technical contributions

PUN-DORA'S BOX or HOW TO PRODUCE ENOUGH PAPERS TO WRAP UP STRUCTURED PROGRAMMING

H. Witt
Siemens Aktiengesellschaft
ZFE FL SAR
Postfach 70 00 76
D-8000 Muenchen 70, Germany

It has been long known that movie titles follow certain substitution patterns and can be easily generated by use of a macro processor,
e.g., "monster-1 MEETS monster-2"
"THE monster-1 RETURNS"
"THE BRIDE OF monster-1"
"THE SONS OF monster-1"
"THE TESTAMENT OF monster-1"
(where for monster-1 you may substitute any of the following:
"FRANKENSTEIN" "DRACULA" "KING-KONG" "THE WOLF MAN" "TARANTULA"
or the name of any user-defined monster. monster-2 is also taken from the same set with the additional constraint that it must be different from monster-1).

If the macro processor allows for nested calls, nested titles may be produced:
"THE RETURN OF THE SONS OF THE BRIDE OF monster-1"
"THE RETURN OF THE TESTAMENT OF THE BRIDE..." etc. ad nauseam.

The time has come, I think, to apply a similar technique to the production of papers on structured programming. The starting signal for me was Paul Abrahams' paper titled "Structured Programming Considered Harmful" in SIGPLAN Notices.

The inverse of "stepwise refinement" is, of course, "stepwise coarsening".

How about the following two templates:
"monster-1 CONSIDERED monstrous", and
"monster-1 DEPLORED",
where monster-1 denotes any of the following:
"S.P."
"FINITE STATE AUTOMATA"
"FIBONACCI NUMBERS"
"STANDARD INTERFACES"
"TEXT BOOKS"
"RECURSION"
"ASSIGNMENT TO GLOBALS"
"COMMITTEES"
"GARLIC"
"ACRONYMS"
"THE CHIEF PROGRAMMER TEAM"
"AUTHORS OF PAPERS ON S.P."
and any other user-defined monster.

For "monstrous" you may substitute any of the following:
"BENEFICIAL" "A NUISANCE" "HARMFUL"

You have probably seen by now, how the basic mechanism works. I forgot to mention that one additional constraint of choosing the template is, that it must be possible also to produce all already existent papers on S.P. Lets try a few more simple templates:

"THE so-and-so PROGRAMMER"
Here for "so-and-so" you may insert:
"ARROGANT"
"VICTORIOUS"
"PROUD"
"MODEST"
"WELL-BEHAVED"
"INOFFENSIVE"
"THUNDER-STRUCK"
"HORRIFIED"
"HUMBLE"

"TEACHING THE illness"
Here "illness" stands for any kind of serious medical condition (to find out what this has to do with S.P. is an exercise left to the reader).

I think "monster-1 WITHOUT TEARS" and "ASSIGNING MEANING TO monster-1" need no further explanation. If your macroprocessor allows for nesting, you get more interesting structures like "ASSIGNING MEANING TO STRUCTURED PROGRAMS WITH THE GOTO STATEMENT CONSIDERED HARMFUL". In some cases additional flexibility is provided by allowing for optional inserts "IT IS /¯NOT_7 /¯NOW_7 TIME TO DEFINE monster-1". A final exercise for the reader:

try to prepare a parameter list for:

"A monster-x BUILDING SYSTEM WITHOUT THE monster-y".

SOME USER-DEFINED CONTROL FUNCTIONS FOR MORE READABLE APL PROGRAMS*

James A. Mason
Department of Computer
Science and Mathematics
Atkinson College
York University
Downsview, Ontario

Abstract:

Some improved control statements for APL are presented, which the user can define and use without requiring either preprocessing or modification of the interpreter. A preprocessor is also given, however, in case improved execution time of programs using the control statements is desired.

Introduction:

The APL language does not permit one to write programs with the kinds of control structure advocated by proponents of 'structured programming'. In particular, APL lacks convenient statements for the control of iteration (e.g., a 'do-while' construction) and for conditional execution (e.g., an