Computer Science Journal of Moldova, vol.**5**, no.3(15), 1997

The study of data and control flows and the user's interface organization in an applied system for chemistry and medicine

N. Shvets

Introduction

As a rule, the applied systems design is at the border of a few scientific areas. Due to the fact, it is combined with a wide range of different problems to be solved in the course of the system development.

As far as the applied systems are not self-containing systems mostly, we have to determine firstly the structure of the system in the whole (or; in other words, the system's macrostructure) and the place of the approach under implementation in the interaction with other external programs used. Secondly, the relationships among the parts of the system are to be studied, too; this point concerns primarily the flows of data and control investigation. After establishing the relationships, one more important problem is to be viewed, namely, how the user interface in the system will be organized.

The three items are strongly interrelated, and changes in one point can cause immediately the changes in the whole system, if the latter is not designed properly. So, the parts of the system are to be independent as much as possible, all the interactions are to be classified and described at a conceptual level and the structure of the dialog with users determined before being implemented.

In the article we are dealing with a previously described applied system for the biological activities investigation [1, 2]. So, we shall remind shortly its purpose and structure before going to the main topic of the paper.

1 Overview of the system for 'structure-biological activities' relationships (SAR) investigation

The applied system implements the so-called electron-topological (ET) approach [3] to SAR investigation. The approach is based on the idea that some fragments, or substructures, common for all active molecules and possessing definite physical-chemical and quantum-chemical properties determine the activity investigated and can be used to predict the activity for new compounds synthesized (possibly, according to the rules previously formulated by the same system).

The system uses its own language of the molecular structure description, ET matrices of conjunction, or ETMC, including data characteristic of the lowest possible, atomic, level of desription (diagonal elements are numerical data characterizing atoms and non-diagonal ones reflect a property of bonds). To obtain the data, a few commonly used programs for conformational analysis (or geometry optimization) and quantum-chemical properties of the molecules calculations are combined with the ET-method [4, 5].

As initial data, we have to take a series of compounds with structures known and activities determained in an experimental way. Among them there must be active and inactive compouns either.

The main steps of the process of SAR investigation are the following:

1. **Geometry optimization.** Results of the program applied to every compound are transferred to the program for quantum-chemical calculations in a text format acceptable by the latter.

2. **Quantum-chemical calculations.** Data obtained (one file for each compound) are numerical values used to form ETMCs. They are transferred for farther use to the local data base, being transformed in an appropriate way.

3. **ETMC formation.** As far as the data base contains a few numerical characteristics for both atoms and bonds in every molecule,

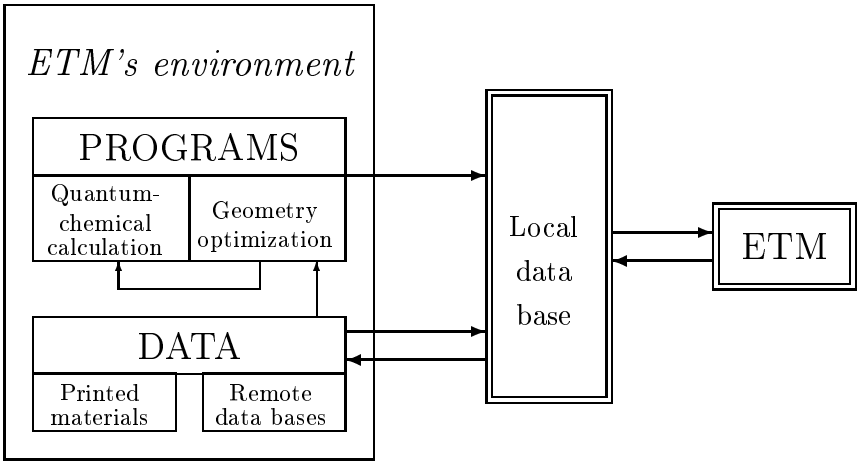

Figure 1: Interaction of the ETM with its environment (data flows)

the user of the system has to choose characteristics of interest for him to be included into all ETMCs.

4. **Activity features selection.** By means of comparison with the most active molecule (i.e. with its ETMC) the main procedure of ET-method (ETM) seeks for the fragments of molecular structure, being submatrices of ETMCs of active compounds only. The procedure is supervised by two sets of parameters, initial ones, reflecting the property of molecular flexibility, and estimating parameters, reflecting the 'goodness' of each fragment.

5. **Building the system of the activity prognostication.** The properties of atoms and chemical bonds entering the submatrices found serve as a base for forming a set of rules used for prediction of the activity investigated.

The macrostructure of the applied system (ETM-system) corresponding to the routine presented above is shown in Figure 1.

Following the macrostructure in the operational mode appropriate for users, a more detailed structure of the system can be elucidated, including the such important aspects of applied systems as user interface and corresponding to it flows of data and control in the system.

2 Interaction with users: its exterior and implementation

2.1 General characterictics of the user interface in the system

Man-machine interface (MMI) called also user interface provides the connection between the system's user and a process carrying out one of the system's tasks. From the point of view of software there are *two main components* in MMI: a set of input/output (I/O) processes and a process of dialog. I/O processes can serve two purposes:

- to accept from the user and to pass him some data needed
- to organize data flows between any subtasks of the system and data processing.

We may point out the following *types of the processes on the data transfer* in the system:

P1:	FILE	$\longrightarrow$	FILE	(FL-FL)
P2:	FILE	$\longrightarrow$	DATA BASE	(FL-DB)
P3:	FILE	$\longrightarrow$	SCREEN	(FL-SCR)
P4:	FILE	$\longrightarrow$	PROGRAM	(FL-PR)
P5:	SCREEN	$\longrightarrow$	PROGRAM	(SCR-PR)
P6:	SCREEN	$\longrightarrow$	FILE	(SCR-FL)
P7:	SCREEN	$\longrightarrow$	DATA BASE	(SCR-DB)
P8:	DATA BASE	$\longrightarrow$	SCREEN	(DB-SCR)
P9:	DATA BASE	$\longrightarrow$	FILE	(DB-FL) ,

and each of them is appropriate for some definite phase of computing. Some of the types serve for organizing interaction between the envi-

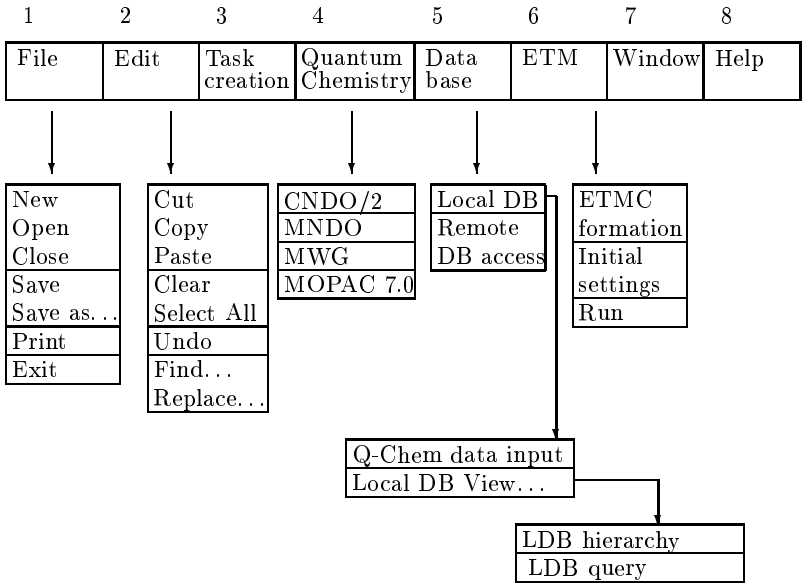

Figure 2: User interface in the ETM-system: the structure of the main menu

ronment of the ETM and the core of the ETM-system, others for the dialog with the user in the frameworks of the ETM-system itself.

As far as the sequence of actions in the system is predefined, the interface may be classified as a *machine-guided interface* (MGI). For such kind of MGI the best representation known is a system of menu.

A main menu representing the user interface in the ETM-system is shown in Figure 2 with the corresponding *submenus*. A short description of the items follows.

1) File.

The corresponding submenu reflecting all actions applicable to files is a standard one:

2) Edit.

Besides the operations on files shown, a simple editor is needed. Its main operations are known, too.

As we see, the two submenues are related with the types FL-SCR and SCR-FL of data transfer.

3) Task creation.

As far as the item is a starting point in the scheme of the ETM implementation, there is to be a dialog with the user aimed in the information on the task to be solved receiving. It concerns the name of activity under investigation, the name of the task, the classes of compounds in the series taken etc. This information is inputted into the LDB to initialize the new task. The dialog realizes the data transfer of the type SCR-DB through the use of standard procedures for DB opening, writing and closing, belonging to specialized Object Windows Libraries (as in the **item 5** of the main menu). It should be noted that the program for geometry optimization is viewed as a separate program; so, we consider all the results of the conformational analysis as obtained in advance.

4) Quantum-chemical calculations.

To prepare input data for the ET-method, the interface is to define first of all the kind of program used at the step (there may be a few of them, as it may be seen from the submenu). Then it chooses an appropriate program transforming output files of the previous step (geometry calculations) to the format accepted by the concrete quantum-chemical program. Programs named '*converters*' serve the aim. Here they realize the FL-FL type of data transfer. The data structures needed to implement the kind of programs are discussed below. A simple dialog is needed here to set directories containing input and output files. It implements the FL-PR data transfer type. (As far as some of the programs use a standard output file, a kind of package processing is to be organized by the interface to rename it in correspondence with the name of the input file on the completion of the program.)

5) Data Base.

Suppose that for all compounds from the series under investigation quantum-chemical calculations are carried out, resulting in a corresponding number of files. From the logical point of view filling in the ETM-data-base is to be the next step. The latter is considered as a *local data base* (LDB). However, to make the ETM-system more powerful, one more possibility provided in the system, namely, the possibility of working with data obtained from *remote data bases* (RDB) is reflected in its submenu.

The first item of the submenu has a continuation, too. First, we have to use another converter which translates the output data format of the previously run program to the data representation characteristic of the LDB (the data transfer's types FL-DB). To view the LDB (the DB-SCR data transfer type), two possibilities are given: a picture demonstrating the LDB tables' names and their relations and a dialog for forming different queries with a definite number of key words given as DB-filters.

Relative to the RDBs, there may be two ways of using them. First one concerns the situation when not only other RDBs can be accessed by the ETM, but the LBD itself may be the object for using by other remote users. In the case the item is for setting relations with remote users on the base of 'client-server' principles. But, to do this, the LDB must be powerful enough. So, for the present time it is enough to obtain data from RDB, only. It may be done in much more simple way: using Internet and WWW, we can receive data from RDB as files or standard tables and then translate them into the LDB representation by means of converters. These actions are in the scope of the submenu for the LDB.

To prevent data from non-control lable use, a dialog through which the data can be inputted into the LDB in a controllable way is realized in the system, being of the FL-DB data transfer type.

6) ETM.

When input data for the programs implementing ETM itself are present in the LDB, two preliminary actions are to be done prior to the process of the activity features selection:

- forming ETMC for every compound and
- setting pre- and post-conditions controlling the features selection (see the corresponding submenu).

Both kinds of action are realized through appropriate dialog windows by means of which a data is requested from the user and passed to the main program (or to one of several equivalent programs) used for the features selection, together with the file of ETMCs for the series given. The actions are the data transfer processes of the types SCR-PR and DB-FL.

For the ETMC formation a number of selections is to be made:

- a property of atoms
- a property of bonds
- a filter (to reduce the size of matrices, if needed).

Initial settings are the following:

- a directory containing the file of matrices
- pre-conditions Δ_1, Δ_2 (the limits for the diagonal and non-diagonal elements variations, respectively)
- post-conditions P_a and α_a (the probabilities of encountering the features selected in the classes of active and inactive compounds)
- a name of a compound taken as a target for comparison with the rest of compounds.

The process 'Run' curries out the cycle of the feature selection for the same file of matrices but with different initial settings till the user is satisfied by the results obtained.

The last two items of the main menu, No7 and No8, are traditional enough: their aims are to switch between active and inactive windows and give an explanation, when needed.

Considering both the choice of an item of the menu described and a program called from the item after a dialog as processes, together with

erroneous situations handling, we can draw then a diagram demonstrating all connections among the processes (see Figure 2; the letter m with indices indicates an item of the menu).

Although there are many types of data transfer (DT), all they have much in common due to their sources and destinations (namely, screen, file and data base). Therefore it is worth to view any kind of I/O as an independent one. The nature and place of any such process in the scheme of the data exchange allows us to consider the processes as objects being passed from one subtask of the system to another one. The use of C++ as the implementation language assists in treating them as objects to a maximal degree.

2.2 Converters

Being alike relative to their tasks and computational nature, the data transfer processes are of a special sort at the same time because they are dealing with the tasks of I/O in the system, delivering data to programs. Two main (abstract) actions that are to be fulfilled by any DT procedure are to take a value from a source and to pass it to a destination. So, we have to find out previously those data types that are used for I/O in concrete programs and, secondly, it is worth to treat separately input and output in every case.

Writing and visualizing databases is a task that can be solved through the standard classes for DBs' use (the classes TTable, TComponent, TMethod etc.). The screen handling is realized mostly through the components dealing with windows and through the standard I/O streams (although we can customize the latter for our needs, too). So, the I/O processes relative to files form the base for the class of converters' implementation, including the most necessary for us operations of I/O.

Determining basic data types used for I/O is especially important for reading/writing formatted texts. Because of calculating character of the task implemented, numerical and alphabetical data are the object for I/O (so far no graphics). The following actions can be taken as basic ones for reading/writing files:

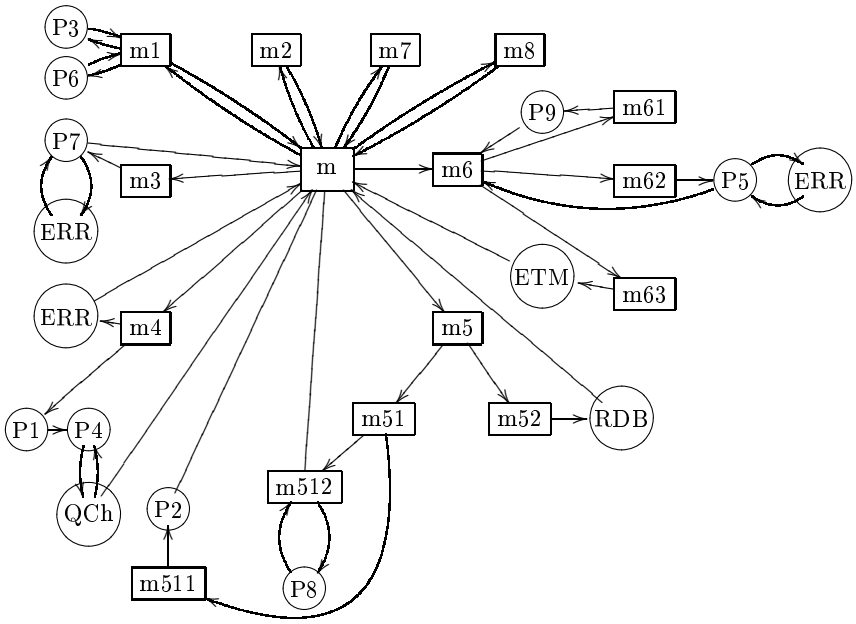

Figure 3: The scheme of data and control flows in the ETM-system

- delete spaces (in the input file)
- write n times a symbol (into the output file)
- write numbers using different formats
- read a string
- write standard strings (constants)
- read/write a file name
- position an input/output file etc.

In addition, some more complex operations on strings, including type conversion, are needed.

In this way, as far as there is an hierarchy of classes in C++ used for I/O, dialog and data base handling, the classes implementing converters and dialog are to be included into the hierarchy as classes inheriting their properties. As far as there are too many of such classes, we will show the most important dependencies for the classes implementing DT processes of the ETM-system in the scheme (see Figure 4).

In addition to the basic processes described above, the processes of a special type related to error handling were mentioned. As far as their nature corresponds to the nature of failing processes, their implementation is to be taken into account when dealing with each group of similar processes through the standard actions contained in corresponding classes.

As it can be seen from the said above, in spite of the complexity of the objects representing all kinds of interaction in the system, their implementation can be realized step by step beginning from simple components to more complex objects easily enough. The schemes developed and the advantages of C++ as the implementation language assist in solving the task.

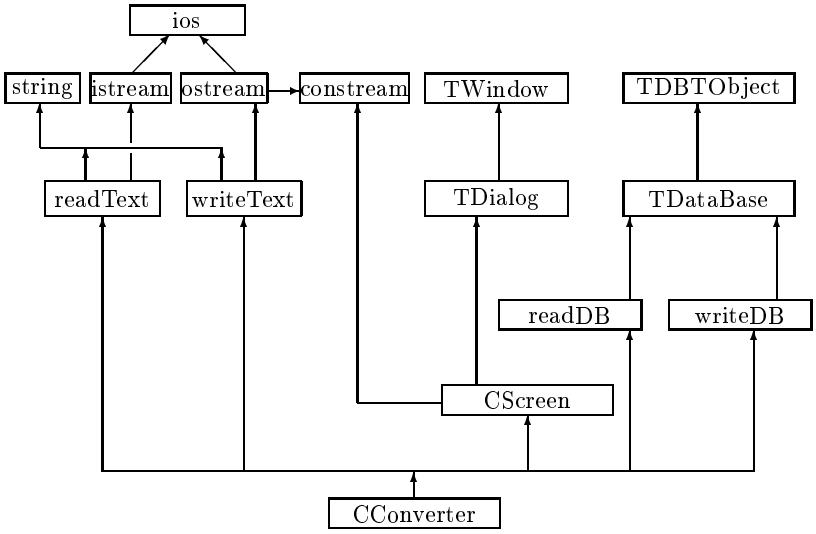

Figure 4: Relations among some of the classes implementing data transfer prosesses

References

[1] N.M.Shvets. Program implementation of the SAR investigations for the series of compounds oriented on the interactive experiment planning. / Proceedings of the 9-th All-Union Conference "Chemical informatics" – Chernogolovka, Moscow, 11–15 of Jan., 1992, p.226.

[2] N.M.Shvets. Applied program system for the prognosys of biological activity of chemical compounds: development and use. // Computer Journal of Moldova, No.3, Kishinev, 1993, pp.101–110.

[3] A.S.Dimoglo. The compositional method of description of the electronic structure of organic molecules oriented to computer analysis of SAR // Chim.-Pharm. Zhurnal, vol.19, 1985, pp.438–442 (Russian).

[4] K.E.Gilbert, J.J.Gajevski. / A MMPMi Molecular Mechanics Program. Indiana, 1985, p.123.

[5] J.J.P. Stewart. / MOPAC, Version 6.0, QCPE Program No.455. 1995.

N.Shvets,
Institute of Mathematics,
Academy of Science of Moldova,
2028, Kishinev, Moldova.
phone: (373–2) 738073
e–mail: 22*nata@math.moldova.su*

Received October 24, 1997