Abstracts of euroFORML '89

October 13 – 15, 1989
Neunkirchen am Brand
Federal Republic of Germany

A Modular Approach to Robotic Control Systems

Paul E. Bennett
25 Court Road
Horfield Common, Bristol BS7 0BU
Tel: 0272-510232
Telex: 9312130658 (SW G)
Fax: 0272-522266 (Agency)

This paper is intended to be a review of the drive control systems used in the robotics field and how a modular approach, such as used in the design of Forth systems, can be extended to the hardware end of the design process.

Many of the products currently available on the market provide a lot more processing power and are capable of controlling a motor on their own, given the information to operate the motor, until a preset condition is attained. However, motor protection and end limits detection is usually minimal. Position accuracy is dependent on the controlling computer system and not local to the drive.

With a distributed drive control system any one drive controller must be capable of looking after the mundane task of protecting the drive itself and knowing the maximum range through which motion is permitted. Thus the controlling computer system has only to demand a position be achieved and expect only a status report in return to ascertain the completion or failure of the task.

Two Levels of Parallelism or A New Approach for Control Systems Design

L. Blagoev, M. Sc.
L. Nyagolova, M. Sc.
B. Makaveev, M. Sc.
E. Michailov, M. Sc.
Institute of Industrial Cybernetics and Robotics
Bulgarian Academy of Sciences
1113 Sofia, Bl. 2 Acad. G. Bonchev Str.
Bulgaria

The paper considers a new approach for control systems software design. This approach is based on exploiting of two mechanisms for parallel processing - multitasking and parallel processing on a technological program's level. The principal of parallel processing is described and the differences between the two variants are pointed out.

Software Development System USW or A Way to Think for Control Systems Design

L. Blagoev, M. Sc.
L. Nyagolova, M. Sc.
B. Makaveev, M. Sc.
Institute of Industrial Cybernetics and Robotics
Bulgarian Academy of Sciences
1113 Sofia, Bl. 2 Acad. G. Bonchev Str.
Bulgaria

The Software Development System USW is a powerful software development environment for control systems software design especially designed to cover the different hardware needs of different producers. It integrates the advantages of the language system Forth with a new concept for real time parallel data processing.

USW system is not only a tool, it's an Ideology for control systems design.

The basis of this ideology is a five level hierarchical scheme for the software. It represents the exact distinction between technologically independent and technologically-oriented program constructions.

The first language level includes the basic language system Forth. The concrete Forth-version is adapted to a given hardware to satisfy the requirements of parallel processing and to provide some development tools. Modifications do not concern the inherent functional features of the basic language model.

Zug Power Station An Application of Trainable Neural Nets

John D. Carpenter

The Zug Power Station is a demonstration of the sort of application that can be implemented with trainable neural nets. The power station is a simple fictional concept involving a realistic environment and problem. A copy of the software which operates under F-PC version 2.25 will be available to experiment with. Included are other "warm-up" applications called COOKIE, which tells a cookie-holic when and under what circumstances he may have his cookie, and TAPE, which is a hypothetical troubleshooter for a cassette tape recorder. The application in this paper is PWR, which looks at power demand and puts power sources on or off line accordingly.

Memory Cards and Forth: An Update

Paul Frenger M.D.
A Working Hypothesis, Inc.
P.O. Box 820506
Houston, TX 77282-0506 USA

Semiconductor memory cards consist of ROM, EPROM, EEPROM, or battery-backup RAM with decoding and support circuitry in a credit card sized form factor. Since their introduction, the cards have been put to a number of interesting uses, including Forth-related projects. Many of these applications are described in this paper.

A Comparison of the Cooperative and Preemptive Concurrent Scheduling Algorithms

Harvey Glass
University of South Florida

We examine and compare the cooperative and preemptive scheduling algorithms in the design of real-time embedded systems. The advantages of each of the techniques is discussed and evaluated.

XShell: A Cross-Development User Interface

Roy Goddard BSc
Micro Processor Engineering, Ltd.

This paper looks at the nature of a PC-based Forth system and its integrated tools and compares this situation to that found for the typical target development system. It then introduces the XShell package from MPE as a solution to the problems of cross-development, describing some of the techniques and strategies used to implement the user-interface, and some of the background. The benefits of such a package, both commercially and technically are described, and some conclusions drawn about the future of Forth development environments.

dbgx
The Harris RTX 2000 C Language Debugger

Tom Hand
Harris Semiconductor
Melbourne, Florida 32904

This paper describes the features, capabilities and interface of dbgx, the symbolic source statement debugger, for the Harris RTX 2000 C Compiler. The RTX 2000 is a highly integrated, high performance microcontroller from Harris Semiconductor. The RTX 2000 C Compiler is for developers of embedded real-time systems who prefer the C programming language as their language for implementation.

The RTX 2000 C Compiler is a very efficient cross compiler that was designed to meet the proposed ANSI Standard for the C Programming Language.

The C compiler has a state-of-the-art window/menu user interface with an embedded help facility.

Module Forth

Urich Hoffmann
DELTA t GmbH
Uhlenhorster Weg 3,
D-2000 Hamburg 76, FRGermany

Large Software systems require modularization for sake of manageability. Traditional Forth systems do not support code organization beyond the vocabulary mechanism. The wordset `MODULE <name>`, `EXPORT`, `IMPORT` and `END-MODULE` is defined, that realizes the functionality which is found in contemporary languages like MODALA and ADA.

An RTX Recompiler for On-Line Maintenance

Jonathan Lee
MicroProcessor Engineering Ltd.

This paper describes an RTX software system which provides for the decompilation, editing and redefinition of previously compiled code. This system allows the programmer to fix bugs and alter the behavior of any existing code while interactively using the system. In short, what has been produced is a Forth recompiler.

A Vision of Transparency

Stephen Pelc
MicroProcesor Engineering Ltd.
133 Hill Lane
Southampton SO1 5AF England
tel: +44 703 631441
fax: +44 703 339691

This paper considers the traditional view of Forth systems, as unified programming environments, and relates this view to the design and implementation philosophies of newer Forth systems, leading to transparency as a design goal which governs implementation techniques. In particular, the use of packages imported from other languages is considered.

A Databased Forth
Introduction to a New Conception of Forth

Frank Raschke

Up until now we knew several types of source-text-management like block- or structured stream-files.

The disadvantages of these conceptions are not to be listed here, but there is one problem that forced us to a new idea. If you try to edit a word definition, you have to know in which file or screen it has been defined. Only when it has already been compiled, there are some utilities such as fix or see which offer you a direct way of obtaining your data.

So we came to consider a new type of **Word-Editing** instead of text-management.

Interactive Remote Compilation for Development and Machine Integration

Alan M. Robertson BSc
RAM Technology Systems Ltd.
POOLE, Dorset BH17 7PD, England

The IRTC route to application generation has many benefits over the more conventional approaches. Specifically:

- The availability of the full Forth resources to simple Target hardware.
- Simplified testing of complex and simple applications alike.
- Only one compilation process.
- Fast generation of new environments without the need to recode the whole of Forth.
- Allows the testing and debugging of applications on the shopfloor. Even via the telephone system.

The Development of an Ocean Bottom Seismograph (OBS) for Refraction Seismic Data Using a DAT-Recorder (Digital Audio Tape)

Karsten Roederer
Institut für Geophysik
Universität Hamburg

Since 9 years the German research foundation (DFG) sponsors the development of Ocean Bottom Seismographs (OBS) at the Institute of Geophysics, University of Hamburg. This support is not only a financial support, but also a personal position for a specially electronic Engineer is sponsored. It that way it was possible to develop in a first step analog recording OBS, for more than 100 hrs recording time. During a period of national and international research experiments, great experience in handling, deploying and recovering these instruments was achieved. On the base of the analog system a digital-recording unit was developed, where the recording instrument is a DAT-recorder with a capacity of 1.3 GByte. Now continuously recording with great dynamic and high resolution for all seismic tasks is possible.

Have Dot-IF Dot-ELSE Dot-THEN

Klaus Schleisiek-Kern
DELTA t GmbH
Uhlenhorster Weg 3,
D-2000 Hamburg 76

Writing software under real life conditions reveals the need to write equivalently executing programs under slightly different hardware and/or software conditions. The intuitive approach might be to make a copy of the original program that was written for machine A and adapt it to machine B. Then while you are busy working on the program for B adapting it to the new machine you will invariably make "enhancements" to the program which would also benefit machine A – but, unfortunately, its source code is somewhere else.

Programming and Our View of Man

Jens Storjohann

It is striking one's attention that the discussions between supporters of different programming languages do not appear like the exchange of arguments about the solution of a technical problem but seem more like the first stages of a religious war which happen to take place only between true believers and some infidels.

At first I thought that this is due to lack of knowledge or character on the side of the fighters. But today I think they are (perhaps unknowingly) right. If one puts the emphasis on the word language it is clear that we are talking about a means for human beings to express their thoughts (and more), not just about a method to make a machine work.

If we accept this, and Forth programmers would usually do, we come to the conclusion that choosing a programming language or system is based on a questions about the nature of man. Here religion, basic philosophical convictions or "Weltanschauung" come into play.

Some Experiences on Implementing Floating-Point in Forth

Frank Stüss

In some applications the problem exists how to use the comfort of floating-point without having much loss of speed. The solve of this problem in software is possible if you don't expect much accuracy. Here, some experiences on implementing a software-flp. with 16-Bit mantissa and 12-Bit exponent are shown.

RTX 4000

Philip Koopman
Rick VanNorman
Harris Semiconductor

Harris Semiconductor has begun design of the RTX 4000, a 32-bit member of the Real Time Express (RTX) product family. The RTX 4000 will be a 32-bit stack processor optimized for real time control applications, with similar design objectives as the RTX 2000.

Object Oriented Menu Creating System for Interactive Forth Applications

Konstantin Vassilev Vassilev
Medical Academy
Institute of Biomedical Engineering
1, G. Sofiiski str.
1431 Sofia, Bulgaria

A object oriented generator for menu creating is presented together with the menu manipulating words. Menus are of two types: menu bar and pull-down. Items of menu are located in windows with automatically determined at compilation time width and height. Items are of four types: skipped, active, suppressed and filled. Type of items can be changed in compilation or in run time. Menu item text and menu item help text is associated with each menu item along with screen position where to be displayed. Full code for menu manipulations is given along with associated commentaries.

From Block Files To The Twentieth Century

Andrew Waters
MicroProcessor Engineering Ltd.

This paper argues against the use of Block Files for mass storage for two reasons: One, their non-standard nature is a hindrance to Forth's wider acceptance. Two, they are awkward to use, slowing down development time. An alternative mechanism is presented that combines the best features of Block Files with those of industry standard Text Files, providing Variable Length Paged Text Files.

Not Screens Nor Files But Words

Wolf Wejgaard
Forth Engineering
Neuhöflirain 10, CH-6045 Meggen
Switzerland

A programming environment for Forth is described which is based on a database and a browser instead of screens or files. The collection of words, which constitute the program, is hierarchically structured into modules and groups. The programmer acts on the program through the browser interface, which shows modules, groups, words and definitions in separate windows. The database structure allows useful actions on the program like a hypertext facility, easy factoring of a programs structure, reloading of single words, glossaries and selective documentation.

This method – named Holon – is now in use by the author (for his bread-and-butter work) and has proved very fruitful.

The Search for the Law

The Committee

At EuroFORML 1989, a contest was proposed, by the conference chairman, to define the "**First Law of syntactical software dynamics.**" This law related directly to the Second Law – as described in a paper presented at that very conference.

Entries were solicited from all delegates, and vast was the entry.

A truly international committee was appointed to judge the many and varied entries, and their deliberations went as follows: