# Abstracts of the euroFORML 87 Conference

*Stettenfels Castle*
*Heilbronn, West Germany*
*September 18 – 20, 1987*

## Communicating FORTH

*Robert Jones*

*5144 Wegberg*

Forth deserves to be considered to be a fourth generation language because it advances beyond earlier language designs.

But Forth was designed about one decade ago. Does Forth today realize programming language design goals that have come to be seen to be desirable? The objective of this paper is to propose a certain answer to this question, and certain extensions to Forth.

## FURTHer FORTH with LEIBNIZ

*Andreas Goppold*

*Diplom-Informatiker, MSc*
*Hauptstr. 31a*
*D-8170 Wackersberg-Arzbach*
*Germany*

A Forth-83 system on an 80386 or 68020 is like a bicycle with a Porsche engine. In the presentation and live demonstration will be shown how one can restore the right relation.

## `METHODS>` Object-Oriented Extensions Redux

*Terry Rayburn*

*forthlab Unternehmensbereich der RSO GmbH*
*Postfach 82 05 23*
*8000 Muenchen 82*
*BRD*

The `METHODS>` approach allows a layer of object-oriented extensions to be superimposed on Forth systems without modification to the underlying structure such as adding multiple code fields. In typical Forth fashion, the operators and operands are defined by the programmer to suit the application. The `METHODS>` kernel only provides the glue to bind them together.

Experience with the approach since the last presentation at euroFORML 1985 has resulted in simplifications to aid portability and a change of syntax that makes code more consistent and readable.

## LIST: A Generator for Object Oriented, Cyclic Linked Lists

*Karl-Dietrich Neubert*

A generator for object oriented cyclic linked lists is presented together with the words for manipulating these lists. The header of a list contains information on properties of the list such that it may be manipulated with identical words independent on the data structure of the list nodes. The access time to the nodes is optimized and typical run-time examples are given for Forth 8088 and NC4016 implementations.

## A LisP-Kernal for the NC4000

*U. Hoffmann*

This article describes the basics on implementing LisP on Forth Machines especially on the NC4000, the only Forth Machine available in these days for a reasonable price. It will regard the similarities and differences of LisP and Forth. The sources of a pure LisP-Interpreter running under mpForth is given in public domain.

## Command Interpreter for Peripheral Devices with an Integrated Help Function

*Klaus Schleisiek*

*Eichenstr. 45*
*D-2000 Hamburg 20*

Peripheral devices get more intelligent all the time and the user interface becomes more demanding. The following paper presents an alternative to the classical Forth interpreter that integrates a help facility.

This approach has been derived while developing a peripheral device for communication over the phone line with high data integrity and security.

## cmFORTH, mpFORTH, the FORTHchip, and Optimizing Compilers

*Marco Pauck*

*Friedensallee 92*
*D-2000 Hamburg 50*
*FR Germany*

I will present some experiences in writing my own Forth for the Novix NC4000P (alias NC4016) FORTHchip. This Forth emerged from cmFORTH and is in the public domain. Problems dealing with cmFORTH in particular and optimizing compilers in general will be discussed.

## The MARC4 — A FORTH Based Single Chip Micro-Computer

*Graham Stout*

*EUROSIL electronic GmbH*
*D-8057 Eching near Munich*
*West Germany*

A new four bit microcomputer, with a core processor tailored to the Forth programming language and a modular approach to on-chip peripherals is presented.

A modern multi-window development system with an efficient Forth compiler and the ease of configuring peripheral modules to the customer's specification, result in a cost effective product without the development time and costs associated with the current four and eight bit microcomputers.

The MARC4 is particularly well suited to low power applications. Special architectural features and EUROSIL's low power CMOS technology allow operation down to 1.5 Volts with a typical power consumption of less than 10μW.

## Trainable Neural Nets in Forth

*John D. Carpenter*

In the course of implementing Chris Maylor's learning algorithm in Forth, I have discovered that the heart of his technique not only follows the Hopfield algorithm but also provides a technique for establishing the weight values therein through training by query and by example. This paper is a tutorial as to how this scheme works in a single node which is like a set of neurons which share the same dendrites but have their own single axons. A scheme for entering examples and training a multinode version through "exercising" is discussed. Some of the unique tools I have developed to study this scheme which have value for general use will be defined. Finally, suggestions for fitting this into a small "Computer Cowboys" type of Novix system for study will be given as well as what a more serious implementation on an array of Novix chips would be like.

## Directed Nets of Rule Sets and a Hybrid Search Strategy

*Thomas Jost*

*Robert Bosch GmbH*
*Geschäftsbereich Industrieausrüstung*
*Abt. IA/EPG*
*7310 Plochingen*

The knowledge a human being uses for solving a certain problem is only a minor part of the total amount of knowledge available. Considering knowledge as sets of rules this characteristic suggests that a search for a solution might be predirected in terms of choosing subsets of rules from the set of rules totally available, i.e. to manage the association of sets of rules with given problem specific informations.

This paper presents a way of structuring knowledge to model this behavior. Since the way knowledge is organized strongly influences the structure of the inference procedure a suitable inference procedure is also presented.

Both, the representation concept as well as the inference procedure has been developed and tested in the Forth dialect Mach 2 under Macintosh Plus. The center of the representation concept is constituted by the calculus of propositional logic but it by no means restricted to them. However, the system has been developed for the purposes of interactive diagnosing processes in technical areas like car motor diagnosis and the declarative nature of propositional logic rises special problems concerning the system behavior during diagnosis processes. The problem will be discussed in this paper and a solution will be presented.

This work is a result of a study for specifying the requirements for automated diagnosis in technical domains and testing appropriate concepts to meet these requirements.

The paper consists of three parts. The first part introduces a concept for knowledge structuring, the second part introduces an appropriate inference procedure. Finally, the third part discusses some Forth specific implementation details.

## Implementation of an Expert System in Forth

*Dr. Manfred Walter*

*Robert Bosch GmbH*
*Geschäftsbereich Industrieausrüstung*
*Abt. IA/EPG*
*7310 Plochingen*

An expert system, written in Forth, will be presented, which was developed especially for automotive diagnosis, but can also be used in other applications. It uses production rules in a quasi natural language form. The sequence of hypotheses is controlled by heuristic valuation numbers. The system updates these valuation numbers automatically in case of a successful diagnosis.