

INTRODUCTION AND BACKGROUND

Laboratory exercises form a significant part of the assessment of many units in the Faculty of Engineering. Typically, they are allocated three hours per session, and sessions are supervised by postgraduate students. Reading materials or written instructions are distributed either to the unit's entire enrolment prior to the laboratory, or to the participating students upon arrival at a laboratory session. Preparation (by students) before the laboratory session is often minimal or not required at all. Students are allocated to a particular session of a laboratory exercise, and the number of students at each session is kept as low as possible with regard to timetabling and cost constraints. The students arrive at their scheduled session, complete the laboratory exercise (often in much less than three hours), and either submit a report upon completion of the session, or by some later deadline. Laboratory exercises may or may not form the basis of questions in subsequent examinations for that unit.

The cost of a laboratory *session* includes materials consumed in the exercise, hire of the facility (or the opportunity cost of making the facility unavailable to others, if there is no hire charge), hire of the supervisor, and the provision of written material for the students. The cost of the laboratory *exercise* (irrespective of how many sessions are run) includes human resources to design the exercise, purchase and maintenance of the laboratory equipment, use of the space in a laboratory room (or the opportunity cost of this space if rent is not charged), and the human resources to purchase and maintain the equipment and maintain the laboratory room.

The benefit of laboratory exercises to students is hard to dispute, particularly in fields such as engineering and surgery which have a strong practical emphasis. Laboratory exercises demonstrate the phenomena which are discussed in lectures (eg that a rotating wheel really does vibrate noisily when unbalanced) in an immersive three-dimensional real-time environment (i.e, reality) often involving visual, auditory, and tactile data. Students participate in the laboratory exercise and receive immediate feedback on how different changes affect the system. Laboratories also provide practice in activities related to the profession (whether that is operating machinery or practicing surgery).

Laboratory exercises present significant difficulties in the arrangement of timetables. This is caused by the small number of students in each laboratory session (requiring many sessions), and the long duration (usually three hours) of each session. When combined with the costs of running laboratory sessions, this results in few laboratory exercises for each unit.

Laboratory exercises may also be undertaken at times far removed from the lecture materials they are supposed to reinforce, due to the timetabling difficulties. For example, a third-year engineering unit has its lectures delivered in the first semester of each year, and its laboratory exercises undertaken in the second semester. Also, laboratory exercises often do not require the full time allotted in the timetable.

Access to laboratory exercises for students with disabilities is widely recognised as a poorly handled problem, and is not restricted to problems of physical access (wheelchairs fitting in laboratories) and safe participation (people in wheelchairs

having chemicals drop on their lap). Psychological, neurological, behavioural and emotional disorders may also prevent full participation in laboratory exercises.

Laboratory exercises are inflexible in that students are allocated times in which to do the exercise, and may find it quite difficult to 'catch up' on that exercise if unable to attend as scheduled. It requires a subsequent session of that exercise to have the capacity to accept extra students, when there may be neither subsequent sessions nor slack capacity.

Laboratory exercises conducted in the traditional manner have only one certain benefit over the same exercise conducted in a virtual environment – the immersiveness of the sensory data. Three-dimensional real-time visual technology indistinguishable from actual vision seems likely to be developed in the foreseeable future, and convincing reproduction of sounds is further advanced even now, but olfactory and thermal/tactile (including haptic) devices are far less developed. With such constraints, it will be some time before a software-based virtual reality environment can be considered a viable alternative for laboratory exercises. An entirely virtual laboratory (such as the 'Position Control' laboratory of Control and Mechatronics 322) has the advantage of having minimal¹ equipment requirements – a computer with an internet connection. This means that all the expenses related to physical equipment are avoided, and that an almost unlimited number of students can participate in the laboratory exercise simultaneously (merely by accessing the appropriate URL).

Between these two extremes of an entirely real experience and entirely virtual experience lies an alternative which may reduce the resources required for laboratory exercises in comparison to the existing system. Exercises which do not rely heavily on tactile/thermal or olfactory experiences, or on three-dimensional visual inspection (that is, when a television interface would not be markedly inferior to the experience of being able to walk around an object, and change perspectives quickly and often) may lose little benefit if the student performs the laboratory exercise using a computer interface to control the equipment, collect data, and display an audio-visual representation of the exercise. This scenario is distinguished from a virtual environment in that the equipment is real, and from a real environment in that the sensory experience is divorced from the physical location of the participating student.

The obvious disadvantage such a system has over the current approach is that the sensory experience is a non-immersive, imperfect reproduction (perhaps even only a graphics-based *representation*) of reality as experienced by students in contemporary laboratory exercises. This disadvantage cannot be avoided: If the lesser sensory experience compromises the laboratory exercise too severely, then there is no acceptable alternative to the current approach for that laboratory.

Using computers as human-laboratory interfaces does, however, avoid or reduce the impact of some of the disadvantages of the current system. If the actual presence of students is no longer required at laboratory sessions, then the sessions may well have reduced timetabling needs. For example, a single-semester unit with fifty students may require ten sessions for one laboratory exercise, and therefore be allocated three hours each week in the semester timetable (39 hours in total), with

¹ Except for the consideration of (typically expensive, where they yet even exist) virtual reality I/O devices.

repercussions for all the other units in that unit's major (and more units in other majors, as well). If instead each student completed the laboratory alone, the laboratory exercise's equipment would need to be available for up to 150 hours. Although students in the same year and major have similarities in their timetables, it seems probable that enough timetable differences, evening hours, week-ends, and non-teaching periods exist so that these 150 hours would require fewer hours on the timetable.

Of course, in the example above, each group of five students could have a different laboratory exercise every week, to as many as ten exercises in one timetable block. Running ten different exercises simultaneously obviously would require as many supervisors and rooms, however. Using a computer interface removes the need for supervisors but not the need for rooms, although less space may be needed when students are not physically present. Providing three hours of access for each student to every laboratory exercise, via computer human-laboratory interfaces, is not more difficult than providing three hours per student for one exercise – provided that whenever one exercise is being accessed by a student, all other exercises are being accessed by other students. To the extent that these resources are not consumed in parallel, additional requests for concurrent access to an exercise would occur. If unscheduled access is denied too often, each exercise may need to have its own timetable, filled in by students according to each individual's timetable on a first-in, widest-choice basis, with failure to access the exercise at the arranged time resulting in that student being forced to arrange a new time when the exercise is not yet booked. This would be more conveniently accomplished if a single booking system dealt with all laboratory exercises (preventing each student from holding concurrent bookings in different exercises). Computer-based remote human-laboratory interfaces introduce new scheduling difficulties but may result in simpler and more flexible timetables.

Another disadvantage under the current system is that a full three hours is timetabled regardless of need. If a booking system such as the one described above proves unnecessary (due to few concurrent access requests), then the laboratory equipment is allocated only for the time needed to complete the exercise. If a booking system *is* needed, forfeited bookings or early completion still results in equipment availability, but using this free time depends on the student who holds the next booking being ready to start early, or another student 'hovering' on the bookings page and willing to attempt to finish the exercise before the next booking begins.

Students with disabilities may face obstacles in using computers, but these obstacles may be less damning than physical participation in a laboratory. Computers are tools capable of providing ways to reduce the impact of disabilities on participation and the quality of the student's learning experience.

Traditional laboratories also require attendance on campus, leading to resource requirements such as parking bays and transportation costs. Computer-based remote laboratory participation may avoid this cost but introduces costs of internet connectivity, and requires additional equipment for the laboratory exercise (automation equipment, including software licences), with all the attendant purchase, installation, development and maintenance costs.

A final benefit to consider is that as on-line resources are developed for units, greater synergy between the unit material and the laboratory exercise can be sought.

DEVELOPMENT OF DESIGN CRITERIA FOR REMOTE-ACCESS, AUTOMATED LABORATORIES

The primary criterion for transferring a laboratory exercise from a traditional physical arrangement to a HMI²-accessed automated arrangement is:

- (1) The richness of the sensory data which the student must experience in order for the exercise to impart satisfactory benefit, must not exceed the capabilities of the interface used** (eg a personal computer with a 2 Kbps modem connection, CD-ROM drive, large hard disk, and a 4MB 3D video accelerator³).

If an exercise requires thorough visual inspection, or olfactory, thermal, or tactile senses, then it cannot be automated. If frequent, visible changes occur, then modem connections may not cope with the amount of data to be downloaded, and delays of several seconds between each update of an image (rapidly increasing as image size is increased) will limit the quality of the sensory experience, if the exercise is to include images.

Server-side Descriptions of Client-side Models

A promising alternative to transmitting actual images to the user, is to use a data stream (perhaps MPEG-4) to describe positions and motions of 3D graphics objects stored on a CD-ROM. Sensors could track movements of each significant part of an experiment's equipment (in contrast to providing actual visual feeds from cameras). Downloaded data does not include actual images, just instructions for such images to be reproduced client-side.

For example, a realistic AutoCAD file already exists, depicting the ABB Telerobot. It is approximately 5 MB in size. The primary arm yaws and pitches, the secondary arm pitches, and the manipulator pitches, rolls, and grasps. This results in six movements to track, and in this case it need not be accomplished by sensors (the instructions to the robot could be intercepted and duplicated, rather than monitoring their effect). The result of collisions with the experiment's wooden blocks is more problematic and could either be modeled by a client-side physics engine or the blocks themselves could be tracked by sensors (perhaps some sort of RADAR-like system, with each block holding a weak radio beacon, or a computer vision recognition system to detect the orientation and position of each block based on comparison with a database of known blocks). Each student could be provided with a CD holding the AutoCAD file, software (perhaps Java, using the Java3D API and converting the AutoCAD file to a J3D-readable format) to display and move the objects according to downloaded data, and a web-browser. The only distinction between such an arrangement and a fully virtual laboratory is that this arrangement still relies on physical equipment rather than software-encoded models of a system's

² HMI: Human Machine Interface, which allows the student to control the equipment and, in these applications, provides the student with audio-visual depictions of the experiment.

³ Considered reasonable given the time required to develop the project, and also given that the initial participants will be engineering students.

likely response to parameters. This means that it is more realistic: provided that no unintended objects enter the experiment, the consequences of a student's instructions never outwit the programmer. Such an arrangement is preferable to downloading camera-feeds because:

- the huge decrease in data transferred allows more object movements to be transmitted each second, thereby increasing the 'frame'-rate, probably to something approaching 'real-time';
- the image is three-dimensional⁴ and therefore richer and more immersive;
- the image can be resized, zoomed, rotated and translated to give a preferred point-of-view;
- because the graphics are client-side, they can have larger dimensions, higher resolution, greater colour-depth, and more detailed geometry; and
- server-side cameras, cables, and image processing are no longer required (at least, not for this function).

If the primary criterion is satisfied, further criteria remain to be met:

(2) The experiment must not become less accessible to students.

This is likely to be true in most cases, as computers are an essential component of the course and are often used as tools for people with disabilities. There is an opposing argument, however: modifying the laboratory exercises into individual tasks may further alienate (albeit to a small extent, most probably) people who already struggle to participate in and receive acceptance from society.

(3) The experiment must automated with safety as the first concern.

In many situations the experiment will become more safe simply by removing students from direct contact with the equipment. If the control system for the experiment is configured to return to a safe default state whenever the control system fails or variables exceed specified values (for example, a drive-wheel exceeds its maximum permitted velocity because the transmission-belt has snapped, or an electric iron exceeds its maximum allowed temperature, or an infra-red barrier surrounding the equipment is disrupted), then the automated system can be set up to run with a sufficient level of safety.

The requirement for a 'sufficient' level of safety means that the real-time actuator control and sensor detection must be local to the experiment's equipment, and must be robust. It must be local because any data connection more complicated than a direct cable connection (such as a fieldbus, radio-transceiver, or LAN) introduces an element of unreliability so that control across such a connection is unsafe. The local controller must also itself be robust if even local control is to provide safe running of the experiment. Given this need for robust local controllers, Programmable Logic Controllers (PLCs) are the best choice for local control, as computers are not sufficiently robust of hardware and software. Thus, each experiment will need its

⁴ It is rendered onto a flat screen, of course, but can be manipulated by the software as if it were a physical object – ie rotated and translated.

own PLC, which sits near the experiment, controlling all of the actuators used to manipulate the experiment and all of the sensors used to check for safety (and probably also the sensors which provide experimental data, where these are not one and the same).

(4) The total cost of ownership of the experiment must not increase by more than the value of the net benefits of automating the experiment.

The total cost of ownership both under the current system and a proposed system is relatively easy to estimate, once a system proposal is fully detailed. Valuing the benefits and disbenefits of an automated laboratory over the traditional system is less straightforward, incorporating as it does elements of quality of student learning, accessibility of laboratories to students, improvements in departmental timetabling, and assistance in attracting prestige and research-funding to the university (and the department).

(5) The experiment's procedure must be capable of being translated into a set of simple GUI commands, executable by few actuators.

If a laboratory exercise requires a student to stir the contents of a vessel, they should not have to estimate how rapidly or forcefully to stir. The commands entered into the GUI by a student must not demand an intuitive 'feel' for actions normally performed according to touch or other senses beyond the scope of the interface.

Similarly, the procedure should not require students to rotate the mouse smoothly anti-clockwise at a steady rate whilst entering data via the keyboard and pressing the left mouse button once per rotation, or any other unreasonable demands (almost any simultaneous or rapid-sequence input activities may be deemed too demanding, except for dragging a mouse whilst holding down a button, or clicking a button twice in quick succession).

The automated laboratory should also not require an abundance of actuators, for two reasons. Firstly, any experiment requiring many actuators is likely to be too complex to be effectively performed via a HMI, and likely to require far too many instructions from the student for each significant step in the procedure. Secondly, a large number of actuators will get in the way of each other and of the equipment itself, obstruct maintenance of equipment and actuators, and provide too many opportunities for devices to fail. Also, numerous actuators will influence the total cost of ownership (criterion 4).

REMOTE EXECUTION OF LABORATORY EXERCISES

Once a laboratory exercise has been designed to satisfy the above criteria, the HMI and the control system must be designed to accomplish the following tasks:

- 1) Correctly allow the experiment's variables to be altered by the student (within allowed ranges) – observing the effects of changes to experimental parameters is a valuable comprehension aid;**

- 2) Supply to the student information and background theory about the exercise, and prompt the student through the experiment's procedure – the student must understand the purpose of the exercise, what they are required to accomplish (and submit), and the theoretical basis for the experiment;
- 3) Capture experimental data and provide it to the student in a useful form – both to aid in understanding immediately and for later use in reports;
- 4) Control the laboratory equipment and revert to default 'safety values' if necessary;
- 5) Provide the data necessary for a sufficiently rich (ie large, high-resolution, many-coloured images – with rapidly refreshed motion, where appropriate – and adequately sampled sound) audio-visual depiction of the equipment as the exercise is conducted;
- 6) Log all accesses, access durations, generated data, errors and alarms – so that the system administrator can construct a chart of students' progress, provide the student with their data if theirs is lost, and monitor the system for problems; and
- 7) Permit access to laboratory web-sites, to identified students (possibly requiring the implementation of a queue and/or schedule). Visitors who are not enrolled in the unit could be referred to an example of a previous student's laboratory session.

Task (1) is accomplished by the student accessing the SCADA⁵ software on the central controller PC (see Figures, pages 14 and 15). The actual process involved varies with the software chosen – Citect⁶ uses an Internet Display Client, BridgeVIEW⁶ uses an internal component named NetDDE to transmit data from the BridgeVIEW software on the controller PC to a BridgeVIEW 'executable file' on the student's PC – one executable file is created for each experiment, defining that experiment (ie the file constitutes the control instructions for that experiment). This file is distributed (with no licencing costs) to anyone who needs to do the experiment.

Task (2) is accomplished using a combination of web-pages (to present theory and describe the procedure) and the software utilized in task (1). Citect displays can have any ActiveX object embedded, including web-pages, so the control interface and the accompanying web-pages could be integrated.

Task (3) is accomplished by the SCADA software on the central controller PC, and then transmitted to the accompanying software (Citect Internet Display Client, or BridgeVIEW executable file) on the student's PC. Data acquisition is a core task of SCADA software.

Task (4) is a critical real-time concern and is therefore handled by a PLC adjacent to the laboratory equipment (and directly connected to all of its actuators and feedback sensors). The SCADA software on the central controller PC queries these PLCs via a communications bus⁷, acquiring data to use in other tasks. The SCADA software

⁵ Supervisory Control And Data Acquisition

⁶ See *SCADA Software*, below.

⁷ See *FieldBuses*, below.

can also change parameters in the PLC as the PLC programming permits, allowing students to perform task (1).

Task (5) is discussed in the above section regarding the primary design criterion. If the chosen method uses client-side 3D graphics objects positioned and moved according to data acquired from the server, then the provision of audio-visual depictions of the laboratory equipment is accomplished by a combination of pre-drawn image files, sensors (in most cases) to track movement of the equipment, the SCADA software to acquire this motion data and send it to the student, and a proprietary program to accept this data and translate it into movement of the graphics objects.

If the chosen method uses 'closed-circuit'⁸ digital video cameras, then the cameras must transmit their data to some image acquisition and conversion software, which in turn must send the images in some compressed, optimized format, to the student's PC (via the controller PC or another server), where the Citect IDC or BridgeVIEW executable will display them.

This introduces a problem: the closer to the camera that the image processing, conversion, and compression is performed, the less the amount of data transmitted over the image-acquisition fieldbus and the less expensive that fieldbus can be for the same level of performance (number of bits transmitted per second). But this processing requires expensive devices (ie computers) – thus either the video fieldbus technology must have extremely high throughput, or the image processing must be done prior to the data entering the video fieldbus. A better alternative may be to forego the cabling advantages of a fieldbus and connect each camera directly (through a video multiplexor) to the Telerobot's image processor, using amplifiers to sustain the signals along additional cable-lengths⁹. This centralizes and integrates the video capture, processing, and transmission (via the Telerobot web server) tasks, although the central controller PC would have to communicate with the Telerobot server to provide the latter with destinations for its images.

It is possible that in 2000 or 2001 National Instruments will release image acquisition boards as FieldPoint modules (currently no image acquisition devices are available on FieldPoint – all of National Instruments' current IMAQ cards connect directly to a computer's PCI slot), and also market FieldPoint Network Modules with a radio transceiver interface (RS-232, RS-485, Ethernet, and Foundation Fieldbus interfaces are currently available). This would allow digital cameras to be connected by cable to a nearby FieldPoint module, which could then transmit processed image data by a radio signal to a central computer.

Once the images are transmitted by a web-server to the student's PC, they are embedded in web-pages or the appropriate software interfaces (BridgeVIEW/Citect) discussed above.

Task (6) is elementary to any SCADA application. Task (7) may also be handled by the SCADA software, or may be easily done using HTML and Java or Javascript.

⁸ ie cameras which do not store the images (except perhaps to buffer them), but transmit them immediately.

⁹ This information supplied by Associate Professor James Trevelyan in a facsimile message, 2/9/99.

CONTROL SYSTEM COMPONENTS

SCADA: Supervisory Control and Data Acquisition Software

SCADA software communicates with fieldbuses, programmable controllers, and devices, requesting data from them and changing values when permitted to do so by the controllers' instructions. It does not actually control the devices except to the extent of changing certain parameters, nor does it program the devices in most implementations.

SCADA software interfaces with the real-time local controllers (eg PLCs), acquires data from them, and uses this data to generate logs, databases, alarms, and graphics (charts, etc). The GUI is customized during configuration to represent the particular plant (or laboratory exercise) being controlled by that project. The software may communicate directly with PLCs, sensors, and actuators, or through a fieldbus.

A limited review was conducted of several SCADA vendors recommended as providing good after-sales support in Perth. Intellution and Wonderware are too limited in functionality (in particular internet connectivity) and composed of many component applications rather than an integrated package. Wizcon for Internet is written in Java 1.1 which is too prone to errors in multi-threaded, numerically intensive applications, and handles graphics and GUIs poorly. National Instruments BridgeVIEW automation software, with integrated LabVIEW graphical development environment, and Citect V5.2 industrial automation software were selected as two superior alternatives.

Both applications operate in Windows 95 or NT and provide all the functionality required for this project in an integrated, sophisticated package. Training is available, and they are well supported by device manufacturers. Citect was recommended as being an application of high quality, strongly supported in Perth by Ci Technologies in Applecross. A Citect Project file would be configured for each laboratory exercise, containing instructions for communicating with the experiment's devices and controllers, and displaying data to the user. Citect can be used as a Citect Internet Display Client, in which it can be restricted to run-time functions (no elements of the Citect project can be configured). Citect licences are required only for the number of simultaneous instances running, which would mean one floating licence per exercise, and one permanent licence for the central controller PC. An email from Ci Technologies' Keith Ong is appended, advising licence costs as a function of the number of 'points' (a point is a variable representing a sensor or actuator, or some other device¹⁰ to be communicated with). Citect is available in an 'educational' version, which automatically closes eight hours after being opened (but can be immediately re-opened), and can handle up to 40,000 points.

BridgeVIEW¹¹ is well supported in devices simply by being a product of National Instruments. It incorporates the popular LabVIEW virtual instrumentation software, with its graphical programming language G which facilitates programming of the control system's instructions, so that real-time control of the system devices (and

¹⁰ or something analogous to a device, such as a memory address.

¹¹ Material and information on BridgeVIEW and other National Instruments products was provided by Icon Technologies (a business card is included in this report).

programming the instructions which establish that control) converges with data acquisition and supervisory functions. This is in contrast to Citect, which has no functionality to program PLCs or other devices, merely to query such devices and interact with them based on their existing pre-programmed instructions. A single licence of BridgeVIEW can only be installed on one machine, but it can be used to create BridgeVIEW executable files, which can be freely distributed and which are the files which actually run the experiment (in communication with the parent program). They perform the same functions as a Citect Internet Display Client running a Citect project file.

FieldBuses

A fieldbus is a communications network (usually comprised of cables, with junctions, repeater devices, and similar components) designed to transmit data (and in some instances electrical power) between devices. Prior to their development, devices were directly connected to controllers, so that controllers either had to be adjacent to the devices, or large lengths of cabling were required for each device. A central SCADA platform would require controllers to be connected to it also. The usefulness of fieldbuses arises because of this high cabling cost. Long cables for each device are replaced by a bus-cable which runs throughout the facility. Each device is connected to the bus at the node closest to where the device is used. Data is placed on to the bus by the devices, and travels around the bus to its destination (a device which has requested the data). Data can be placed on to the bus at any node, and read off the bus at any node, rather than travelling a cable between the source device and one other device. Each type of fieldbus is described by a design protocol which stipulates the type and configuration of physical components, the voltage of the cable signals, the format (message-bits, data-bits, and each bit's significance) of the data messages, and the protocol devices must follow in placing messages on the bus, reading them off the bus, and passing them along the bus.

Many fieldbuses were investigated (ignoring the requirement for good support in Perth, although this would be a crucial factor in a decision on fieldbuses) using the internet¹². Various reasons for unsuitability (distance between nodes being too short, nodes being too few, data transfer rate being too slow, data overhead being too large, or being unavailable in Perth), eliminated most products, leaving Interbus, Profibus, and Foundation Fieldbus. The prospect of FireWire fieldbuses being developed was also considered.

In the initial implementation, at least, the fieldbus used should be the engineering building's existing 10BaseT UTP-cable TCP/IP network, to reduce costs and complexity. More thorough analysis of fieldbuses is required before a subsequent decision is made. Interbus is an attractive protocol for its 'looped' topology (allowing rapid detection and replacement of faulty components), but is not as widely supported (by PLCs) as Profibus or Foundation Fieldbus.

FireWire

FireWire (IEEE standard 1394) is a type of high-speed input/output bus developed by Apple Computer Inc for use in connecting peripherals with large data I/O demands (such as digital camcorders) up to computers. In many ways it is a higher-throughput cousin of the Universal Serial Bus – FireWire-connected devices are hot-pluggable

¹² As of Tuesday October 12, no email requests for further information have been answered.

(unlike “Plug’n’Play” devices), automatically configured upon plug-in, and (in contrast to SCSI devices) needs no device identity numbers or cable-terminators. Whereas a USB port can support up to 127 devices; a FireWire port up to 63. A FireWire bus consumes about 6 Watts and transmits data at 400 Mbps (50 MB per second) in it’s first specification, and 1.2 Gbps (150 MB per second) in the emerging second version. FireWire would be ideal as a fieldbus, with only one flaw – the maximum length of a FireWire cable is 14 metres. Given the relatively low cost of FireWire, however, a computer dedicated to providing FireWire buses to all experiments within a 28 metre radius seems practical.

Local Control

PLCs

Programmable Logic Controllers are established, well-understood devices for controlling sensors and actuators in factory and laboratory automation. Although less user-friendly than computers, their simplicity of construction and of software renders them more robust and reliable. PLCs may have analogue and/or digital input and output terminals (‘points’), to accept a range of electrical or electronic devices. Control of and communication with devices is accomplished by variations in the voltage or current signals of devices. PLCs are typically programmed by the uploading of instruction-code over a serial cable connection, from a PC with the PLCs primitive (proprietary) programming software installed.

In deciding which PLCs to use in laboratory automation, consideration should be given to cost, after-sales support, compatibility with other components, and the number of points the PLC has. The PLC must be matched to each laboratory exercise’s requirements, including operating environment, the number and type (digital or analogue) of points needed, and the voltage and current demands of the devices. Modular PLCs have the benefit of not being categorised according to their number of points, because an additional module can be attached if more points are required. However, if the PLC is modular as part of a rack, then the scalability is limited to the capacity of the rack to accept more modules. Modular PLCs are also likely to have higher purchase costs. It seems logical to limit the PLCs used in any network of automated laboratories to one brand (and preferably one model in that brand), to save training, servicing, replacement and other maintenance costs.

FieldPoint Distributed I/O System

FieldPoint devices are an alternative type of controller developed and marketed by National Instruments. Some information about them is appended to this report. They are newer and less well-known than PLCs (and therefore less familiar to maintenance personnel), but offer some advantages in that they can be easily programmed over a network by configuration software. Future developments may include radio transceiver network modules, removing the need for physical connections to the central controller PC, and image acquisition I/O modules.

APPLICATION: THE CA205 ELECTRIC IRON LABORATORY EXERCISE – 2 ALTERNATIVES

Materials relating to the Computer Applications 205 laboratory exercise “Electric Iron Modeling Exercise” are appended to this report. Two alternative system designs are presented – one based on Citect, the other using National Instruments equipment (assuming some devices are released in 2000/01 as indicated). The principle followed in these designs is that the design should be scalable, so that an entire system can be constructed using multiples of the equipment presented below. This was originally intended to be demonstrated in a prototype system, but there proved to be insufficient time to learn Citect or BridgeVIEW, and how to program a GE Fanuc PLC.

The sensory data supplied by the experiment is low – the students see an electric iron, and if they are rash they can feel the heat of it’s face-plate. There is no kinetic, olfactory, auditory, tactile, or thermal data of any significance. A large, true-colour image of the iron and a close-up of a thermocouple suffices to reproduce the visual data with adequate richness.

The experiment is currently quite accessible to all students. It is not run at scheduled times (students do the experiment when the iron is unused), and if access is a problem then the iron could easily be re-located.

The relay switch (see below) should fall into the ‘off’ position when the PLC fails (and the PLC should set the switch to ‘off’ whenever a critical error occurs). If this is done then the experiment has become more safe (because students are removed from the iron).

The total cost of ownership is likely to increase discommensurately, because the cost of the electric iron exercise is currently so low. However, for the purposes of this report this fact is ignored (as the intent is to demonstrate a scalable automated network structure).

Control of the experiment through a GUI will be extremely simple, involving one button to turn the iron on and off. The display will also include a timer and thermocouple read-outs. The only actuator needed is a relay switch.

Sensing and actuating requirements

(1) The iron: power supply (mains AC).

(2) AC power on/off by means of a relay switch connected to a digital output terminal of the PLC or FieldPoint 8-Channel SPST Relay Module¹³, with the active cable of the iron’s power supply running through the relay switch, and the ground cable of the iron’s power supply connected to the ground of the PLC/FieldPoint. Electric irons are typically rated at about 1100 W, so a maximum current of 4.6 rmsAmps must be accommodated by the relay terminal.

(3) A thermocouple or RTD, connected through a temperature transmitter to an analogue input terminal of the PLC. A FieldPoint device would instead use either an 8-Channel Thermocouple Input Module with a thermocouple connected, or an 8-Channel RTD Module with an RTD sensor connected, and the module configured to return data scaled to temperature (Celsius and Kelvin are both available). To do this,

¹³ Specifications for the National Instruments components may be accessed on the Instrupedia CD-ROM.

the RTD must be platinum and have a temperature coefficient of resistance of 3.750, 3.851, 3.911, 3.916, 3.920, or 3.928 m/_/_°C (other RTDs can be used but calibration must be performed manually). The thermocouple or RTD is attached to the face-plate of the iron. A second thermocouple or RTD is used to measure the ambient temperature.

Requirements	
Citect	National Instruments
----- Electric Iron (1100W) -----	
----- Relay switch -----	
PLC	FieldPoint Ethernet Network Module FieldPoint Terminal Base FieldPoint 8-Channel SPST Relay Module
----- 2 Thermocouples or RTD sensors -----	
2 Temperature transmitters	8-Channel Thermocouple Input Module OR 8-Channel RTD Module (both with a 2nd FieldPoint Terminal Base)
Fieldbus compatible with PLC Controller PC running Citect Student PC running Citect Internet Display Client	existing 10BaseT LAN as fieldbus Controller PC running BridgeVIEW Student PC running a BridgeVIEW executable file

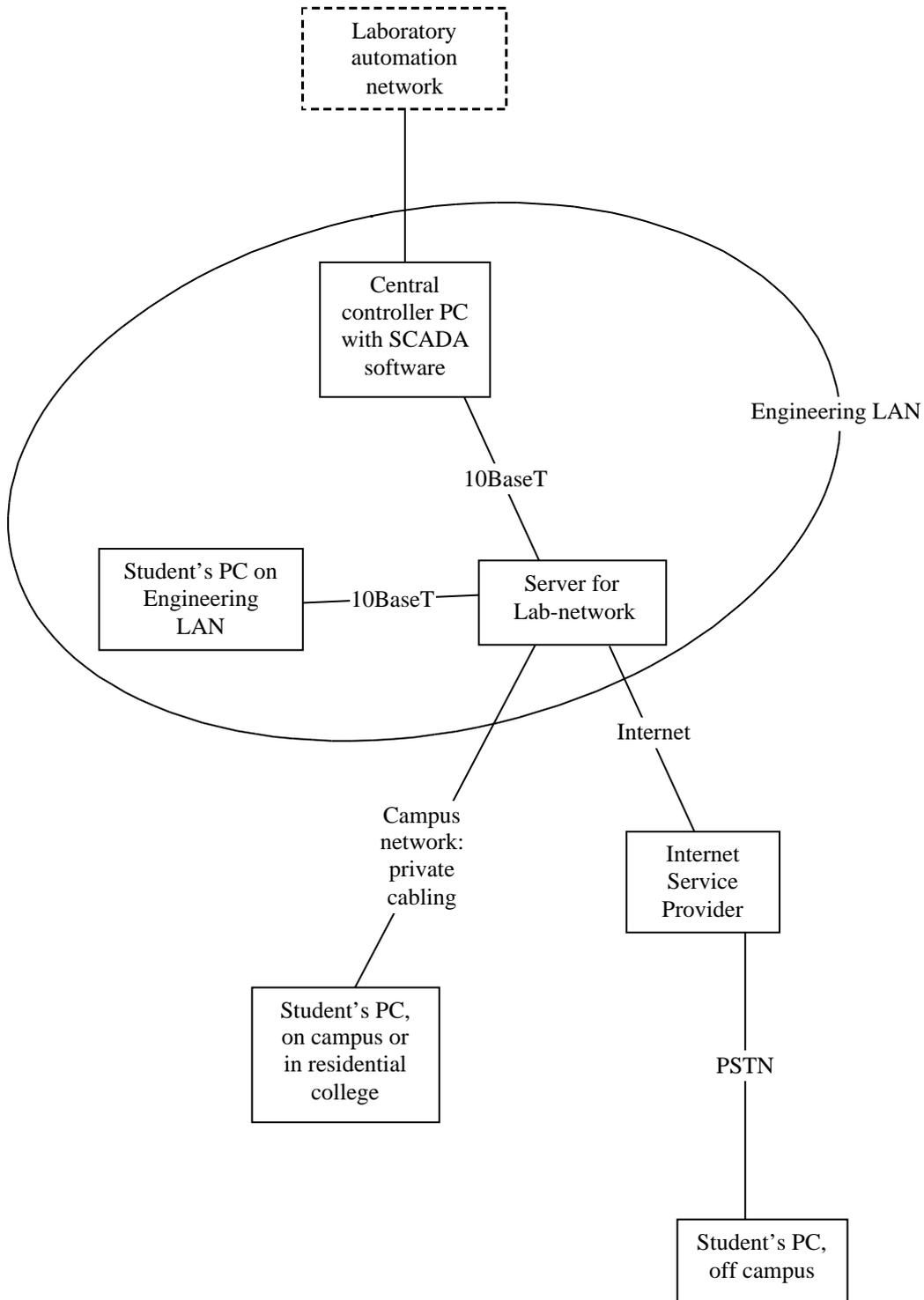
ACKNOWLEDGEMENTS

Except where otherwise accredited, almost all of the knowledge underlying this report was supplied by Jan Baranski (see attached business card), or through materials provided by him. The remainder I either knew already or obtained from the Internet and product manuals (as there seemed to be a paucity of text-based materials on the subjects of internet-based control systems and image acquisition in control systems. Indeed, a simple keyword search for “fieldbus” at the university library yielded one result, which was irrelevant. There was a small amount of material on more traditional control systems, but none relating to the areas discussed in this report.)

Dr Mark Trotman of Icon Technologies (see attached business card) provided information on the National Instruments components (BridgeVIEW and FieldPoint). Keith Ong of Ci Technologies Pty Ltd (see attached business card) supplied information on Citect licensing and prices.

Two GE-Fanuc PLCs were generously provided by Motherwell, but were not used in this project due to difficulties in learning how to program them. This is the reason for there being no prototype system included in this project.

**COMMUNICATIONS INFRASTRUCTURE FOR REMOTE ACCESS TO THE NETWORK OF
AUTOMATED LABORATORIES**



AUTOMATION INFRASTRUCTURE FOR THE NETWORK OF AUTOMATED LABORATORIES

