not in its results, but in its methodology. The system is designed
on the basis of intuition, rather than analysis, and the authors do not
even pay lip service to the ideal of evaluating the system by pro-
gramming real problems on it.

It is rather disappointing that the authors' only "evaluation" of
their proposed system consists of counting the number of switches
(about 27,000) and pronouncing it "reasonable."

Since the paper under review is only a seven-page summary of an
M.S. thesis, one might expect that the detailed analysis and evalua-
tion were merely omitted from the summary for the sake of brevity.
This is not the case. The original thesis suffers from the same limita-
tions. It appears that neither the paper nor the thesis itself makes a
substantial advance in the state of knowledge about this problem.

GEORGE HANNAUER
Electronic Associates, Inc.
Princeton, N. J.

B. MULTIPROGRAMMING

R68-52 The Structure of the "THE"-Multiprogramming System
See also Computing Rev., vol. 14, p. 979.

This paper is worthy of attention. Written with the facility one
has grown to expect from Prof. Dijkstra, this is an announcement of
a new operating system designed and constructed at the Technologi-
cal University at Eindhoven for the ELX8.

While the operational characteristics of the new system are not
unusual, the claims for the confidence level of both the design and im-
plementation of the system certainly are. "We have found that it is
possible to design a refined multiprogramming system in such a way
that its logical soundness can be proved a priori and its implementa-
tion can admit exhaustive testing." For one of Dijkstra's stature in
the community to make such a statement is to demand the considera-
tion of all of us.

The author's abstract and his "key words and phrases" provide
a terse summary of his announcement:

A multiprogramming system is described in which all
activities are divided over a number of sequential processes.
These sequential processes are placed at various hierarchi-
cal levels, in each of which one or more independent ab-
stractions have been implemented. The hierarchical
structure proved to be vital for the verification of the logi-
cal soundness of the design and the correctness of its imple-
mentation.

Key words and phrases: operating system, system
hierarchy, system structure, real-time debugging, program
verification, synchronizing primitives, cooperating sequen-
tial processes, system level, input-output buffering, multi-
programming, processor sharing, multiprocessing.

Professor Dijkstra is quite precise in his abstract. Some amplifica-
tion is offered below.

Two unusual claims are made for the THE system.
1) Its logical soundness was proved a priori (i.e., before imple-
mentation).
2) Its implementation admitted exhaustive testing.

Consider them in turn.

A Priori Proof of Logical Soundness: In the THE system a set
of abstractions are realized in an hierarchy of "levels." These ab-
stractions concern parallel processes, virtual memory, input-output
streams, and other useful facilities. Though each abstraction is im-
portant in that it realizes some facility in the context of resource
restrictions, the mere concept of hierarchy, in conjunction with
parallel processes and the means for synchronizing their operation is
fundamental to the objective of proof of design for the system.

Designed well within Habermann's universe, the THE system
enjoys the benefits of "harmonious cooperation." One aspect of such
cooperation is the avoidance of "deadlock," or "the deadly embrace,"
one of the bane of designers of parallel-task systems.

But just how does one achieve a priori design? Granted
commitment to a Habermann hierarchy (so as to enjoy the benefits),
just how does one prove that a particular society composed of mu-
tually synchronized processes does indeed satisfy all requirements in
its time behavior? Dijkstra is not explicit here. He presents only the
announcement that he and his compatriots have "learned the art of
reasoning" by which this was possible.

Admission of Exhaustive Testing: Given a correct design, how
does one achieve a valid (correct) realization of it? Dijkstra's tech-
nique is that of dividing and conquering. One relies on the relative
simplicity of functions on a given level, plus the ability to maneuver
a subsidiary level, using cooperating sequential processes, to search
out, establish, and demonstrate the correct operation of the system in
each of its "relevant states." In effect, one attempts to design such
that the combinatorial aspects of the operations are additive, rather
than multiplicative.

The author's argument for exhaustive testing relies on our confi-
dence in the perception of the designers on one hand, and of the
testers on the other. The designers must keep the relevant states on
each level to a manageable number. The testers must identify all
relevant states, and devise tests (also to be verified) that 1) maneuver
the system into each relevant state, and then 2) demonstrate the
correct operation in that state. Detailed arguments are not provided
in Dijkstra's paper.

Bonuses: Two elegant new primitives (the "P" and "V"
functions) for the coordination of cooperating sequential processes are
briefly described in an appendix to the paper.

A brief sketch of the proof of "harmonious cooperation" leads us
to Habermann's work: apparently significant, and hopefully avail-
able in the general literature soon.

CODE WELLS
The MITRE Corp.
Bedford, Mass.

1 N. A. Habermann, On the harmonious co-operation of abstract machines.

C. INTERACTIVE GRAPHICAL PROGRAMMING

R68-53 A System for Interactive Graphical Programming—W. M.

This paper describes a scheme for organizing and creating the con-
trol portions of an interactive graphical program. Since the response
of such a program to some console input may depend on the history
of previous inputs, the author structures his control programs as
finite-state automata. Thus state diagrams are used as the notation
for defining the behavior of an interactive graphical program. Such
notation does indeed add clarity and order to the description of this
information.

The control concepts presented have been implemented in a
working system. A written description of the state and transition