

ISO/IEC JTC 1/SC 22/WG 9 N 465

Response of ISO/IEC JTC 1/SC 22/WG 9 to SC22 Resolution 5-15

Forwarded by James W. Moore, Convener, SC 22/WG 9, James.W.Moore@ieee.org

At its September 2005 plenary meeting, SC 22 approved the following resolution:

Resolution 05-15: Grammar Engineering Project

JTC 1/SC 22 notes "Grammar Engineering" is a project being run from the Free University of Amsterdam that is of relevance for existing and future JTC 1/SC 22 projects. JTC 1/SC 22 instructs its subgroups and project editors to:

- review this work, available from <http://www.cs.vu.nl/grammars/browsable> and discussed in JTC 1/SC 22 N 3977;
- investigate areas for co-operation; and
- report at the 2006 JTC 1/SC 22 Plenary upon progress made and issues found.

For SC 22/WG 9, the review was performed by Randall Brukardt, the project editor of ISO/IEC 8652, Programming Language Ada. His report is appended.

Report on "Grammar Engineering", as requested by SC22 Resolution 5-15

Randall Brukardt, Project Editor, ISO/IEC 8652:1995

The Free University of Amsterdam is working on a project entitled "Grammar Engineering". SC22 Resolution 5-15 directed working group editors to a web site:

<http://www.cs.vu.nl/grammars/browsable>

The site explains a process for creating "browsable grammars". The primary purpose appears to be to provide a cross-linked grammar for the purposes of human understanding of the syntax of the language. A partial example of a "browsable grammar" for Ada95 (ISO/IEC 8652) is provided on the site.*

I believe that many of the capabilities offered by these "browsable grammars" are in fact already offered by the Ada standard (ISO/IEC 8652:1995) and available tools based on the standard. It is my opinion that any missing capabilities would be better provided in future versions and formats of the Ada standard, rather than in a separate place likely to be overlooked.

When the 1995 version of the Ada standard was created (hereafter called "Ada95"), the contractor used a complex set of Scribe macros to create the document. These macros automatically constructed a number of Annexes to the standard. One of these annexes is a

* This example mistakenly includes the informational prefixes as part of the non-terminals; thus it has many undefined non-terminals (termed "bottom sorts").

complete syntax summary of the language. The syntax summary also includes a cross-reference index which shows all of the uses of every non-terminal. Both listings in the annex reference the body of the standard. In addition, all of the non-terminals and their uses are included in the main index for the Ada95 standard. This provides a relatively compact representation of the syntax of the standard, as well as a number of useful indexing possibilities. (In the Ada "browsable grammar" sample, the syntax summary and cross-reference index are provided, without the references to the body of the standard, as "context free grammar" and "all sorts," respectively.)

The Scribe macros have now been replaced by an Ada program, which provide more flexibility in creating and maintaining the standards. (Currently, the Ada standard is maintained by developing Corrigendum and Amendment documents; there have been one of each since Ada95 completed standardization in 1995). Various organizations have used these tools to create unofficial versions of a merged syntax summary, including some in HTML. These syntax summaries reference the appropriate provisions of the standard.

Some of the other features provided by the "browsable grammars" were considered for the Ada standard but rejected in favor of other approaches. For instance, syntax diagrams were considered, but they were found to obscure the relationship between the syntax and actual Ada text. Organizing the syntax in a similar manner to the way Ada text is written proved to be more useful to the reader of the standard. In particular, the way the syntax of the Ada language is written implicitly provides a recommended style for indentation and organization of statements, subprograms, and the like. These conventions would be lost in the diagrams and flat syntax provided in the "browsable grammar" sample.

The "browsable grammar" samples distinguish between lexical and context-free syntax. The Ada95 standard does not do this; such a separation was considered to be artificial. The division between lexical and higher-level grammar depends on the technology used to implement a standard, and this is inappropriate to specify in the standard. Ada simply defines a set of character classes and lexical symbols, and everything is built from those.

In my opinion, the style of the samples is unattractive. I believe that to be especially true of the syntax diagrams; the construction of the diagrams from text (rather than HTML tables or graphics) makes them somewhat difficult to read. Modern audiences are used to more elegant documents.

Some of the information in the "browsable grammars" seems to have little value for a standard itself. The grammar statistics would have no place in a standard. The list of "top sorts" (start symbols) should be very short, and "bottom sorts" (undefined non-terminals) should be empty; these provide no information to the reader of a standard. Probably, the most important use of these lists for the purpose of standards development would be to detect errors during the maintenance of a standard.

I realize that there may be other uses for such grammars outside of the context of standards (such as for comparing programming languages); I did not attempt to evaluate those uses.

In summary, I already use tools similar to those of the browsable grammar in maintaining the Ada standard. I am pleased, though, to see an effort to make similar facilities available to maintainers of other language standards. I would prefer to continue using my current approach because it is appropriately specialized to the needs of Ada and because its

implementing tools are linked with other tools used to maintain the language. Of course, I stand ready to co-operate with an SC 22 project should one be initiated. Under appropriate circumstances, it might be possible to make my tools available to such a project.