ADAPTIVE DIALOGUE - THE BASIS FOR PERSONAL COMPUTER SYSTEM

Victor Briabrin

Computing Center, Academy of Sciences, Moscow, USSR

1. Personal Computer Systems (PCS) represent nowadays a significant trend in the professional and amateur use of computers, in particular of mini- and microcomputers. We believe that such systems must have the means for:

(a) knowledge representation about the problem domain, the system itself and the particular users;
(b) man-machine interaction, taking benefits of both partners' abilities;
(c) access to application-oriented programs and data as well as to other computer systems.

Our experience with natural language question-answering systems has proved that the share of QAS users preferring real NL is rather small. Among the reasons one can point out such features as inadequate heaviness and slowness of NL-processors, their unportability. These reasons, on one hand, and the apparent success of simple table-driven and production rule systems, on the other, have led us to the development of Adaptive Dialogue System as a basis for PCS.

Our approach is based on the integration of 3 core components allowing an efficient system implementation within the limited microcomputer resources:

the Object Base (OB) constitutes a media for storing and manipulation of systems', problem-oriented and lexical objects;
- the Adaptive Dialogue Monitor (ADM) provides the guidance

for man-computer communication through dialogue processes;

- the P-Machine (PM) accomplishes the necessary operations supporting system s own behaviour as well as the access to applied programs and data.

Any process in the system takes place within a certain environment which is characterized by a set of mutually accessible objects and by initial dialogue process. A dialogue process or D-process comprises a number of interconnected states and supports particular man-computer activity on achieving some goal. Almost every state of D-process accomplishes a simple act of communicating with the user, analyzing his reply, performing associated actions and transiting to another state. Thus an analogy could be drawn between D--process and ATN, with the important difference that D-process supports ongoing dialogue with the user rather than analysis of preentered string.

There are 3 state types:

MENU - prompting the user to choose among alternatives;
ASKVAL - accepting the asked value from the user;
XVAL - getting computed value from external program.

By means of MENU and ASKVAL D-process is communicating with the user´s terminal, XVAL connects it with applied programs. The actions within the states are expressed as strings of P-codes performing predication of values, manipulation of different objects, calling/returning to another D-processes, running external programs etc.

Adaptivity is of primary concern in the presented approach since it gives the user, on one hand, the ability to subject the system to his particular needs; on the other - to put himself into an active or passive position during some communication process. Adaptivity is provided by strictly following one principle! any kind of system´s activity pertinent to man-computer communication must be controlled by definite objects which can be created and modified by the same or other activities.

2. There are 3 basic kinds of objects in the OB: system supporting (S-objects), problem-oriented (PO-objects) and lexical (L-objects). Each of these kinds is further divided into categories or types.

S-objects hold the control over man-computer interaction and the access to applied programs and data. They describe anything residing or happening in the system and therefore are called "descriptors". The following descriptors are of the most importance:
DE - environment descriptors
DP - dialogue process descriptors
DS - dialogue state descriptors
DX - program descriptors
DD - data descriptors
DG - dialogue scenario ("game") descriptors
DU - user descriptors

PO-objects describe concepts, properties, relations and values allowing to build problem-oriented models. PO-objects are differentiated by their categories (types):
C - classes of physical objects: "country", "person"
R - semantic relations: "negotiation", "disarmament"
V - structured value descriptors: "date", "NATO"
T - terminal value descriptors: "ATOM", "STRING", "INTEGER"
P - object properties substantiated by V and T - values: "population", "birthday", "address", "weight"
CP - object properties substantiated by the other C-objects: "capital" (of country), "relatives" (of person)
RP - semantic relations arguments: "participants" (of negotiation), "place", "goal" (of any activity)
CI, RI, VI, TI - individual objects and actual values - substantiations of C, R, V and T-objects.
For external representation of objects we use a formalism of F-language developed for DILOS system /1/. Simple examples:
capital (CP): IS city; OF country

politician (C): IS person; from country; post official-post
official-post (V): IS list; OF ATOM; EL (president defense-minister...)
NATO (V): IS list; OF country; EL (Belgium Gr-Britain...)
Weinberger (CI): IS politician; from USA; post defense-minister
defense-minister (TI): IS ATOM; ELOF official-post

The problem domain model could be expressed in a form of conceptual semantic network /2/ with the nodes represented by C, R, V and T-objects and the arcs labeled by P, CP and RP--objects. Representation of actual world fragments would then have a form of terminal semantic network with CI, RI, VI and TI-objects being the nodes, connected by P, CP and RP-arcs.

L-objects or "lexems" provide access to all other objects in the OB using the words and phrases of problem-oriented natural language (POL). Each lexeme has external representation (= the name of L-object) derived from the input word form by simple morphological transformation. Each L-object has special properties which serve for two purposes: (1) pointing to other objects - denotations of the given lexeme and (2) carrying grammatical marks for efficient POL analysis.

Presently L-objects are divided into the following 10 categories:

LN - lexical names of PO and S-objects
LS - synonyms pointing to other L-objects
LC - components of word combinations and idioms
LQ - question tokens - pronouns and adverbs ("who", "how")
LJ - conjunctions of different types ("and", "or", "but")
LR - referencial pronouns ("these", "his")
LP - prepositions with indication of implied semantic cases for different relations
LD - "definitors" establishing paradigmatic relations between objects ("is", "denote", "contatin")
LA - articles
X - ignored or unknown lexemes

3. The system starts its operation by running initial D-process which, first, makes acquaintance with particular user and, second, prompts him with the main menu to choose among several basic activities:

(A) filling the OB with the new objects or modifying the old ones;

(B) entering particular problem-solving task;

(C) getting information on system components and processes.

In (A) one might concentrate on creating the new dialogue schemata which is done by building new DP and DS-objects. This special kind of activity results in a new communication language (sublanguage) which might be immediately tested and used for practical needs. Another kind of (A) is connected with the creation of PO and L-objects. The user can do it two different ways:

- bý entering pattern-controlled expressions which define particular object categories;
- by calling special D-processes responsible for building appropriate objects.

Although from the user´s standpoint these alternatives differ in the share of man-machine activities, both of them are based on the same <u>rules</u> having the general form:

```
<pattern>  [:  <condition>] .⇒ actions
```

<pattern> defines the combination of object categories accepted from the user, no matter if they come sequentially (one object per D-state) or simultaneously - as one line of text. Optional <condition> allows checking particular properties of the accepted objects. The < actions> amend the old objects or create the new ones. A simple rule might look as following:

```
(X ´OF  C1  LD  C2) =    Put(X, ´TP, ´CP)
    Put  (X, ´OF, C1) Put(X, ´IS,  C2) Put(C1,  X,  C2)
```

Such a rule can handle the expression: "president of country is a politician" which might be entered in one sentence or spread over several ASKVALs. The result would be creation of new CP-object "president" and modifying the C-object "country".

In (B) the change of initiative can be softly controlled by the user in the sense that he should not apply special knowledge or feel any barriers when moving from one mode of interaction to another. In any mode, however, a result of ADM operation is reflected in F-expression /2/ which carries the semantic meaning of the input sentence.

Thus, for example, a phrase "Defense ministers of NATO met in the capital of France on October 18th" could be transformed into the following F-expression:

```
(Meeting: participants (politician (s):
                    post defense-minister
                    from (country (s) : ELOF NATO))
          place   (capital : OF France)
          time    (date : M October D 18)
```

Such an expression is directed to PM where it is interpreted with the result of creating/pointing at some nodes in the terminal semantic network thus representing the meaning of the entered text.

References:

/1/ N. Asafjeva, A. Borkovsky, V. Briabrin, V. Ponomarev, G. Senin. Knowledge Representation and Natural Language Processing in DILOS. In "The Problems of Applied Systems Development". Ed. A. Narin'any. Novosibirsk, 1979.

/2/ V. Briabrin: F-language - a Formalism for Knowledge Representation in the Intelligent Dialogue System. In "Applied Informatics", V.1, ed. V. Savinkov, Moscow, 1981.