Zimin V.P., Sinyavsky V.V. The model system KOPTES to research thermal and electric processes in thermoemission systems of energy transformation. [in Russian] // Space-rocket engineering: Works. Ser. XII. – Issue. 1–2. Calculation, designing, engineering and tests of space systems / V.V. Sinayavsky (ed.). – Kaliningrad, Mosc. region: RCC «Energy», OSTI, 1998. – P. 60–78.
9. Babushkin Ju.V., Zimin V.P., Khomyakov E.A. The software and results of thermoemission system modeling. [in Russian] // Izvestiya Tomskogo Politekhicheskogo universiteta. – 2006. – V. 309. – No 3. – P. 53–57.