Assembling unlooped substituted SSL files from SSL files

SSL files are parsed as follows:

(1) All variable substitution and file insertions are carried out, resulted in a substituted SSL file
(2) All loop operations are performed, resulting in an unlooped substituted SSL file
(3) The unlooped substituted SSL file is compiled

Here's how (1) works. The parser starts with a queue containing all lines in the input file, an output file (initially empty), and a list of variables together with their values (initially set in the original SSL call). Each line is parsed, in order, as follows:

(i) Variable substitution is performed on the line: that is, each defined variable (those in the variable list) is replaced with its value. (The order in which this is done doesn't matter for any extant SSL code.)
(ii) If the resultant line has the form VARIABLE(scsvariable=scsvalue), the variable scsvariable is added to the variable list with the value scsvalue
(iii) If the resultant line has the form INCLUDE FILE(file), the parser looks for file. If it doesn't find it, the install fails. If it does find it, file.ssl is appended to the top of the queue.
(iv) If the resultant line doesn't satisfy either (ii) or (iii) (that is, if it's neither a VARIABLE nor an INCLUDE FILE command) then the line is just appended verbatim to the bottom of the output file.

Note that the VARIABLE and INCLUDE FILE commands are processed in parallel. So

```
VARIABLE(cat=dog)
INCLUDE FILE(cat)
```

and

```
INCLUDE FILE(cat)
VARIABLE(cat=dog)
```

give different outputs.

I assume the rules for performing loop operations are simple enough that I don't have to explain them. But note that because loops are evaluated after file inclusion, something like

BEGIN LOOP(scsvar||cat;dog)

INCLUDE FILE(scsii\animals\scsvar.ssl)

END LOOP

doesn’t give quite what you might expect. There is actually a BEGIN/END OUTER LOOP command in SSL, which is evaluated before inclusions, but it was basically obsolete once VARIABLE was added, and I don’t think any SCS script uses it.