A LINGUISTIC APPROACH TO THE DESIGN OF A LANGUAGE FOR COMPUTATIONAL LINGUISTICS

V. Andrewshtshenko

Faculty of Numerical Mathematics and Cybernetics,
Moscow State University, USSR

Computational Linguistics is a sphere of science and its applications lying between linguistics and computer science. The main task of computational linguistics is developing methods and designing tools for man-machine communication. To this task its two main directions are subordinated: natural language data processing, including machine translation, and automation of linguistic research, including automatic lexicography. The formation of computational linguistics requires designing a unified language, rich enough to satisfy diverse computational conditions arising in the above-mentioned applications.

As such a language we propose LICOL (LIngua COmputationum Linguisticarum), an inevitably short review of which is given in the present report. The linguistic approach to the design of this language consists in viewing it as a semiotic system and forming its units in accordance with R. Jakobson's linguo-semiotic functions.

LICOL is intended for communication between man (native speaker, user) and computer (interpreter of this language) in communicative situations of automatic and automatized (user directed) natural language data processing. The concepts and constructions of LICOL presuppose a rather broad range of users - from nonprofessional ones (translators, lexicographers, editors, etc.) to computational linguists and traditional programmers. LICOL can be used as a command language in

information retrieval, as a data description and data manipulation language in data base design and as a programming language in the usual sense. It is intended as means for defining and control both in dialogical (on-line) and in batch (off--line) processing.

As a semiotic system LICOL consists of signs - bilateral entities - composed by signifiants (sequences of letters) and signifiés - those entities of real or conceptual world which are objects and/or means for automatic processing. The role of signifiés in this language can be played by the signifiants and by the signs of the same language. This "world of language" we name the system of its concepts the relations of which are expressible in terms of relations between the signs of the language. To use LICOL, specifically to program in this language, is to express one's thought, knowledge, notions in the framework of the "world of LICOL" in accordance with the rules of its grammar (syntax). In this language the function-argument form is chosen as means for stating relations between the concepts: suffixal compound stems and prefixal and/or suffixal incorporated syntagms, the part of the suffixes being played by names of operations. The notation of LICOL therefore has the reversed Polish form.

Since LICOL is a language for computation, the main concept of its "world" (i.e. the main sign of its semiotic system) is the notion of Computational Construction (CC). The CCs are constructional material both for the programs and the data, the underlying form for which are IC trees. Since the trees are easily representable in a linear form by means of the Polish form of notation, it is natural to interpret programs as data and vice versa, the data being fractured into relationally-hierarchical data network. This allows to consider the data base also as a data base for procedurally represented knowledge and therefore not to draw distinctions between the two main forms of data representation - a naming and a procedural one.

According to the type of signifiants the CCs are divided into representing and processing constructions, each of which are further subdivided: the representing class into names and pictures and the processing class into controllers and operations. If the signifiant is, semiotically, a symbol, we have to do with a naming construction, if it is an icon, we have a picture construction; the index-sign represents either a controller or an operational construction. The signifiés for the CCs are the so-called descriptions consisting of a descriptor (which corresponds to the concept of the sign) and of its referent (value).

According to the type of signifiés the CCs are divided into real, virtual and notational constructions. The real CCs correspond to external data of the usual programming languages. They are structured into elements, chains, fields, records, fragments, sets and bases. The virtual CCs correspond to the user's notions of processing and are structured into atoms, sequences, trees, bunches (arbitrary graphs), blocks, files and (file) systems. The notional CCs correspond to constitutive parts of entities of signifiants. These are: letters, strings, groups (of strings), segments, modules, corpus and packets (of texts).

According to the form of value the CCs can be subdivided into scalars, vectors and lists. The scalar CCs have the following types: numbers, codes, logicals, figures, symbols, keys, references, descriptors and masks. The notion of the vector corresponds to one of array, its components may be not only scalars but also vectors or lists, provided their components are of the same type. The lists may consist of scalars, vectors or sublists which may be of route, tree, structure or executive type.

Such multibase classification of the CCs has the following sense. The operations of LICOL are defined on the virtual CCs having various origins: either notational (textual)

constituents or denotational, virtual and real CCs. They may be intended either for displaying in textual form, or they may be used in further processing as virtual ones, or they may be transmitted in external environment in the form of real CCs.

The CCs can be defined either by description of their type and the mode of evaluating or by a picture, the simplest type of which is a literal. Two or more CCs can be associated together one of them being an object and the rest of its features. There are the following possibilities: implicit transformation of data from one type into another; indirect definitions of operands; participation in operations by objects and their features both separately and jointly; evaluation of the operands via pictures, the operands may be thereby procedures. Diverse operations on sequences, sets, graphs with labelled and unlabelled nodes and arcs are defined. This allows operating both on the constituents and dependences, to form both the paradigms and syntagms, to examine alternatives and to control this processing by putting diverse conditions and restrictions on evaluating objects by pictures without explicit description of the processing sequence. Specifically, some operations on files and systems can be immediately interpreted as operations on dictionaries.

The system of units in LICOL is defined by a system of linguosemiotical functions, i.e. it is necessarily close to the structure of functions of natural language, specific features of programming are taken into consideration. This allows to proceed from expressions in a natural language to expressions in LICOL, i.e. the highest function is fulfilled, the metalanguage function, the existence of which is ensured by the fulfilment of lower functions: the cognitive one, the communicative one etc.