Systems Laboratory Refurbishment Proposal

Systems Laboratory Refurbishment Proposal

1.0 Introduction

This document outlines a proposal to restructure and upgrade the systems laboratories. It is being offered in the light of complaints regarding the increasing unreliability of the systems lab equipment, the general obsolescence of much of the test tools, the increasing gap between lecture & lab coverage and the problem of falling numbers (& quality !) of students opting to follow systems degrees.

This document will propose a set of actions which will:

* ***solve the current reliability problem***
* ***replace obsolete equipment by "state of the art" gear***
* ***provide support for uncovered lecture topics***
* ***increase systems and general student numbers by:***
  - ***providing a highly interesting and novel laboratory environment***
  - ***providing a general showcase for student recruitment***
* ***economize on laboratory costs by restructuring space***
* ***provide a unifying laboratory framework enabling:***
  - ***the department to capitalize on its unique combination of skills (eg enabling combined AI & Systems projects etc)***
  - ***closer interaction between staff hitherto isolated in groups***
* ***accommodate team work at varying levels (as mentioned by BCS)***

2.0 Current Problems

2.1 Unreliability

The principal source of unreliability lies in the ten 680xx development systems. Each system comprises; a bus (based on Sinclair QL signalling), processor card (68008), memory board, floppy disc controller, 2 I/O cards, software monitor and assembler. Most of these items were developed in-house or were modifications of material in the public domain. The original D&D of these components has left them with various bugs (both hardware & software), some of which would require major hardware redesign to fix. We are currently surviving by reducing the total number of assignments and by cannibalizing systems to maintain a smaller working pool. Despite doing this we encountered complaints about reliability (eg at the last staff-student meeting).

In addition, the current assignments require some of the PCBs to be removed, resulting in wear of the connectors which have now reached the end of their life.

It is felt that attempting to repair these systems is not a viable option as, apart from the difficulties associated with rectifying the technical faults, they no longer support newer course material such as bus arbitration.

2.2 Obsolete Equipment

Most of the systems laboratories are currently being supported with either obsolete or inappropriate test equipment. It is well known, for example, that analogue oscilloscopes and multimeters are not the most effective tools for synchronous logic development. Analogue tools produce high levels of redundant time & voltage information and low volumes of relevant state & data-domain information. This is confusing to beginners. The proper tools are such items as logic probes, logic analysers and emulators. Whilst asynchronous logic can often be profitably examined with high-speed oscilloscopes, the frequently non-cyclic nature of digital signals means they are better served with a storage oscilliscope (often part of a modern logic analyser).

Good test equipment is expensive and it is therefore suggested that possible economies from sharing test equipment between systems labs be fully explored. To this end, moving the systems labs to share a common space could be beneficial. For example, the current timetable has both the 1st and 2nd year labs scheduled for Monday & Friday afternoons. If they were timetabled on different days, weeks or terms, it should be possible to share equipment.

2.3 Curricular Changes

Whilst it is acknowledged that laboratory work can never provide total coverage or support for lectures, the current labs omit the following essential topics:

* ***Bus Arbitration***
* ***Man-Machine Interfaces***
* ***Hazards, Reflections & Noise***
* ***Digital T&M + Emulation***

This proposal will later suggest how these topics may be included. To assist those who are unfamiliar with these areas, a brief statement on each follows:

Bus Arbitration - Arbitration is an essential feature of any shared-bus multi-processor architecture. All major buses support this feature (eg VME & Multibus).

Man-Machine Interfacing - This complements the Machine-Machine aspects of the CC205 course. It is more commonly referred to as Human-Computer Interfacing (HCI) and includes; mice, pens, widgets, GUIs (eg X-Windows) & interface building (eg Next Interface-Builder). These techniques are becoming increasingly popular in embedded computer system design work.

Hazards, Reflections & Noise - The speed of computer hardware is continually increasing (eg RISCs). Above 10MHz signal delays and other phenomena need to be included in design theory (eg bus matching). Also, embedded computers frequently require the application of special techniques to operate correctly in electrically hostile environments.

Digital T&M and Emulation - Digital system design and test has generated problems which are unsuitable for traditional analogue tools. Thus, most leading computer R&D departments use specialised digital tools.

2.4 Declining Numbers

In relation to other Computer Science degree courses such as AI, Systems has difficulty in recruiting students. The reasons for this are undoubtedly complex and may involve such factors as the national trend away from science & engineering. However, it is clear from the success of AI that, where it is possible to offer subjects that appeal to the aspirations of the younger generation, it is possible to attract good students in reasonable numbers.

It is also apparent, that whatever our own assessment of the value of our existing courses, the ultimate judge and jury will be the A-Level students themselves. Clearly, with the exception of AI, we are failing this test. Given the current plans to expose universities further to market forces, it would seem that the situation is unlikely to improve of its own accord in the foreseeable future.

With these thoughts in mind it is suggested that the planned upgrade of the systems laboratories be done with regard not only to preserving our academic purity but bearing in mind the wishes and aspirations of our potential customers, the sixth form students.

It has been suggested in private conversations with both staff[1] & students at Essex that part of the success of AI schemes may be due to the fact that they more closely match students' aspirations, which have been inspired by the media and popular science fiction. For instance, many potential students are motivated by the thought of being involved in the development of human-like intelligent robots. Whilst most members of the academia would recognize this ambition as being somewhat removed from reality, it must also be said that imagination is undoubtedly the food of aspiration and inspiration. Further, if market forces are to be allowed to permeate universities, perhaps it would be well also to remember the fate of companies who disregard their customers' requirements. Thus, one avenue this proposal will attempt to explore is whether systems can be packaged in a way which is more compatible with the aspirations of school-leavers, without sacrificing academic respectability. It is suggested that a robotics theme (possibly coupled with some aspect of virtual reality)[2] may have the required characteristics.

3.0 Systems Laboratories - An Overview

3.1 Physical Layout

[1] Ack. J Robinson & A de Roeck

[2] Ack. Paul Scott & Jim Doran

Currently the undergraduate systems teaching laboratories occupy three sites:

* ***5.518; 3rd year projects***
* ***3.515; main systems lab***
    - ***CC102/CC205/CC207 (undergraduate)***
    - ***CC951 (Diploma)***
* ***4.405; Apollo Lab***
    - ***CC102/207/308***

In addition, the following VLSI Design/Graphics work uses the DEC CAD facility provided by computing services:

* ***4.302b; University Graphics Lab (DEC)***
    - ***CC207 (undergraduate)***
    - ***MC301***

3.2 1st Year Systems Lab

The first year laboratory serves a broad audience of 1st year students, many of whom will not be Systems Architecture specialists, nor indeed single honours computer science students. Current assignments consist of three main types of experiments; architecture simulation, logic design and microprocessor programming (using an intergrated software and hardware work bench). The architecture simulation has recently been updated to run on the Apollo systems. This provides a modern colour GUI environment which has proved particularly popular with students. The logic design exercises are conducted using a traditional "bread-board" based on a fine array of connector sockets. These appear to be reaching the end of their useful life as the sockets' connections are becoming unreliable due to wear. Software development (68xxx assembler) is conducted on 15 ageing Sage computers. In addition to their dated appearance, they are also victims of advances in technology and changes to the curriculum. For example, the Sage facilities offer an outdated and unfriendly programming interface in the form of a dumb terminal and an assembler which is hosted by the p-system/Pascal (it's 5 years since Pascal was taught in the department)[3]. Also, the primitive nature of the Sage bus does not make it suitable for 2nd year use (ie there is no multi-processor arbitration or VXI support etc). Perhaps, the sharing of new 68xxx racks (& workstations) required by the 2nd year may be a possible solution. This would have the merit of providing a more consistent development environment for the students.

---

[3] Ack. Iain MacCallum & Dave Lyons

3.3 2nd year Systems Lab

As detailed in the introduction to this proposal, the unreliability and obsolescence of equipment in the 2nd year ug systems laboratory was the principal reason for the current review exercise. The 2nd year systems lab is very much at the heart of the systems teaching philosophy at Essex as students are exposed to it at a pivotal point in their ug life. Thus, the precise structure and content of the systems lab is of vital concern to the systems group as a whole.

Currently, there are 8 experiments which can be divided into 3 types; 68xxx assembler systems programming, EPLD based hardware design and CAD for VLSI design. Programming assignments currently dominate the experiments amounting to approximately 50% of the workload. In addition to the unreliability problems, the experiments themselves have not kept up with changes to the lecture curriculum. Given the importance of this lab to the systems teaching at Essex, it is important that these problems be rectified as soon as possible.

3.4 3rd Year Systems Lab

The third year laboratory should support work which forms the culmination of a systems student's academic achievements. As such, it should have facilities to support the most advanced undergraduate computing concepts. Currently, the existing laboratory provides three IBM PCs, two old and rather unfriendly logic analysers and a digital oscilloscope. Clearly, given the diverse and complex nature of 3rd year systems projects, this provision is inadequate. In particular, IBMs are not suitable 3rd year architectures as:

* ***the assembler language greatly differs from that taught in the courses***
* ***it has a discontinuous address space***
* ***it will not run software used elsewhere in the department***
* ***it is full of idiosyncratic and poorly documented hardware & software features***
* ***the bus is too primitive (eg no arbitration)***
* ***there is no embedded system development support***

As a consequence, too much of the students' precious time is absorbed by fighting the system rather than dealing with the underlying computer science or engineering issues. Use of a 68xxx based architecture such as a VME system would overcome the first five problems. It is widely acknowledged in industry that hardware emulators are essential for efficient development of embedded systems. Emulators are available either as stand alone systems or as add ons for IBMs. Therefore it is suggested that an effective solution would be to use VME based racks for OEM card development and to convert the existing IBMs to become 68xxx integrated software development and emulation tools (by buying add-on cards/boxes & software). For embedded system development, greater use should be made of microcontoller ICs (ie microprocessors with on-board RAM, EPROM and I/O). It is recommended that all such additions be based around the 68xxx family. In addition, there is a need for facilities allowing the use of modern programmable devices such as EPLDs. Undertaking projects at the leading edge of computing call for excellent equipment and neither the quality or quantity of equipment currently available is adequate. This frequently impedes the progress of students. There are new generations of logic analysers and oscilloscopes which can greatly improve the productivity of students. As good equipment is expensive, moving the project lab into the existing 2nd year area could create the potential to share such items thereby making the overall upgrading and running of the labs more cost effective. An additional problem which needs to be examined is how technical manuals or data books should be made available to students. Currently there is no formal system for managing the supply of technical books. Students complain that some essential manuals have been lost and that there is not an adequate selection of data books. It might be possible to set up a small technical library managed by the lab staff.

3.5 Conversion Lab

The architecture component of CC922 requires use of first year level computer simulation and 2nd year level EPLD development facilities to support course work. In previous years *no provision* has been made to provide hardware facilities for these students. It is hoped to rectify this unsatisfactory position by the inclusion of CC922 in the current lab planning. As the requirements for this course are a subset of the general ug facilities it would seem that CC922 can be accommodated by making due allowance in the quantity of equipment being planned and the scheduling of the lab.

3.6 Diploma Lab

The practical work associated with CC951 is housed and supported by the main systems lab. The work is largely project based (eg the design & construction of embedded microprocessor PCBs). Its requirements are relatively modest consisting only of soldering irons, power supplies, multimeters and high-speed oscilloscopes. The only current deficiency is the lack of provision of high-speed oscilloscopes. The procurement of high-speed oscilloscopes for the main laboratory will address this problem.

4.0 The Proposed 2nd Year Systems Laboratory

One of the goals within this upgrade proposal is to package systems coursework within a framework which will appeal to school-leavers as well as providing a truly stimulating environment for our students. After much consideration and debate it was concluded that a robotics theme is the best approach. This has many virtues such as;

* ***robots involve a variety of techniques and can be used to demonstrate all important aspects of systems work.***
* ***they appeal to the imagination of the younger generation, thus potentially helping recruitment (they also provide good photographs for brochures, which much of our work doesn't).***
* ***the techniques involved extend beyond systems work and could be utilized by other courses, and even act as a unifying framework for the major departmental specializations.***
* ***assignments can be organized in such a way as to consist of a package of stand alone experiments***

which can combine to produce a working robot (providing more motivation than simply viewing an abstract result on some specialist equipment).

* ***they lend themselves to team-work.***
* ***whilst their physical form may change, robots are unlikely to go out of fashion in the next 10 years.***

4.1 Robotic Theme

Essentially, this proposal amounts to the use of a small number (approx. 4) of small free roaming robots (or arms) to form a focal point for a variety of student coursework assignments.

A top-down level view of such a lab would see a set of robots, a robot maze, a set of sensors (camera, ultra-sonic, pressure-pad etc., appropriately positioned about the robot work area), infra-red/radio robot communication links, bench mounted bus racking hardware systems, EPLD & EPROM design and programming tools, logic analysers, hardware emulation systems, simple PCB design and construction facilities, multiprocessor simulation (eg existing transputer development system) and a set of powerful object-oriented (& assembler) development workstations.

The robots would essentially only consist of a simple mechanical chassis with a minimal amount of on board electronics (battery, motor control and infra-red/radio serial communication interface). All the clever computation and position sensing would be done externally via hardware, software & sensors distributed and controlled elsewhere in the lab. This philosophy is proposed so as to enable the development and application environments to be split. This has the following advantages:

* ***a virtual robot (software robot emulation) can be utilized as an interim target, reducing the number of real robots required (ie there is no need for a 1:1 student to robot ratio).***
* ***simpler robots mean they would be smaller, lighter and therefore cheaper. Having fewer components, they will be easier to maintain and their reduced weight will consume less power, require smaller motors and be generally more reliable.***
* ***new robots or other applications can be added with a minimum effect on the system (virtually no hardware is dedicated to the physical robot)***
* ***hardware and software development can be conducted using a traditional bench development environment.***

4.2 Nature of Assignments

The suggestion is that a number of courses at different levels should be able to utilize this lab. A complete set of working hardware and software modules required by a fully operational robot would be kept, allowing staff to 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.

It is proposed that students following existing 2nd year systems courses (CC205 & 207) would be divided into teams. Over the course of a year they would develop a set of modules to create a working robot. To add a little interest and provide additional motivation, each team would enter a competition to see which robot design best completed a set of test tasks at the end of the year . A full set of assignments is proposed below.

4.3 The New Lab Structure

Clearly, the 2nd year systems experiments need to be designed to follow on to the 1st year in as smooth a manner as possible. Also they need to support all systems courses to 2nd year level and set the ground work for 3rd year courses and projects (the 3rd year architecture course has no coursework associated with it).

There are clearly many options in the design of such a lab but after careful thought, it is proposed that a dual-track approach be adopted; an *experiment fair* within a *unifying theme*. Thus, it is proposed that the laboratory consist of a set of stand-alone experiments (similar to the existing approach) but that they be designed in such a way as to make it possible to combine them in order to form a more complex system. This approach has many advantages such as:

* ***students can build more complex and interesting systems***
* ***both individual & team-work can be accommodated***
* ***team-work & large system development more realistically reflect post-university life***

4.4 Systems Topics to be Covered

In-house discussions, external visits and planning sessions have led to the compilation of the following list of topics which the 2nd year lab should support in some way:

* ***EPLDs***
* ***Computer bus; data transfer, arbitration, matching***
* ***Processor design; RISCs, Neural Networks, Parallel Processing, Transputers***
* ***Peripherals; memory systems, discs, keyboards & VDUs, pens & mice***
* ***Interfacing;***
  machine-machine - A/D, ACIA, PIA, SCSI, Ethernet, polling, interrupts, DMA
  man-machine - GUI, X-Windows, NextStep
* ***Test & Embedded Systems; Probes, Logic Anaylsers, emulators, instrumentation***

4.5 The Proposed Experiments

A general view of the envisioned lab is given in the following diagram:

The reasons for adopting this physical structure have already been explained. It is being proposed that students be divided into teams at the beginning of the academic year. Each team would produce a set of modules required to produce a fully working robot. Each team would then be divided into student pairs and each pair would complete 4 assignments during the year. To ensure proper coverage of essential principles, an assignment selection framework might be needed. For instance, the experiments could be grouped into sets and an associated rule could be used to constrain the students' selection (eg choose one from each group). Each assignment would be a stand-alone piece of work. At the end of the year it should be possible to combine individual assignments with the work of other team members to produce a working robot. It is hoped to create *design oriented* assignments, with the goal of providing higher levels of freedom and motivation than in procedure based alternatives. To enhance motivation further, it is suggested that the culmination to the year's work might be a competition to see which team's robot is better at completing certain tasks.

Whilst it is necessary for the technical framework to provide a well defined software and hardware interface, the exact nature of any experiment could be left to the lecturer concerned. Thus, the following experiments are offered solely as an illustration of the type of assignment being envisaged (more suggestions welcomed !):

Bus Management *(CC207)* - This is a hardware & software design exercise which aims to reinforce the extensive hardware bus notes given in CC205. This would involve the design of a parallel arbitration system for a VME using an EPLD.

VXI Instrumentation Bus (CC205) - This is a programming exercise to illustrate how computers can be used to manage instrumentation networks. It concerns programming a VXI control card (the VME based successor to the GPIB-IEEE488) in order to control the camera .

Parallel Machine-Machine Interfacing (CC205) - This is a PIA programming exercise. It will involve programming a pre-fabricated board to collect robot sensor data. It will demonstrate the techniques of memory-mapping, register peripheral control and interrupt programming.

Serial Machine-Machine Interfacing (CC205) - This is an ACIA programming exercise. It will involve programming a pre-fabricated board to collect robot sensor data using a serial infra-red communications link. It will demonstrate the techniques of memory-mapping, register peripheral control and interrupt programming.

SCSI & DISCS (CC205) - This experiment will be designed to expose the underlying principles of the SCSI interface together with the fundamental principles associated with floppy and hard discs.

Transducer Interfacing (CC207) - This assignment will show how computer systems can be made to interface to analogue signals. It will consist of the completion of an A/D board design and the production of a software driver to monitor the robot battery voltage. It can use this information to determine battery condition (or an alternative warning of presence of obstacles!).

Interface Building (CC205) - This experiment is designed to illustrate how window based GUI systems (such as X & NextStep) can be used to communicate with, and manage, embedded computer systems. It will do this by creating a virtual control panel for the robot.

Video Frame Grabber (CC207) - This is an EPLD experiment. It would involve the design of a video timing circuit using an Altera EPS464 Synchronous Timing Generator. It would demonstrate logic design using a timing waveform as input data. The Video Frame grabber board would be partially built with only the need for the STG to be designed and inserted.

Risc Design (CC207) - This is a processor design experiment. It will involve the completion of a partially built RISC based image analysis coprocessor engine. The work involves design and construction of a RISC/VME signal interface.

Neural Network (CC205) - This is a processor design experiment. It will involve programming a_ RISC based image analysis coprocessor engine used to manage an obstacle avoidance system based on simple pattern matching.

CAD - PCB Design (CC207) - This assignment is targeted at imparting some of the extensive "engineering aspects of computer design" topics covered in CC205 such as hazard removal, reflection elimination, cross-talk & noise eradication. The design of a simple memory expansion board will be used as an example.

Object Oriented Control (CC205) - This laboratory will introduce the notion of object oriented systems by showing how the simple robot management systems operates. The student will then be invited to add some additional functionality to the system via creating a few new objects and invoking delegation.

Hardware Objects (CC207) - This laboratory will emphasize the role of modularization as a fundamental property of hardware and software system design. It will show how object-oriented systems support this notion and how hardware modules can be made to become objects within a software/hardware OO framework.

Parallel Computing (CC205) - This assignment will demonstrate how parallel processors, in the form of a small *transputer farm,* can be used to increase the performance of a computer system. In addition to introducing the student to principles of parallel computation, it will illustrate the need for computational metrics and benchmarks.

4.6 Technology Framework

It was considered desirable to continue the current 1st & 2nd year tradition of illustrating systems within an overall integrated hardware & software environment. Further, to provide consistency with other areas of computer science, 68xxx based hardware and software was chosen. Modularization is widely acknowledged as being a most effective design strategy and has been embodied in the form of a shared bus development rack and object oriented systems. The particular technologies chosen are VME, which historically is strongly associated with Motorola and Unix systems, and Objective-C, which can be regarded as a super-set of C. Thus, this choice is consistent with material found in other parts of the degree scheme.

Undoubtedly, there are many ways to manage the kind of modularized robotics system proposed in this document. Rather than recommending some rather pedestrian approach, we are suggesting the use of a forward looking and intellectually stimulating approach based on object oriented systems (OOS). OOS embraces the modularization concept in the form of modules known as objects which offer well defined interfaces. The originator of Objective-C refers to these objects as software-ICs and frequently draws close parallels to hardware development methodology. Next computers utilize OOPs as their underlying development environment, in particular Objective-C. The forthcoming release of this environment includes support for global network based objects which could be used to communicate with hardware objects. The OOPs paradigm is undoubtedly an intellectually stimulating and highly topical environment, offering the potential of managing a distributed net of mixed hardware and software objects or modules. In addition, object oriented techniques are now proving to be the most popular method for developing modern GUI based HCIs (also featured on CC205). British Telecom, in line with many other manufacturers, have recognized the ability of such systems to interface to exchanges and other embedded computer systems.

Thus, the new lab could be viewed as consisting of a palette of hardware and software modules which can be assembled to produce experimental systems. By using a network, these modules (interchangeable between software and hardware wherever possible) could physically reside anywhere within the lab network (ie they might be part of an embedded system such as a robot arm, or they could be in a separate hardware development rack, or they could be software residing in a supporting workstation). The OOS notion of objects would be an ideal mechanism to support this.

By utilizing the network to effect connectivity on a lab wide scale, robots and other real instruments (together with all their components) could be made members of a networked laboratory object community. Thus, objects would range from such things as GUI widgets through to real hardware circuits such as stepper motor control circuits. Full use of virtual control panels and instruments could then be made to create a stimulating and exciting experimental environment for students.

This framework would be very flexible and not lock us into any particular application (eg robots) and, if required, would allow a mix of applications with varying degrees of coupling to the OOPs framework. An OOPs dialect of C (plus 680XX assembler) would be preferred form the Systems development viewpoint which could also be made to map nicely on to the C presented in CC101.

4.7 Other Uses of Systems Laboratories

It has been suggested that the following AI topics may be suitable for assignments in a new systems lab:

* ***Vision (Robot Scene analysis)***
* ***Planning (Robot Route & limb movement)***
* ***Natural Language (eg contextual character recognition, speech processing)***
* ***Neural Networks***
* ***Virtual Reality (maybe a more sophisticated version of the virtual robot simulation !)***

However, although the systems lab might provide the bare tools to support these, specific assignments would need to be created and produced by the AI staff to serve their courses (should they so wish !).

In addition, members of the AI group may wish to consider the nature of the robots and the problem in more detail. For instance, rather than considering only simple maze navigation, mobile robots could be used in a dynamic mapping role[4]. Here, the concept of the maze is extended to cover the entire floor space of the lab. The roaming robots could then work as a team to map both the static (eg walls, furniture) and dynamic (eg people, trolleys and even the robots themselves) inhabitants of the lab. There is also a case for having (or replacing) the mobile robot with a simple, static arm/grabber version, which would facilitate use of the "Blocks World" concept (already used in the 1st year Maths course)[5]. Also, the amount and type of feedback is an important issue[6].

4.8 Space Reorganization

Recent changes to the systems laboratory course structure (eg moving the 1st year register-machine simulation) have combined to make it possible to move the 3rd year systems project laboratory (5.518) into the systems laboratory (3.515). In addition, it is proposed to accommodate the new systems course work associated with the conversion course CC922. This reorganization will allow better use of the department's resources, in terms of both manpower and equipment. This should have the following main advantages:

- ***produce economies from equipment sharing***
- ***produce economies in supervision***
    - ***allow one technician to provide safety cover for many hardware courses***
    - ***extend the late cover from the project to all labs (with no extra cost)***
    - ***enable demonstrators to be more effectively utilized***
- ***increase motivation to specialize in systems (1st/2nd years seeing 3rd year projects)***
- ***better space utilization (defending us against predators searching for under-used space)***
- ***provide room for the new software lab.***

The present distribution of 1st, 2nd, & 3rd year (+ diploma) lab space owes more to history than the current requirements of the related courses. There is a case for reorganizing these labs in functional terms (already happening in the Apollo lab) so as to obtain better utilization of space and equipment (eg savings from sharing expensive equipment) and improved "out of hours" access. Currently only the 3rd years have limited access outside 9.00-5.30. In an integrated lab, this would automatically be extended to all labs without incurring additional cost.

---

[4] Ack. Mick White

[5] Ack. Iain MacCallum & Jeff Reynolds

[6] Ack. Edward Tsang

Before such a move could take place, obsolete equipment (and junk) scattered about the lab needs to be removed. In addition, there appears to be a problem with the lack of heating at the end of the main laboratory. There is only one small radiator situated in this area. Two roof mounted heaters appear to be unserviceable. Heating would need to be restored to this area prior to its use in the winter.

Also, consideration might profitably be given to rejuvenating the PCB & dark room so that it could be utilized for 3rd year & M.Sc systems projects instead of being allowed to rot away !

The existing office (cabin) is large enough to be shared by Robin & Malcolm (the existing 1st/2nd/3rd year technical staff). Perhaps it might also be large enough to house a small technical library (mentioned in 3.4).

Finally all the systems labs could do with some redecoration.

5.0 Laboratory Development Plan

5.1 Equipment Required

Based on the laboratory described in these notes it is estimated the following replacement equipment would need to be purchased**:**

* ***VME/VXI or STE rack x15 (plus associated PCBs)***
* ***Altera MaxPlus2 EPLD design software x8 (plus new universal programmer)***
* ***OS/9 (or equiv) real-time operating system (with 68xxx assembler & C) x15***
* ***high speed logic analyser/digital scope x8***
* ***mobile robot chassis , arms or other components x4***
* ***IBM 486 (16MB RAM, 240 MB HD) (see note[7])***
* ***NextStation Color (see note[6])***
* ***PCB CAD & tx-line simulator x4***

5.2 Work Schedule

The existing systems lab represents many man years of work. Most of this work has been contributed in an evolutionary manner (ie occasionally introducing or replacing experiments over a period of years). It is hoped that a similar evolutionary plan can be employed for the proposed

---

[7] Altera MaxPlus2 and Objective-C are our preferred laboratory development tools. Only Next and IBM platforms will support both of these. NextStep (for IBM & Next) offers the only fully integrated OOPs C dialect development environment commercially available. The current 2nd year systems laboratory supports 14 stations and it is anticipated that the new lab will require no less. The optimum mix of Nexts and 486s will need to be decided later when PCSoft & NextStep PC are evaluated.

refurbishment programme. The lab itself is modular and all the experiments are stand alone. Thus, from a development viewpoint, it should be possible to introduce them as time and money allows. Even the OOPs framework can be regarded as a separate development exercise. Clearly, if the modules are to lend themselves to working in an integrated mode, then, as a matter of priority, the interfaces need to be defined. Work has already started on this, and an M.Sc student[8] under the supervision of Paul Chernett and Vic Callaghan is investigating the optimum solution. A tentative plan for upgrading the laboratory is as follows:

Development Phase

Task 1 (asap-July 93) Lab Interface Definition

This work has commenced and the initial results are expected at the end of September. However, some further work by a CO will probably be necessary from Oct 92 to Sept 93.

Task 2 (Oct 92 - Sept 93) Experiment Development

- procure a minimum development set of equipment[9]
- allocate responsibility for developing assignments to academic staff
- allocate responsibility for developing support framework to technical staff
- develop support framework (software & hardware)
- develop a minimum set of rack based 2nd year experiments (2 say)
- develop a 2nd year HCI experiment

Implementation Phase

Task 3 (July 93 - August 93) Relocate Labs

- remove/restore old equipment & junk
- redecorate labs
- move workbenches & rewire

Task 4 (August 93 - Sept 93) Procure & Install Minimum Set of Experiments

- procure minimumset of laboratory equipment[10]
- introduce developed experiments

---

[8] Paul Marriott

[9] Development (x1 unless stated otherwise)

- a VME (or STE) rack
- OS/9 (with assembler & C)
- processor, VXI, memory, ethernet, scsi, floppy disc, ACIA, PIA, prototype (x2) boards
- robot, control interface, comms link
- Altera MaxPlusII, Next SoftPC
- IBM 486 (Colour VGA, 16MB Ram, 240MB HD)
- NextTurbo (Color, 16MB Ram, 240MB HB)

[10] The minimum equipment set will be some multiple of the development pack plus missing items from 5.1. The current 1st year systems laboratory supports 15 stations and it is anticipated that the new combined lab will require no less. It is anticipated 4 robots will be required.

Task 5 (Oct 93 & onwards - 5 years, say) Evolutionary Introduction of New Experiments

- design and build further first[11] and second year assignments as time & money allows
- rejuvenate PCB/darkroom.

5.3 Manpower

As mentioned elsewhere, it is hoped to adopt an evolutionary refurbishment programme, thereby distributing any associated work over a number of years. In addition, it is postulated that the adoption of a design-oriented assignment philosophy will further simplify the task of preparing new experiments (eg the rather tedious "step by step" procedures should not be required). Thus, *if* the implementation of the supporting technical framework can largely be left to the technical staff (with academic guidance), and *if* the preparation of assignments is distributed amongst the systems staff, it should be possible to complete the refurbishment programme without unduly unbalancing the focus of resources and effort within the department. Clearly, due consideration of the optimum balance of human and material resources, in relation to development time, is a critical matter.

Currently, Ian Boddington is organizing the gathering of P&D quotations so that the proposal can be costed.

Should the proposal be implemented in some form, then *high-grade* assistance from the technical staff will be required to help with the construction of in-house hardware and restructuring the labs (eg Malcolm & Robin). This work could commence in Oct 92 and would continue until the completion of the lab.

In addition, Paul Chernett (or another Computer Officer with *excellent* programming ability) would be required to help with the development of the object oriented lab management software (eg virtual robot toolkit) and to act as a systems manager for the workstations. His load would peak during the period Oct 92 to Sept 93 although, as mentioned above, some support would be required throughout the life of the lab (to manage the workstations/computers and associated software).

Also, the task of developing assignments would need to be divided between the relevant members of the academic staff (with technical support).

6.0 Summary

The 2nd year labs have become so unreliable as to require comprehensive refurbishment. It is suggested that considerable advantages can be accrued for the department, with little additional cost, by adding a double theme; robotic applications and object-oriented technology.

[11] As the 1st year upgrade is not a priority, the porting of the current 1st year assignments to the new system has been left to this late stage of the programme. It could be included in the development phase but suitable manpower would need to be provided.

It is also proposed that the work be accomplished in two stages; namely a development and implementation phase. The development phase would commence asap and last for approximately 12 months. It would involve the purchase of a minimum set of equipment on which new experiments would be developed and their feasibility assessed. Upon successful completion of this phase, a full set of equipment would be purchased and the full plan be implemented.

Appendix A - Requirement Summary Table

Appendix B - Overview of Manchester Systems Labs

Manchester has the biggest and perhaps best respected computer science department in the UK. Therefore it was decided that a visit to their teaching labs would be useful in our lab-design deliberations. Simon Lavington arranged with Prof. Steve Furber and Dr Doug Edwards for Martin, Jerome and myself to visit their labs.

It was immediately apparent that the name "*systems*" is used differently at Essex and Manchester. Manchester provide a *Computer Systems* course which has little hardware content. Our *Systems Architecture* scheme equates to what they refer to as *Computer Engineering*. Given that we were slightly confused by the terminology, potential admissions students could be forgiven for having similar difficulties !

Our initial impression, on arriving in their engineering labs, was that they had an active but relaxed feel. A striking difference to Essex was that they did not physically separate workstations from the hardware development benches ! Hardware development was often being conducted next to workstations running simulations or supporting program development. We also noted their strong preference for Suns. Surprisingly, there were also large numbers of Ataris being used for Computer Systems work (eg interrupt programming). The dominance of Sun appeared to be associated with economies of purchasing and maintenance rather than academic or technical arguments. It quickly became apparent to us that they have a much stronger electronics bias to much of their hardware design. Unlike at Essex, there were no experiments supporting the integration of software and hardware, a feature which is a central to the philosophy of Essex systems labs. In many respects we found them to be in a similar position to us, as they were actively considering updating their lab material (they still use Pascal as their main high-level programming language). For instance, Steve Furber has introduced an interesting 1st year assignment involving the use of ViewLogic to design Plessey FPGAs for a muliti-purpose embedded processor board. However, although this is certainly a clever and interesting assignment, it has raised what are currently unresolved issues regarding the resultant changes that will be needed in the following years. As in Essex (and industry in general) there was a noticeable shift towards the use of programmable devices and VLSI design. This was evident in their extensive use of computers together with powerful design and simulation packages. In the first year they used ViewLogic (originally an IBM PC based package but poorly ported to Suns) and for their second and third years they used Cadence & Verilog (with fabrication at 3rd year level). Our existing facilities are broadly similar except we do not introduce programmable devices into our first year. We concluded that this was not appropriate to the lecture structure at Essex.

Our overall impression was that we preferred our focus on the use of software, logic, programmable devices and VLSI in an ***integrated*** systems environment to their more isolated approach. The sheer size of Manchester means they are better able to accommodate specialization by being able to offer more courses. However, we felt an integrated approach to be more appropriate for smaller systems groups such as Essex.

Appendix C - Summary of Staff Opinion

The idea for a robotics based theme systems lab was put to staff (via e-mail) during the Christmas term 1991. They were invited to offer their opinions on the worth of the idea and on whether they would be able to use such a lab to support any of their coursework. A summary of the findings is as follows:

* ***General Staff Comments; about 75% of the staff replied. All were supportive of the aims (some very !). Nobody opposed the idea.***
* ***Systems Staff; analysis of their requirements & comments has led to the conclusion that a lab application theme based on robots has the potential to convey all the essential principles required in the teaching of Systems Architecture. Nobody raised any problems that they felt could not be catered for by a robot theme.***
* ***AI Staff; there was wide general support. Two members stated they would be able to utilize a robotics based systems lab effectively to support their courses.***
* ***Theory & Numerical Staff; nobody said they would have a use for such a lab although, the opinions which were expressed supported the idea.***