

# Toward the Interoperability of Language Resources

July 13-15, in conjunction with the 2007 LSA Summer Institute



## Position Paper

Mike Maxwell  
CASL/ U MD

Interoperability can be addressed from synchronic and diachronic perspectives. From the synchronic perspective, one can ask whether the lexical, grammatical or phonological description of some language can be processed by existing tools, and whether the input and output of language data expected by those tools can be processed by other existing tools. The same issues arise in the diachronic perspective, where one asks whether the language descriptions and language data used by existing tools will be interpretable by tools yet to be created.

The issues of interoperability as applied to lexical and annotated text data were considered by Gary Simons and Steven Bird in their paper "Seven Dimensions of Portability for Language Documentation and Description." Not discussed by Simons and Bird are how these issues apply to grammars. Descriptive grammars have been produced for thousands of years, and certainly the grammars that have been and are now being produced are portable, so long as someone can read the language they were written in. Nevertheless, such prose grammars fail to meet several desiderata for language description, including lack of ambiguity, testability for completeness of coverage, and testability for accuracy of coverage--as anyone who has attempted to use such a grammar to build a morphological or syntactic parser knows.

These deficits of prose descriptive grammars cannot be made up for by corpora, because the corpus cannot be easily tested for adequate coverage of the constructions in the prose grammar.

In the last few decades, it has become possible to build computationally implemented grammars, in the form of formal morphological or syntactic grammars which can be loaded into a parsing engine and tested over a corpus. These grammars, expressed in the formal but proprietary language used by this or that parsing engine, have their own set of drawbacks: they are difficult for humans to understand, particularly by someone who is not already familiar with the particular parsing engine's programming language; and they are as ephemeral as the parsing engine that they run on. The second drawback could be addressed by building such grammars in some non-proprietary formalism, perhaps expressed in XML; but the first drawback, while it might be ameliorated in such a formalism, is not entirely avoided.



Fortunately, the advantages and disadvantages of prose descriptive grammars and formal grammars are in complementary distribution. This suggests that by combining the two, the strengths of each could make up for the weaknesses of the other. Exactly such a combination of strengths was introduced by Donald Knuth as a means of documenting computer programs, in the form of literate programming. Literate programming can now be done with DocBook, an XML format widely used for writing books in an open format, which lends itself to grammar writing. At the Center for Advanced Study of Language at the University of Maryland, we are now engaged in a project to combine prose and formal grammars using Literate Programming, and to convert the formal grammars automatically to the programming language of one or more parsing engines. The formalism is intended to integrate with existing standards (or de facto standards) for lexicons. I hope by this to overcome the interoperability problems that exist with current parsing engines, as well as to provide a better way to describe the grammars of languages for future generations of linguists, thus addressing both the synchronic and diachronic aspects of interoperability.

