A FORTRAN PROGRAMMING SYSTEM FOR COMPUTATIONAL PROBLEMS

D. B. Agafonov and A. P. Smirnov

The article describes an approach to the design of a programming support system for the construction of modular computational programs. The system automates generation of programs for simulation of multi-parameter mathematical models.

1. INTRODUCTION

The KOMFORT programming system (Russian acronym for KOnstruktur Modul'nyi programm na FORTrane — Fortran Modular Program Generator) is intended for automatic generation of computational Fortran programs. Programs are generated from a set of specially designed ready-made Fortran modules. A distinctive feature of the system is that it does not use an explicit computational model of the application domain for which programs are generated.

The computational model is needed to construct the problem-solving algorithm and to synthesize the program from modules. Application domain models are often specified by a functional-semantic net, which carries information about the objects participating in the solution of the problem and functional relationships among them (PRIZ [1], MIKROPRIZ [2]). A different approach to application domain modeling is used in the DIMO interactive system [3], where the knowledge base containing information about the application domain is created by a dialogue scenario editor.

In the KOMFORT programming system the application domain is defined directly by the available set of program modules, more precisely, by functional descriptions of these modules, their internal structure, and interaction options stored in natural-language module headers (in the form of Russian or English text). Module headers are manipulated using a hypertext technology, which is becoming highly popular in information systems. The use of natural languages produces an informal description of the application domain. The abandonment of a formal computational model ensures subject-independence of the programming system and also permits reducing the source language to a purely interactive language.

The article describes the underlying principles, structure, and operation of the KOMFORT programming system.

2. HYPERTEXT TECHNOLOGY

A key concept of hypertext technology [4-6] is the hypertext, which is a nonlinear text organization with text fragments placed at the nodes of a network structure (a hypertext network). Material presented as hypertext can be used in any order allowed by links between the network nodes: use is not restricted to a linear sequence, as in classical text representation. The existence of a link between hypertext objects signifies their semantic closeness.

Hypertext nodes can be accessed in different ways. The main access technique uses node-visualization active zones. An active zone is usually implemented as a text fragment highlighted in a window corresponding to the node; it opens access to nodes with explicit links to the given node. A particular node also can be reached by specifying it explicitly, for instance, in a browsing window. Many hypertext systems also provide an option for scanning the contents of the hypertext network according to some preselected index of its nodes. The sought node also can be located with the aid of implicit links between the nodes. This technique requires specifying a set of key words, some of which may be selected in the body of the text corresponding to one of the active nodes and others may be entered from the keyboard. The system then selects hypertext nodes that are linked with the active node. To simplify the search for the target, the user can group hypertext nodes by some attribute.
3.1. General Description of KOMFORT

The KOMFORT programming system [7] is intended for generation of modular Fortran programs for problems in computational physics. KOMFORT is designed for applied mathematicians with minimum experience in Fortran programming. KOMFORT includes several interconnected components, which constitute a single integrated environment. We list and briefly describe these components.

1) **Program Module Archive** (the application part of KOMFORT). The Archive contains Fortran code of program modules stored in KOMFORT format.

2) **Fortran Program Generator** (the core component of the system part of KOMFORT). The Generator generates Fortran programs from the modules stored in the KOMFORT Archive. It consists of two subsystems: Integrator and Link Editor. The Integrator integrates archive modules into programs in an intermediate format (see the section on Integrator below). The Link Editor organizes the interaction of the modules assembled by the Integrator (see the section on Link Editor). Both the Integrator and the Link Editor use elements of the hypertext technology. The Generator, like other KOMFORT components, is accessed interactively using menus and prompts. To protect the Fortran programs from accidental erasure and at the same time to save disk space, the Generator creates a protocol log that contains all the information required for complete and unambiguous reconstruction of the Fortran programs.

3) **Fortran Program Structure Analyzer**. The Analyzer determines the functional structure of the program by partial parsing. The Analyzer is used in the automatic archive updating subsystem (see below) to extract modules in KOMFORT format from Fortran programs.

4) **Executor** (subsystem for execution of ready Fortran programs). Without leaving the KOMFORT environment, the Executor assembles object modules, compiles, debugs, and runs programs using a custom Fortran compiler. For programs generated by these executions without directly accessing the program code. This option is also available for programs created outside the KOMFORT system. In this case, the Formatter is first used to insert special configuration control elements into the program code. The subsystem includes a message translator that translates compiler messages into Russian.

5) **Formatter** (subsystem for updating the Fortran Module Archive). The Formatter creates new KOMFORT modules. It functions in two modes: a) extraction of modules from ready Fortran programs using the KOMFORT Analyzer (see above) and their conversion to KOMFORT format; b) creation of new modules “from scratch.” The Formatter also can be used to insert configuration control elements into any Fortran program.

6) **Special-Purpose Text Editor**. Used practically by all KOMFORT components.

7) **Custom Fortran Compiler**. Used by the Executor.

3.2. Operation of the KOMFORT System

Given a set of some well-documented program modules (an archive), their descriptions may be regarded as semantically linked fragments of some hypertext that contains information about a certain application domain.

KOMFORT modules may be of four types: P modules contain main (PROGRAM) fragments in Fortran; S modules contain the source code of Fortran subprograms; F modules contain the code of Fortran functions; I modules contain Fortran code for insertion.

The module documentation is stored in the module header, which is created interactively when the module is prepared for inclusion in the Archive. The documentation includes 1) a detailed general description of the module functions, 2) descriptions of each subprogram and function call occurring in the module code, 3) information about allowed insertions into the module and available switches that extend the scope of applicability of the module, 4) a brief characterization of each module variable.

All this information is stored a) in the form of natural-language text fragments (for the user), b) in the form of strings of key words (for KOMFORT).

The Formatter is responsible for creation of documentation, proper formatting of calls to subprograms and functions, insertions, switches, and some other operations that convert a Fortran module into a KOMFORT format when a new module is added to the Archive.