# Errata

Due to an error in the typesetting process, the following abstracts were omitted from the list of invited speakers to the 1988 Rochester Forth Conference, printed in the *Journal of Forth Application and Research*, Volume 5, Number 3, pg. 433. We regret any inconvenience this omission may have caused.

## Forth on Unix Workstations

*Mitch Bradley*

*Sun Microsystems, Inc.*
*Bradley Forthware*

Sun Microsystems is the leading supplier of Unix™ based workstations. This talk focuses on the use of Forth at Sun, and especially the efforts that have led to the current "Fcode driver" project. The "Fcode driver" project attempts to make it possible to plug in a peripheral board, such as a disk controller or display device, and to use that device for booting a computer. The Fcode boot drivers are written in an encoded form of Forth (Fcode), and are independent of the CPU instruction set.

## Infrared Image Acquisition and Analysis in Forth

*William J. Forrest*

*University of Rochester, Rochester, N.Y.*

*Justin Schoenwald*

*Cornell University, Ithaca, N.Y.*

The control and operation of an infrared camera using a 32 x 32 cooled InSb array interfaced to a DEC LSI 11 computer is discussed. Control of the memory management hardware in Forth enables up to 4 Megabytes of computer memory to be used for any mixture of programs or data desired. The special camera software is organized into smaller tasks, each of which is contained in semi-autonomous "modules". The modularization makes development and modification of the code more efficient and rational. Several enhancements, including on-line dictionary lists, decompiling, and debugging, make the system easier to develop for the programmer and easier to learn and re-learn for the user.

## Cellular Automata Machines: A new environment for modeling[†]

*Norman Margolus*

*MIT Laboratory for Computer Science*
*Cambridge, MA*

Cellular automata (CA) are uniform arrays of simple processors with local interconnection—they may be thought of as the computer scientist's counterpart to the physicist's concept of a *field*. Because of their physics-like structure, they can on one hand be efficiently implemented, and on the other hand are well suited to physical modeling tasks.

CAM-6 is a general-purpose CA machine intended to act as a laboratory for experimentation, a vehicle for communication of results, and a medium for interactive visual demonstrations. It fits into a single slot of an IBM-PC compatible host computer, and uses F83-based Forth driving software. Here Forth, which was originally used by Charles Moore to control telescopes, is being used to control scopes of another kind: windows into a variety of synthetic universes.

In this paper, we discuss the use of machines such as CAM-6 as the basis of an environment for developing and analyzing CA models.

[†] This research was supported by grants from the National Science Foundation (8214312-IST), the Defense Advanced Research Projects Agency (N00014-84-K-0099), and from International Business Machines (3260).

## X-Script

*Paul Snow Cliff Click*

*KBSI*

*100 West Brookside, Bryan TX, 77801*

X-Script is a complete implementation of the PostScript™† standard as published and placed into the public domain by Adobe Systems Inc. X-Script is written in Fifth, a Forth-like language. Basic considerations of implementing X-Script are given, with an overview of its design.

## ASYST: A Structured Interactive Environment for Scientists and Engineers

*Sue Semancik David Smith*

*Asyst Software Technologies, Inc.*

The transformation from *ASYST the development language* to *ASYST the end-user language* does not strip ASYST of its Forth-based origins. However some differences do exist, primarily driven by the different audiences of the two systems.

Whereas Forth is a full-fledged language championed by systems level programmers and computer scientists, ASYST is marketed as a set of data acquisition, analysis and graphics tools for the scientist and engineer. These users, found in the research and development staffs of industry and academy, are often not programmers. Some of the differences between ASYST and Forth are aimed at protecting the non-programmer from himself.

## RPL: A Mathematical Control Language

*W.C. Wickes*

In 1984, a project was started at Hewlett-Packard Corvallis Division to develop a new software operating system to streamline calculator development and support a new generation of hardware and software. Previously, all HP calculators were implemented entirely in assembly language, a process that was becoming increasingly cumbersome and inefficient as the memory sizes of the calculators increased. The objectives for the new operating system were as follows:

- To provide execution control and memory management, including plug-in memory;
- To provide a programming language for rapid prototyping and application development;
- To support a variety of business and technical calculators;
- To execute identically out of RAM and ROM;
- To minimize memory use, especially RAM;
- To be transportable to various CPU's;
- To be extensible; and
- To support symbolic mathematical operations.

Several existing operating systems and languages were considered, but none could meet all of the design objectives. We therefore proceeded with the development of a new system, which merges the threaded interpretation of Forth with the functional approach of Lisp. The resulting operating system, known unofficially as *RPL* (for *Reverse-Polish Lisp*), made its first public appearance in June of 1986 in the HP-18C *Business Consultant* calculator. Subsequently, RPL has been the basis for the HP-17B, HP-19B, HP-27S, HP-28C and HP-28S calculators, demonstrating that it meets the objective of supporting a variety of calculators. The HP-17B, 18C, and 19B are designed for business applications; they and the HP-27S scientific calculator offer an "algebraic" calculating logic, and the underlying operating system is invisible to the user. The HP-28C and HP-28S scientific calculators use an RPN logic, and many of the facilities of operating system are directly available as calculator commands.

† PostScript is a trademark of Adobe Systems, Inc.