

Towards a Multilingual Natural Language Understanding Interface *

(Extended Abstract)

T. HORVATH, Z. ALEXIN, T. GYIMOTHY

Research Group on the Theory of Automata at the
Hungarian Academy of Sciences,
Szeged, Aradi vértanúk tere 1, H-6720

K. FABRICZ

JATE University,
Szeged, Egyetem u. 2, H-6722

This abstract is about the experience obtained from a series of investigations into the creation of natural language understanding interfaces. Our previous reports provide an account of the efforts to establish a general meta-level specification for natural language interfaces. The present contribution tackles the problem of developing an interface that is capable of processing input from various languages.

A step forward in achieving better performance and ease of use is the implementation of **natural language understanding interfaces** (NLUIs), i.e. interfaces that provide for a full lexical-semantic and syntactic coverage for a specific domain. Natural language understanding interfaces [NLUIs] differ from other kinds of interfaces based on natural language in that they work in a domain whose semantics is fully specified for the given interface. In other words, a natural language **understanding** interface contains all information necessary for manipulating a complete subset of the language involved.

From the theoretical point of view, such an interface can be seen as a formal model with full semantics for a specific domain. Therefore, the creation of NLUIs is an important contribution to knowledge representation and comprehension modelling. The experiment conducted has been based on the THALES program package.

The experience gained from designing and implementing THALES has led to the following conclusions. A basic question in designing NLIs of the THALES type is that of assessing which parts can and which cannot be generated. We argue that the parts of a NLI which can be **generated** from some high level specification appear to be those of the lexical, syntactic and static semantic analyzers, prepared on the basis of an attribute grammar description. Of crucial importance is the notion

of static (compile time) semantics. We define static semantics for a conventional programming language as the set of context-sensitive properties that can be evaluated, checked or calculated during compile time. The most important properties of this kind are: *a)* the differentiation between defining and applied occurrences of variables; *b)* the assessment of the scope of defining occurrences; *c)* the problem of identification: finding the defining occurrences that match the applied occurrences; *d)* the question of type compatibility.

Following on research into the design and implementation of the English natural language interface, we tried to work out interface modules for other languages. At present, we can compare the Hungarian module with the interface for English. The conclusions we have arrived at so far can be briefly summed up as follows.

The evaluation of base attributes for linguistic structures is *a)* little dependent on the application; *b)* grossly language-specific (due to relatedness to a specific syntax), whereas the attributes themselves are fairly universal.

Symbol table manipulation is

- a)* little dependent on the specific application since in a new application only type and relation definitions have to be rewritten (see section 4.2.1);
- b)* highly generalizable because only that part of type and relation definitions has to be rewritten that is language-specific (object names, adjectives etc.)

Dynamic semantics is

- a)* largely dependent on the specific application;
- b)* fully independent of the language concerned.

* The authors gratefully acknowledge the financial support of Cogito Ltd., Philadelphia