[From: Internat.conf. *Methodology & Techniques of Machine Translation*, British Computer Society, Cranfield, 13-15 February 1984]

Control and Data Structures in the MT System SUSY-E

Heinz Dieter Maas

Universitaet des Saarlandes
Saarbrueken, FRG

ABSTRACT

The MT system SUSY-E which has been developed since 1972 in the Sonderforschungsbereich "Elektronische Sprachforschung" of the University of the Saar can be divided into three major subsystems: background, dictionary and kernel systems. The background system represents the interface to implementers, linguists and users. The dictionary system supports the construction and maintenance of the different dictionaries and provides the description of the dictionary entries. The proper translation processes are carried out by the use of the kernel systems containing the linguistic knowledge in different representational schemes and allowing for syntactico-semantic analysis and generation of texts. The most elaborate kernel system of SUSY-E is SUSY which has been constantly developed and tested in the past ten years. Apart from SUSY there exist several new "prototypes" which in their architecture show considerable differences between themselves and especially with regard to SUSY. These new approaches are called SUSY-II systems.

The different variants of SUSY-II are based on a common data structure, the so-called S-graph, which essentially is a chart. By defining dominance and neighbourship relations it is possible to represent the sequence of constituents of phrases as well as their internal structure (in the form of labelled trees).

In contrast to SUSY's data structure (which is organized as a network, but has difficulties in representing sequences of constituents) SUSY-II operates exclusively on trees and sequences of trees - at least from the linguist's point of view. An important advantage of the S-graph is the possibility to represent naturally lexical and structural ambiguity. Moreover, the S-graph is the basic structure of all subparts of SUSY-IT, whereas in SUSY a heterogeneous set of data structures is used.

An even more important difference between the kernel systems exists with respect to control structures. In SUSY, the control over the analysis modules is totally programmed

and therefore in principle unchangeable. Only minor changes can be achieved by parametrizing rules or sets of rules or by switching off whole modules. In SUSY-II we have created the possibility of describing the control over all analysis operations by the use of a special formal language. In this way the analysis can easily be adapted to special text types (e.g. instructions, headlines etc.).

In constructing a SUSY-II control structure we will distinguish the following elements of the control language: rules, operators, and modules.

1. Rules: They contain the elementary linguistic knowledge. The left hand side of a rule is always a sequence of 1-4 tree structures. If this description matches the actual data structure, the rule delivers normally one new tree. All these rules are programmed. They do not consider any context or competing structures, and are therefore much simpler than SUSY rules.

2. Operators: Each operator names exactly one rule, together with the conditions under which this rule should be applied. Left and right context can be specified, as well as the mode of application of the rule: substitution and addition. An operator can be iterative: in this case it will be applied as long as it produces changes in the data structures.

3. Modules: Each module names a sequence of modules or operators. It can be stated under which circumstances the module should work, and whether it is iterative. A sequencing parameter allows the specification of three different modes of processing of the submodule sequence:

    a. preferential: the n-th process stops, when its preceding submodule returns a result (n≠l).
    b. stratificational: the n-th submodule will be activated only if the (n-l)th has delivered a result (n≠l)
    c. unconditional: the submodules are applied in sequence.

The control language provides the linguist a comfortable tool for the description of his analysis process by specifying a control tree whose nodes are modules (non-terminals) and operators (terminals) . Apart from the control structure, the user has to define a formal description of the possible content of the nodes of the analysis trees. These properties are related to the conditions stated within the modules and operators. This description is used for the "compilation" of the control tree, which results in a compact control structure that can be interpreted by the SUSY-II software system in a comfortable way.

The advantages of the SUSY-II variant which allows for separate definition of the control mechanism consist in an increased flexibility in constructing analysis processes and an easily readable documentation of its architecture. As compared to SUSY, SUSY-II is certainly less efficient as far as runtime is concerned. The main reason for this disadvantage, however, is not the flexible control structure definition, but the necessity of using additive (i.e. non-deterministic) operators.