in Improving the Quality of Teaching and Learning in Computing, published by SEDA, July 95, pp11-25

THE BROOKER[1] LABORATORY FOR INTELLIGENT EMBEDDED SYSTEMS

Victor Callaghan
Paul Chernett
David Lyons

Department of Computer Science
University of Essex
Wivenhoe Park
COLCHESTER CO4 3SQ
email robots@essex.ac.uk

1.0 Introduction

The breathtaking pace of current developments in computer science brings many challenges to educators. One is to sustain students' interest as they wrestle with the often difficult theory underpinning the scientific and engineering methodologies which will be essential if they are to become effective professionals. Another is to provide a setting in which students can learn to integrate the ever-widening and deepening range of specialisms within the subject.

This paper describes a laboratory based on "state of the art" technology which seeks to provide an innovative *theme-based teaching environment* which facilitates the *integration of courseware* across the entire computer science curriculum and fosters a creative attitude whilst providing a framework that develops theoretical ability, subject knowledge and practical implementation skills. It does this by utilising a unique *combination* of physical systems, networks, simulators and cross development tools, which has a substantially higher cost effectiveness than the traditional laboratory it replaced.

1.1 Teaching requirements

The remorseless advance of technology requires all computing facilities to be periodically updated. During the summer of 1992, whilst planning the refurbishment of the systems architecture teaching laboratory, the idea of replacing this single scheme laboratory by one which could serve and integrate many of the hitherto separate strands of computer science was born. It was suggested that a laboratory based on a mobile robot theme might have the potential to:

- appeal to the imagination of school-leavers
- provide a stimulating environment for students

1. The laboratory is named after Professor R.A. Brooker who was a founder member of this department and is now in very active retirement.

- offer facilities to explore the most important aspects of computer science, thus acting as a unifying framework for the major departmental specializations
- allow assignments to be both stand-alone exercises and sub-systems that could be combined to produce more complex working products.
- accommodate both individual and team-work
- contain state-of-the-art but industry-standard software and hardware
- contain software in which the underlying mechanisms are both safely accessible to students and academically sound
- support student project work which could explore and implement recent research.

Robots are examples of embedded systems (a term which simply implies any product with one or more computers integrated into its structure). As sales of processors for embedded systems now considerably outnumber those for traditional computing applications and as this technology subsumes traditional computer design, its mastery would clearly improve the marketability of graduates. The Brooker Laboratory can support development of the full spectrum of intelligent embedded systems even though, for the purpose of student coursework, the focus is on mobile robots.

1.2 The Autonomous Vehicle Theme

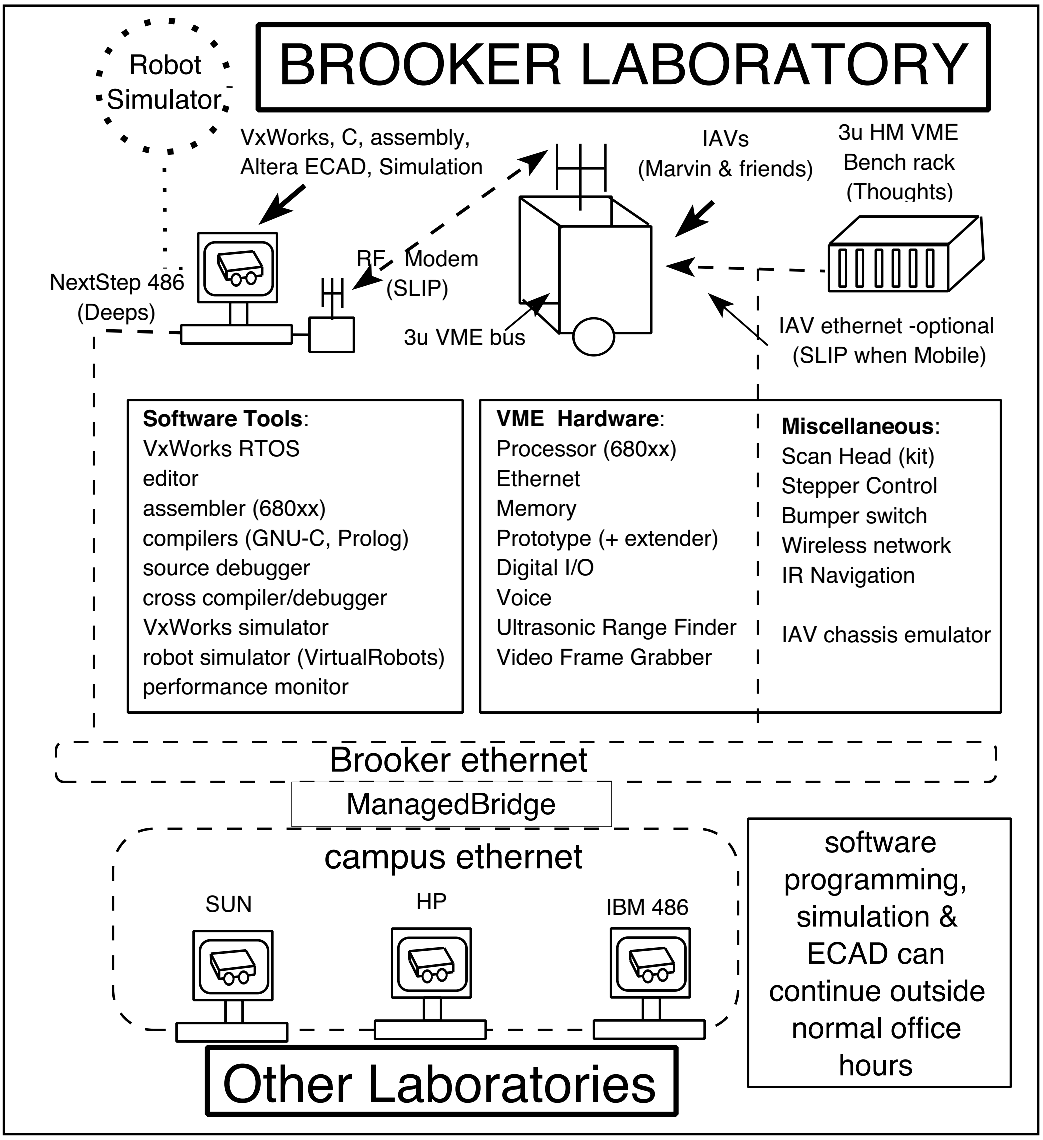
The robotics theme of the laboratory concentrates on the mobility and navigation of robots rather than traditional engineering aspects such as control theory, kinematics or dynamics of motion. In the research world such mobile robots are often more accurately referred to as Intelligent Autonomous Vehicles (IAVs). IAV research involves the study of the whole gamut of computer science techniques including distributed artificial intelligence, machine learning, vision, planning, programming methodology, hardware/software interface, operating systems, data-communications, parallel architectures, reprogrammable and custom chip design.

We have found that the IAV theme is ideal for educational purposes. It is highly motivating in that even the weakest student can produce a software or hardware module that can be immediately seen to function within a larger system. The most able can make a real contribution to a research project.

Unlike many other institutions, IAVs are not used as the primary focus of the teaching but rather as a means to encapsulate some of the practical applications in an interesting and universal assignment framework. Thus, robotics as a field of study is very much secondary to the underlying educational aspects of Computer Science. The complications of design for efficient robots very much take second place to the provision of a solid set of teaching tools.

2.0 An Overview of the Brooker Laboratory

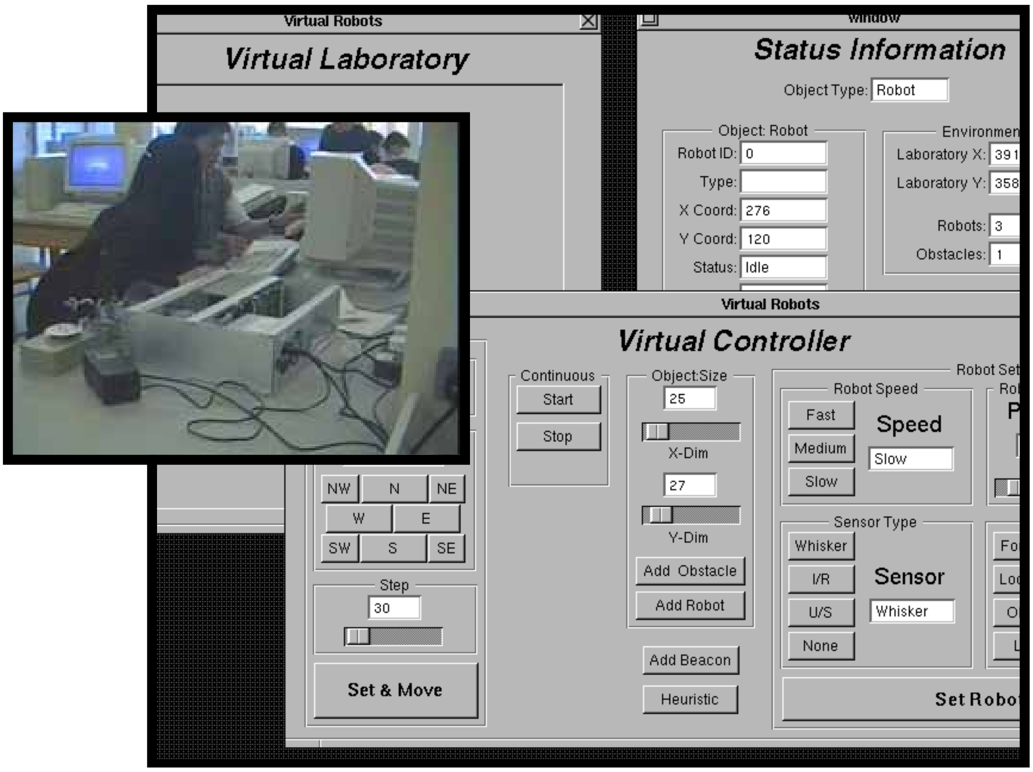
As can be seen in the following figure, the laboratory is composed of **host** computers which provide an environment for program and ECAD development, associated **target** systems in the form of conventional bench racks, as well as wheeled versions of the racks namely **IAVs**.



All machines are connected by conventional Ethernet, although the IAV's, which only require the network for initial program loading, may also use a wireless modem. The laboratory network is connected to the departmental and university network via a very restrictive, managed bridge which acts as a "firewall" to protect the wider network from incompetent or malicious students.

2.1 Host Environment

The bench based host computers can run either NeXTSTEP[1] or Windows under MS/DOS[2]. NeXTSTEP, as well as providing a standard Unix environment and tools, provides a well thought out object oriented language, and a comprehensive set of CASE tools to go with it. All the development and compilation for the target machines and IAVs is done on these hosts.



2.2 Target Environment

The bench targets are small (3U height) industry standard VME based systems[3], which offer a long established technique in system building.

In addition, a box emulating the IAV chassis functions is provided. The software environment is VxWorks[4], a highly modular real time operating system, which is described more fully later in this article.

1. NextStep is a trademark of Next Computer Systems Inc. It incorporates Unix functionality.
2. Windows and MS-DOS are trademarks of Microsoft Inc.
3. Supplied by HM Computing and PEP Modular Computers
4. VxWorks is a trademark of Wind River Systems, Inc.

2.3 IAVs

A few VME racks have been built incorporating motors and sensors to form a fleet of IAVs.

Using VME racks for the robots (together with the cordless network link) enables us to provide a development environment which is wholly consistent (i.e. all hardware and software appears to be identical on the IAVs and bench racks). Thus, the full range of debugging tools can be applied to IAVs, even while they are executing real tasks. The wireless link can also be used to effect remote control or virtual presence.

2.4 The Software Environment

The target and IAV machines run a highly modular, commercial, Real Time Operating System called VxWorks. Compilers, high level debuggers, simulators and the like all live on the hosts. The cross-development nature of VxWorks means operating code is placed and debugged on the targets in a manner transparent to the host. The characteristics of the operating system are shown in the following table:

Required Operating System Features

- *true hard real time priority working with guaranteed upper time bounds on all important functions*
- *key functions (semaphore operations, for example) are simple and academically sound, so that they can be taught and used without "baggage"*
- *the operating system itself should provide an exemplar of good coding and documentation practices*
- *small size*
- *dynamic loading not just of application programs but of device drivers and interrupt handlers*
- *programming is in ANSI C, or C++ (and the potential to port other languages)*
- *full support for TCP/IP and its major applications (such as RPC, NFS, FTP)*
- *ability to run diskless, while allowing full disk access (as seen by the user) to disks attached or mounted to the hosts*
- *an ANSI compliant library, to reduce the learning time of students (and staff!)*
- *to run under the heterogeneous set of hosts and operating systems at Essex (or have the potential to be ported)*

Many of the tools are extensions to the GNU toolset which were supplied for our Sun environment., We have ported them to NeXTStep and HP Apollos and have written drivers for the HM[1] VME boards. Thus, VxWorks integrates smoothly into university Unix environments which, coupled with its elegant principles, makes it an excellent computer science teaching tool.

2.5 Simulation and Emulation Tools

Even though we are relatively well endowed with hardware, software simulation and hardware emulation tools are widely used in industrial pre-fabrication design work; simulation is also useful for reducing the hardware loading.

We use two types of simulation. The first is VxSim and is a commercial product that simulates a complete VxWorks OS running as a process under UNIX. It is possible to log into a VxSim process, download a program and run it exactly as if it were a real target running on its own processor.

1. HM is a trademark of HM Computing Ltd. of Worcester, UK

The second is DRS (Distributed Robot Simulator) and is the subject of an on-going research project now several years old (information on this and other robot related topics can be found in the archive at Essex[1]). The idea is to provide a “behavioural” simulator able to test higher-level IAV algorithms such as those for route planning and especially algorithms involving multiple, communicating IAVs. Normally, the code transfers from simulation to VxWorks without modification. The user can start any number of IAV programs; the simulation provides a two-dimensional graphical view of the experimental space in which various obstacles can be placed. The simulation is fully distributed and can be run across any number of workstations.

Hardware emulation is provided in the form of cheap bench units containing buttons and a stepper motor to mimic the IAV chassis which, in combination with bench rack, substantially reduces the number of real IAVs needed.

2.6 The Network as a Virtual Laboratory

The ability to replace targets by simulations enables students to conduct work on any machine capable of running them. In addition, the underlying cross-development architecture of the VxWorks means that even when real-targets are being used, their physical relationship to the host development system is only an issue if the students need to interact physically with the target hardware in question. Thus, by utilising a network connection to the target hardware or the simulation servers, a student situated outside the laboratory may have a *virtual presence* in it. This increases the productivity which may be obtained from the facilities by extending access beyond the time limits imposed by usual safety or security considerations associated with hardware laboratories.

2.7 ECAD Tools

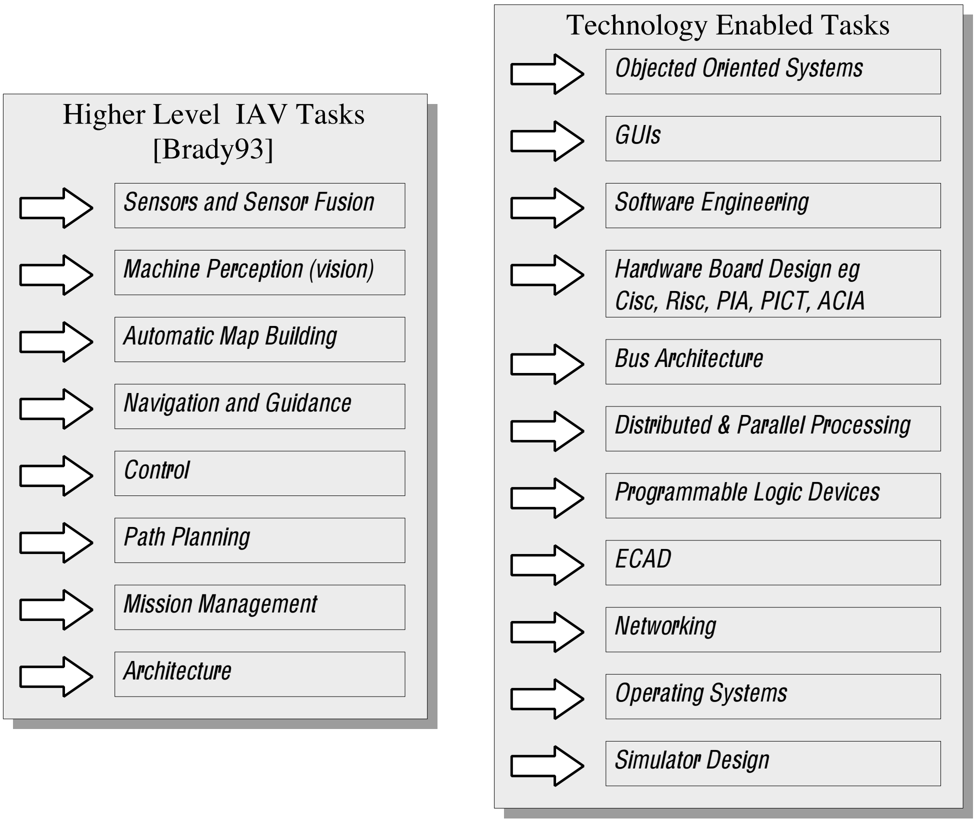
Programmable logic devices and associated electronic computer aided design (ECAD) packages have replaced much of the discrete, component design based methodologies which were commonly used in the construction of previous generations of digital systems. This approach has considerable advantages for education. It reduces the student time needed to implement hardware designs by providing computer assistance with the task of circuit entry, simulation and fabrication, allowing devices to be used in a similar manner to that of conventional processors. It is now quite possible for students to experiment with a number of variants of a hardware circuit within a single assignment period. The ECAD software used in this laboratory was supplied by the Altera Corporation of San Jose, USA (who invented the first EPLD, the EP300, in 1984) and runs on Sun, HP and MS-DOS platforms.

3.0 Supporting Computer Science Curricula

We use the laboratory in two quite separate ways. In the first, the IAV **theme** provides an educational thread from basics to a working, integrated IAV system. In the second, the power of

1. ROBOTS is a small archive dedicated to storage of robotics related information. It can be accessed via Janet or Internet at ftp.essex.ac.uk. The robot archive is located in pub/robots. It can also be accessed via WWW at http://www.-essex.ac.uk.

the facilities is used to augment the teaching of a number of **stand-alone** courses at different levels. The following tables summarise some of the activities offered by the laboratory.



It can be seen from this list, which is by no means exhaustive, that the field covers the gamut of the computer science repertoire. Indeed, it is typical of the multi-disciplinary nature of most functioning computer systems.

3.1 The IAV Theme and Contest

Within the theme, staff set individual assignments to suit their courses by selecting an incomplete set of modules and getting their students to develop the missing one. These could be hardware, software or AI based. Wherever possible, we try to create design oriented assignments, which we feel provide higher levels of freedom and motivation than procedural based alternatives.

As a climax to the year, a competition (for a small prize) is organised which allows the students to combine separate assignments to build a fully working mobile robot able to solve some problem. This approach not only adds to the interest and motivation but offers the educational benefits of providing a mechanism for supporting teamwork and larger projects.

The IAV theme clearly draws on the research domain whilst simultaneously offering students immediately applicable industrial skills. There is thus great potential for integrating IAVs into computer science.

3.2 Assignment Support for Specific Curriculum Areas

The following are not exhaustive descriptions but rather give an indication of the type of issues addressed.

3.2.1 Hardware Design

The methods and equations used to design hardware are strongly dependent on understanding the underlying natural phenomena (unlike with software) and our ability to identify and analyse the numerous variables involved. Current limitations mean hardware models are frequently based on assumptions and simplification which students sometimes mis-interpret as complete truths. Thus, they frequently find themselves bewildered by the failure of simple circuitry which apparently conforms well to their logic theory. Hence, at Essex, we have a deep conviction in the need and value of practical construction as an aid to a deeper understanding of hardware design.

To support hardware design, an in-house VME prototyping kit has been produced which facilitates construction work at a component, chip and systems level. This year, two assignments were offered, one involving the design of a DRAM controller using Altera EPLD chips and the other an A/D interface. In addition, project work such as a vision pre-processor engine and a smart ultra-sound proximity detection system have been hosted by this system.

3.2.2 Basic Systems Programming

Three laboratory assignments have been created to augment this area. The first acts as an introduction to the laboratory and allows the student to write an application program, in this case for the allophonic voice synthesizer. This device was chosen for its simplicity. The assignment also introduces the semaphore notion and process synchronization.

The second assignment allows the student to drive a device (in this case, a simple switch and stepper motor interface) directly from their C program. This introduces them to the organization of digital I/O devices.

The last of this series requires the students to build on knowledge gained to create a full interrupt-driven device driver, properly documented and tested.

3.2.3 Operating Systems

Until recently, it was difficult within our department to provide students with direct access to operating system primitives without interfering with others. Hitherto, such assignments were simulated. The elegance and academic soundness of VxWorks allows that access. Assignments are being built that allow the student to

- substitute their own code for intrinsic OS functions
- experiment with scheduling, queuing and, most importantly, functionality that is time critical

3.2.4 Data Communications

Network experimentation also can interfere with others, and on a grand scale. The Brooker Laboratory both provides access to very low level network functions inaccessible by students elsewhere, and prevents access outside the confines of the Laboratory of such dangerous traffic. For example, one experiment graphs the efficiency of the network under various traffic requirements (length and frequency) against background loading on the contention based ethernet.

3.2.5 Artificial Intelligence

One of the most exciting aspects of the new facilities is the potential for practical experimentation in some of the latest AI techniques hitherto only available to students as theories or in simulation. To this end we have ported two implementations of the Predicate Logic based language Prolog to VxWorks.

Four AI topics in particular are being explored for potential student assignments or projects:

- Reactive Planning in which plans are dynamically adjusted in the light of changes in external state.
- Machine Learning methods such as Genetic Algorithms and Neural Networks where computer systems can “learn” a desired “behaviour” in response to some kind of “training” or positive and negative stimuli from the environment.
- Machine Vision[1]
- Distributed AI in which several autonomous systems may co-operate in some common mission such as providing a map of an area.

3.2.6 Software Engineering

The laboratory provides a superb infrastructure for software engineering. NeXTSTEP comes complete with excellent program construction and interface building CASE tools (sometimes called “lower” CASE tools). They are based on Objective C[2] [Cox91], an augmented C derived from SmallTalk. Of course, any Unix based “upper” CASE tools will run upon it, and if necessary, the cheaper PC based tools can run on the same platforms under MS/DOS and Windows.

In addition, the need for students to work in teams provides them with a requirement to adopt good software engineering practices. VxWorks itself provides a very good example of a well designed product, which demonstrates the virtues of structure, documentation and programming discipline.

---

1. Using cameras supplied by VLSI Vision Ltd
2. Objective C is a trade mark of the Stepstone Corporation.

4.0 Evaluation

The laboratory accepted its first students in September 1993 and at the time of writing (March 94), it has been running for only two terms. Currently, 12 host/target combinations and 4 IAVs are servicing the needs of some 300 undergraduate and postgraduate students, providing them with approximately 6000 hours access. Up to 55% of this access is attributable to the virtual laboratory presence feature. Clearly, the innovative application of these tools offers a significant increase in laboratory utilisation resulting in cost savings to higher educational establishments. At this early stage, the success of the laboratory in meeting the goals set out earlier in this paper has not yet been formally evaluated. Informal observation and feedback has suggested that existing and potential students do indeed find it a most stimulating environment.

5.0 Summary

An innovative coursework environment which integrates hitherto separate computer science strands at the University of Essex into an IAV theme has been described. It utilises state of the art concepts and tools such as OOS, PLDS and RTOS to allow the provision of a highly modularised assignment construction infrastructure with maximum flexibility and minimum complexity. The laboratory has spearheaded both academic and commercial arrangements which are now available to the wider UK higher education community. From conception, this laboratory has taken almost 2 years of intensive planning and construction from a team of 8 academic and technical staff. The sense of stimulation and fun felt by all involved in this project has made our working life a pleasure, a feeling we now hope to share with our students.

6.0 Acknowledgments

The authors are pleased to acknowledge the contributions of the following members of our department whose skillful work and inspiration were needed to convert these ideas into reality (in alphabetical order): Ian Boddington, Martin Colley, Robin Dowling, Erik James, Tony Lawson, Malcolm Lear, Mike Millman, Simon Moore, Jerome Robinson, Adrian Simmonds, John Tierney, Chang Wang and Adrian Wheal[1]. We are should also like to express our gratitude to Paul Scott (Head of Department) and Simon Lavington (Head of the Systems Architecture Group) for their encouragement and vigorous support throughout the duration of this work.

It gives us great pleasure to acknowledge the generous donation of ECAD software from Altera Inc which forms the basis of programmable logic design in this laboratory. Special thanks are due to Camille Brooks of Altera who has patiently managed this process. It is also a delight to acknowledge Wind River Systems and in particular Steve Harris who had the foresight to understand the potential of RTOS in practical computer science education and who pioneered the birth of an educational package that makes it possible for universities to afford the high-technology required to set up this type of laboratory. We are also grateful for the sponsorship of a Wind River Systems prize to the winning IAV competition team.

1. Now head of computer services at Canterbury Christ Church College

References

[ACM93] ACM Special Issue: Technology in Education (including educational systems and design, educational tools, educational policies and issues), Communications of the ACM, Vol.36 No.5 May 1993

[Biermann94] Biermann A W, "Computer Science for the Many", Computer Vol.27 No.2 February 1994, 62-73.

[Brady93] "Overview IAV-93" invited talk at 1st IFAC International Workshop on Intelligent Autonomous Vehicles, Southampton 18-21 April 1993, proceedings published by Pergamon Press.

[Coltrane 94] Coltrane P "Education, Technology and Change", Computing and Control, IEE, Vol.5 No.2 April 1994, 52-54.

[Friedman93] Friedman B, Kahan P H, "Educating Computer Scientists: Linking the Social and Technical", Communications of the ACM, Vol.37 No.1 January 1993

[Henderson91] Henderson J, "Is There a Computer Science", BCS Computer Bulletin, February 1991

[Notkin93] Notkin D, Schlichting R D, "Computer Science in Japanese Universities", Computer Vol.26 No.5 May 1993, 62-70.

[Oakley90] Oakley B "The Limits to Growth in IT", Computing & Control Journal, January 1990

[Papert80] Papert S, "Mindstorms: children, computers, and powerful ideas", Harvester Studies in Cognitive Science; no.14 Published Harvester Press, 1980.

[Soloway93] Soloway E, "Should We Teach Students to Program?", Communications of the ACM, Vol.36 No.10 October 1993