Architecture abstraction as an aid to Computer Technology Education

S P Maj, D Veal
Department of Computer Science
Edith Cowan University
Perth, Western Australia

Abstract

Reports such as the 1991 ACM/IEEE-CS Joint Curriculum Task Force set benchmarks for award accreditation and provide the foundations of computer science curriculum worldwide. The report identifies recognizes the 'need for diversity and well-intentioned experimentation in computing curricula'. Computer Science is a relatively new discipline and given the rapid advances in technology is subject to on going debate, development and fragmentation. It is typically the requirement of many disciplines, such as Multi-media, Software Engineering, E-commerce etc to incorporate computer technology as part of their curriculum. However, a detailed market analysis within Australia clearly indicated that both students and employers perceive the standard computer technology curriculum as increasingly irrelevant.

Work to date clearly indicates that this standard approach provides technical detail and complexity that is inappropriate for introductory courses on computer and network technology. As part of an international study the same investigation is currently being conducted with several European universities. The results to date parallel those obtained from the WA study. Accordingly a new curriculum was designed to address this problem. This new curriculum is based on a modeling a PC as an interconnection of nodes. Evaluation of the curriculum indicates that this abstraction can be used as a new educational framework allowing technical detail to be introduced and controlled thereby ensuring that it is meaningful and therefore readily understandable to students not only from computer science but also other disciplines. Work to date indicates that this new model is not only technically valid but also supports increasing levels of technical complexity and hence articulates to the standard computer technology curriculum. Furthermore the abstractions used in this model are independent of technical detail and can therefore accommodate rapid changes in technology.

1. Introduction

Reports such as the 1991 ACM/IEEE-CS *Computing Curricula* [1] provide the foundations of computer science curriculum world wide and set benchmarks for accreditation by professional bodies. Within Western Australia an exploratory market audit was conducted of a wide range of industrial and commercial companies. This was complemented by a further detailed analysis of the IT department of a statewide rail company. From this survey a set of guidelines were developed for the type of skills expected of computer science graduates

entering the field of computer and network support. Using the criteria developed, a random selection of ten final year ECU computer science undergraduates were interviewed from a graduating population of approximately one hundred. According to Maj:

> "It was found that none of these students could perform first line maintenance on a Personal Computer (PC) to a professional standard with due regard to safety, both to themselves and the equipment. Neither could they install communication cards, cables and network operating system or manage a population of networked PCs to an acceptable commercial standard without further extensive training. It is noteworthy that none of the students interviewed had ever opened a PC. It is significant that all those interviewed for this study had successfully completed all the units on computer architecture and communication engineering" [2].

The computer architecture and communication-engineering units were: Computer Technology, Microprocessors, and Data Communication & Computer Networks. These units follow the standard approach taken by most universities. The Computer Technology unit introduces students to computer systems and hardware i.e. number codes, assembly language (Motorola 6800), machine architecture etc. The Microprocessor unit is a detailed examination of microprocessor technology and an in-depth treatment of assembly language (Intel). The Data Communication & Computer Networks unit provides an understanding of the physical and logical elements of data communications with a detailed discussion of the ISO OSI model. Furthermore, interviews conducted with five ECU graduates employed in computer and network support clearly indicated that they were, to a large degree, self-taught in many of the skills they needed to perform their job. Preliminary investigations indicated a similar situation with computer science graduates from other universities within Western Australia. This problem is exacerbated not only by the constant and rapid changes in technology but also the requirement to teach technology to students from a wide range of discipline such as multimedia, e-commerce etc [3]. According to Campus Leaders:

> "the predominant reason why they (students) have gone to university was to get skills, knowledge and a qualification that would assist them in either gaining employment or enhancing their prospects for promotion or a more rewarding job." [4]

Other countries similarly have professional accreditation. In the United Kingdom (UK) the British Computer (BCS) accredits university courses and has an internationally recognized examination scheme in two parts with Part II at the level of a UK honors degree in computing. The initial ECU student questionnaire, first used in 1993, was also conducted in 1999 at two universities in the UK. A similar study is currently being undertaken in Sweden. The first university has well-established degree programs and is fully BCS accredited. The second university recently redesigned their IT awards, some of which are now BCS accredited. The degree programs at the first university offer students the opportunity to examine a PC in the first year as part of a module in Computer Organization. However they never take a PC apart. Students are taught network modeling, design and management but they do not physically construct networks. The results clearly demonstrate that students lacked knowledge about PC technology and the basic skills need to operate on computer and

network equipment in a commercial environment. This is despite the fact that most students thought such knowledge would be beneficial. The survey indicated that any practical knowledge students have of hardware is largely a result of experience outside the course. At the second university the results demonstrate that these students had a broad, hobbyist's understanding of the PC but no knowledge of health and safety law. Significantly, the students interviewed identified that their skills and knowledge of PCs and networks came from self-study or employment, not from courses at university. Again student responses indicated that such knowledge would be useful. Furthermore, according to a study of Multi-media students by Maj,

> "It is significant that every student interviewed expressed the view that it would be extremely beneficial to have a much better knowledge of computer and network technology" [3].

A demand that traditional courses in computer technology may not be meeting. We therefore examined developments in computer and network technology curriculum.

2. Computer and Network Technology Curriculum

The problems associated with teaching computer technology are not new. Units in microcomputer systems are fundamentally important to both computer science and engineering students. These address issues that include: computer organization, memory systems, assembly language, digital logic, interrupt handling, I/O and interfaces. Mainstream computer science education is well supported by journal articles on various aspects of re-programmable hardware for educational purposes [6] and assembly language [7]. Simulation has proved to be a very useful tool [8, 9, 10]. Reid used laboratory workstations to allow undergraduate students to "build a complete, functioning computer - in simulation" [11]. Pilgrim [12] took an alternative approach in which a very small computer was designed in class and breadboarded in the laboratory by students using small and medium scale TTL integrated circuits. Thereby, according to Pilgrim, providing students with the "knowledge and experience in the design, testing and integration of hardware and software for a small computer system" [12]. According to Parker and Drexel simulation is a preferred approach in order to provide students with the 'big picture' [13]. The difficulty of providing a suitable pedagogical framework is further illustrated by Coe and Williams. Coe et al address this problem by means of simulation [14]. Barnett [15] suggests that standard computer architecture is too complex for introductory courses and recommends a simplified computer for educational purposes.

However, it is possible to consider the PC and network technology from a different perspective. The PC is now a (relatively) low cost consumer item. This has been possible due to design and manufacturing changes that include: Assembly Level Manufacturing (ALM), Application Specific Integrated Circuits (ASICs) and Surface Mounted Technology (SMT). The result is PCs with a standard architecture and modular construction – so simple that high school students take them apart. However, traditionally computer technology education is typically based on digital techniques, small-scale integration IC's, Karnaugh maps, assembly language programming etc. Operation on PCs at this level simply does not exist any more within the field of computer and network support. Valuable though simulation and breadboarding may be a typical PC support environment demands other knowledge and

skills that include: upgrading PCs, fault identification and correction procedures, safety, ability to recognize different system architectures etc. Simulation provides no experience of practical problems such as inserting a new input/output card into a PC and the associated skills that are needed. Furthermore, a new conceptual model is needed that provides abstraction in order to control detail is required as the foundation of curriculum for the diverse audience now wishing to study computer technology.

3. Constructivism

Prior to examining how to improve student learning we attempted to attain a deeper understanding of how students learn and construct knowledge. Constructivism is the dominant theory of learning today, the basis of which is that students must actively construct knowledge rather than passively absorb it via lectures. According to Ben-Ari considerable research has been undertaken in this field but commented:

> "However, I could not find articles on constructivism in computer science education compared to the vast literature in mathematics and physics education' and that 'it can provide a new and powerful set of concepts to guide our debates on CSE (Computer Science Education)" [16].

According to this theory students have their own cognitive structures each of which is the foundation of the learning process. By example, a PC is understood very differently by Computer Science, Multimedia and Business IT students. Failure to do so results in fragile and incomplete learning in which new knowledge is merely a collection of facts to be memorized [17]. The importance of the students own mental model is illustrated by Scott Brandt who wrote, "The user's ability to apply a previously held mental model to the target (knowledge goal) will enhance the incorporation and construction of new knowledge" [18]. We suggest that it is a common experience to perceive the PC as a modular device (CDROM, Zip Drive, Modem etc) used to store, view and process either local or networked data. The traditional method of teaching computer technology (digital techniques, assembly language etc) is not a good constructivist approach. According to Scragg:

> 'most (perhaps all) first courses in computer hardware are created "upside down" - both pedagogically and pragmatically'. This has the consequence that 'Pedagogically, this approach provides no "cognitive hooks", which might enable students to relate new material to that of previous courses - until the semester is almost complete" [19].

Accordingly Scragg recommends a top down approach starting with material already familiar to students and then working towards less familiar material. We attempted therefore to find a common conceptual framework held by students, from different disciplines (especially multi-media), as the basis for a cognitive structure.

4. The PC – a constructivist model

Models are used as a means of communication and controlling detail. By example, a transistor can be modeled by a simple diagram with parameters directly relevant to an engineer. The details of semi-conductor theory are not relevant in this context i.e. detail is encapsulated and hence controlled. Similarly digital techniques such as sequential logic are a higher level modeling technique that masks the details of individual transistors. Models should have the following characteristics:

- Diagrammatic
- Self documenting
- Easy to use
- Control detail
- Hierarchical top down decomposition

Clements suggests that a PC may be considered as a loosely coupled Multiple Instruction, Multiple Data (MIMD) device. According to Clements:

> "Although most people do not regard it as a multiprocessor, any arrangement of a microprocessor and a floppy disc controller is really a loosely coupled MIMD. The floppy disc controller is really an autonomous processor with its own microprocessor, internal RAM and ROM."

and also that:

> "Because the FDC has all these resources on one chip and communicates with its host processor as if it were a simple I/O port, it is considered by many to be a simple I/O port. If it were not for the fact that the FDC is available as a single chip, engineers would be designing "true" multiprocessor systems to handle disc I/O" [20].

A PC is a complex collection of heterogeneous devices interconnected by a range of bus structures. However it can be modeled as a MIMD architecture of sub-units for nodes. Each node (microprocessor, hard disc drive etc) can be treated as a data source/sink capable of, to various degrees, data storage, processing and transmission. This simple model may provide the basis of a suitable conceptual map and hence the framework for an introduction to computer technology. This model is conceptually simple, controls detail by abstraction and may allow students to easily make viable constructs of knowledge based on their own experience. However, to be of significant value this model must also be a tool that can actually be used by students. Furthermore the model must be not only be technically valid but also provide a basis for more advanced studies. We therefore examined PC performance to further develop this model.

5. PC Performance - Bandwidth nodes

Benchmark programs considered directly relevant to a typical single user, multi-tasking environment running a de facto standard suite of 32 bit applications include : AIM Suite III, SYSmark and Ziff-Davis PC Benchmark. Consumer magazines use Benchmark suites to

evaluate PC's and publish their results [21]. Intel mark their microprocessors with a part number and the maximum rated clock speed. Furthermore they publish a special series of Benchmarks called the Intel Comparative Microprocessor performance Index (iCOMP) that can be used as a relative gauge of microprocessor performance. For many AMD microprocessors the model designation does not correspond with the associated clock speed. For example, the AMD K5 PR133 has a clock speed of only 100Mhz. Cyrix, IBM, SGS Thompson, and AMD jointly developed the alternative P Rating (PR) system. This Benchmark is based on the Winstone, a de facto standard, windows based Benchmark suite. As an aid to more meaningful measurements specialist interest groups also evaluate equipment using specific applications [22]. As a relative guide Benchmarks are an aid to selection, however, all of these results must be interpreted and many questions still remain for users. Questions include:

- What difference in performance can a user expect if the bench mark value result is higher by 1 or 2 units or by a factor of 10 or more? For example, what difference in performance would a user expect between an IBM Aptiva EQ3 (Business Disk WinMark 98 value of 939) and a Gateway G6 300 (Business Disk Win Mark 98 value of 1,380)?
- How does the iCOMP rating compare to the PR rating?
- What difference in performance can a user expect from a Pentium 100 (iCOMP 90) and a Pentium 200 (iCOMP 142)? Are the scales linear, logarithmic, hyperbolic?
- As a user, how is the difference in performance manifested and perceived?
- How can different types of devices be compared, e.g. how can the performance of a hard disc drive be compared to a microprocessor?

We conclude that the plethora of benchmarks, though useful, do not provide the basis of a coherent conceptual model of a PC. Any measurement standard, to be of practical value to PC users, must be relevant to human dimensions or perceptions and use units based on the decimal scaling system.

To a first approximation the performance of PC nodes can be evaluated by bandwidth with units in Bytes/s. Though useful it may not be the best initial, user oriented unit - a typical user runs 32 bit windows based applications. The user is therefore interacting, via a Graphical User Interface (GUI), with text and graphical images. For general acceptance a Benchmark must be easy to understand and should therefore be based on user perception of performance and as such be simple and use reasonably sized units.

We suggest that a useful unit of measurement is the display of a single, full screen, full color image. For this paper we define a full screen image as 640 x480 with 4 bytes per pixel, which represents 1.17Mbytes of data. This appears to be the standard image for the new generation of video display adapters. The performance of a PC and associated nodes can still be evaluated using the measurement of bandwidth but with the units of standard images/s or frames/s. This unit of measurement may be more meaningful to a typical user because it relates directly to their perception of performance. To a first approximation, smooth animation requires a minimum of 5 frames/s (5.85Mbytes/s). Obviously sub multiples of this unit are possible such as quarter screen images and reduced color palette such as 1 byte per pixel. We first established a general experimental method to determine data transfer rates between nodes within a PC to evaluate the use of frames/s as a measurement of performance. A C program was used to transfer data between two nodes, a Hard Disc Drive (HDD) and

Synchronous Dyanamic RAM (SDRAM), flagging the start and stop of this operation on the parallel port. An oscilloscope (100 micro second resolution), connected to this port measured the data transfer rate in Mbytes/s. The results obtained related directly to the manufactures technical specification of the HDD. We were able to detect the influence of HDD caching and also track and cylinder latency thus verifying the experimental method. A single, uncompressed 640x480, 4bytes/pixel-video image was generated and transferred from the HDD to both SDRAM and a third node, the video adapter card. The data transfer rate from HDD to SDRAM was 1.48Mbytes/s, which can be expressed as 1.21 frames/s. From the HDD to video adapter card the data rate was 1.37Mbytes/s ie1.1 frames/s. The data transfer rate for the video card is 18.6Mbytes/s i.e. 15.1 frames/s. We have therefore a common unit of measurement, relevant to common human perception, with decimal based units, that can be applied to different nodes and identify performance bottlenecks. In this case the HDD is the limiting factor and unable to provide a bandwidth suitable for smooth motion in an animation sequence. The concept of using images to evaluate PC performance can be made directly relevant to users from different disciplines, in particular Mulit-media.

Each node (microprocessor, hard disc drive etc) can now be treated as a quantifiable data source/sink (Frames or Mbytes) with an associated transfer characteristic (Frames/s or Mbytes/s). This approach allows the performance of every node and data path to be assessed by a simple, common measurement – bandwidth. Where Bandwidth = Clock Speed x Data Path Width with the common units of Frames/s (Mbytes/s).

6. The PC as a Bandwidth Node Model

The heterogeneous nature of the nodes of a PC is clearly illustrated by the range of measurement units used varying from MHz to seek times in milliseconds. Evaluation of these different nodes is therefore difficult. However, it is possible to compare the performance of different nodes using the common measurement of bandwidth in standard Frames/s or Mbytes/s.

6.1 Microprocessor

All microprocessors, regardless of any internal architectural details, transfers data via the processor data bus. The Pentium processor has an external data path of 8bytes with maximum rated clock speeds in excess of 400Mhz giving a bandwidth of more than 2735 Frames/s (3200Mbytes/s).

6.2 Primary Memory – DRAM

Regardless of DRAM organizational structure, performance is measured in nanoseconds, which can easily be converted to MHz. The performance of SDRAM is now often quoted in MHz, e.g. 83Mhz (12ns) or 100Mhz (10ns). For example, Dual In Line Memory Modules (DIMMs) rated at 60ns (16MHz) with a data path width of 8 bytes have a bandwidth of 109 Frames/s (128Mbytes/s).

6.3 Secondary Memory – Hard Disc Drive

The true maximum, sustained data transfer rate in Mbytes/s can be calculated from the sector capacity and rotational speed (data transfer rate = sector capacity x sectors per track x rps). Typical figures are in the range of 4.3 Frames/s (5 Mbytes/s) with the number of sectors averaged thereby taking into account Zoned Bit Recording. The data transfer rate of interfaces is often quoted in latency, which can easily be converted to Mbytes/s or Frames/s.

6.4 Peripherals

Modem performance is typically measured in Kbits/s which can be converted to Mbytes/s or Frames/s. CDROM performance is quoted in speeds e.g. x32 speed. A single speed CDROM provides a defined transfer rate of 150kbytes/s. When CDROM devices are used to transfer digital data it is possible to speed up the data transfer rates, all of which are multiples of the original single speed drives. CDROM speeds can easily be converted to Mbytes/s or Frames/s.

6.5 Bus Structures and Chip Sets

According to Mueller [23], the maximum transfer rate of a bus in MBytes/s can be calculated from the clock speed and data width. This can then be converted to Frames/s. It is possible to construct a diagrammatic representation of the bus structure hierarchy showing also the chip-sets used to interface between the different hierarchical levels.

A PC can be therefore be understood as a hierarchical collection of nodes, interconnected by different buses, operating at different bandwidths. Work to date indicates the bandwidth node model may be used for all the latest Multi-media devices (Video camera, Digital Video Disk (DVD), flat bed scanner etc). We then examined if this bandwidth node model was valid for increasing levels of technical complexity.

7. Sub-Optimal Operation – Technical complexity

Nodes typically operate sub-optimally due to their operational limitations and also the interaction between other slower nodes. For example, a microprocessor may need two or more clock cycles to execute an instruction. Similarly a data bus may need multiple clock cycles to transfer a single data word. The simple bandwidth equation can be modified to take this into account i.e. Bandwidth = Clock x Data Path Width x Efficiency. The early Intel 8088/86 required a memory cycle time of 4 clocks cycles (Efficiency = ¼) however, for the Intel 80x86 series, including the Pentium, the memory cycle time consists of only 2 clocks (Efficiency = ½) for external DRAM [22]. A 100Mhz Pentium, with a data path of 8 bytes has a therefore bandwidth of 341 Frames/s (400Mbytes/s).

Dual In Line Memory Modules (DIMMs) rated at 60ns (16MHz) with a data path width of 8 bytes have a bandwidth of 109 Frames/s (128Mbytes/s). When DRAM devices cannot complete a read or write request within the correct number of clock cycles the memory controller indicates to the microprocessor that another instruction cycle must be implemented, i.e. a wait state. [24]. Each wait state increases the memory cycle time by one clock, hence an extra wait state for the 80x86 series would reduce the efficiency to 1/3. For a 100Mhz Pentium the bandwidth would be reduced from 341Frames/s (400Mbytes/s) to 227Frames/s (266Mbytes/s). It can be clearly seen that five wait states are needed to match the microprocessor and DRAM (60ns) bandwidths. The Peripheral Component Interconnect (PCI) bus is also a 32-bit bus but operates at a fixed frequency of 33Mhz. The PCI bus uses

a multiplexing scheme in which the lines are alternately used as address and data lines. This reduces the number of lines but results in an increased number of clock cycles needed for a single data transfer. The maximum data transfer rate for the write operation (2 clock cycles) is 56Frames/s (66Mbytes/s) i.e. (33Mhz x 4 bytes x 1/2). However, the PCI bus is capable of burst transfer modes of arbitrary length in which case E can be taken as unity.

The bandwidth node model is capable of top down decomposition. A PC can be considered as a single node or a collection of nodes (microprocessor, electronic memory, hard disc drive etc). Each of these nodes can be modeled as a collection of nodes. By example a hard disc drive can be described as two nodes, the Electro-mechanical drive mechanism (motors, platters etc) that communicates to the hard disc controller (responsible for data separation, error correction, buffering etc) that in turn interfaces to the motherboard bus.

We conclude that the proposed bandwidth node model may be extended to accommodate the sub-optimal performance of PC sub-units. Furthermore this model may be used to provide greater technical depth. However no account is taken of compression or the effect of operating system overheads. Further work is currently being undertaken to analyze the effect of these variables.

8. Architectural Independence

There are many rapid changes in computer technology – the technical standards of today are likely to be obsolete tomorrow. It is possible, using this technique, to model all microprocessors (8086 to Pentium), all types of hard disc drive (ST 506, ESDI, IDE etc) all buses (ISA, EISA, MCA, LB and PCI). Preliminary investigations indicate that it is possible to model all multi-media devices using this bandwidth node technique.

9. Nodes as a Constructivist Framework

As a result of the initial investigations at ECU a new curriculum was designed, implemented and fully evaluated at ECU [25]. Unlike the standard computer technology curriculum students are not taught digital techniques, assembly language programming etc. Rather the curriculum is based on a constructivist approach recommended by this paper. This curriculum has always been oversubscribed, has a very low student attrition rate, attracts students from other Faculties in ECU and students from other universities in the state. When one new unit from this curriculum was first introduced, from an enrolment of 118, only 66 were computer science students the others were from a wide range of disciplines, especially multi-media. Workshop exercises include: install master/slave hard disc drive; upgrade PCI video card, load an operating system. Other more advanced exercises in a subsequent unit include: the installation and testing of: Digital Video Disc (DVD), flat bed scanner, PC video camera, Infra-red communications link, Zip Disc, a video conference communications link via a local area network. Such workshops represent many of the common tasks required of students when entering the workforce.

An educational expert conducted a detailed analysis of student learning. Five students, chosen at random, were interviewed. Interviews were semi-structured consisting of a number of closed and open ended questions and respondents were encouraged to comment on any positive or negative aspect of the course and its effect on their learning. The results that the curriculum:

> "is perceived as very valuable by students from different disciplines; supports learning in other units; increases students' understanding of computers and computing; generates a demand for further curriculum in this field" [25].

This new curriculum, though perceived as valuable by students when first introduced, arguably lacked a coherent conceptual framework. Last year we used bandwidth nodes as such a framework for the first time and evaluated the results. Using this conceptual framework the PC is considered as a series of nodes that can store, process and transfer data. The operational characteristic of each node is considered in detail and its bandwidth calculated using the common units of Bytes/s and Frames/s. Using the standard ECU course evaluation questionnaire the unit was highly rated by students. Student understanding was evaluated by means of two assignments in which they were required to obtain the technical specifications for a PC and construct a nodal model with associated bandwidths in both Mbytes/s and Frames/s. Furthermore, a more detailed study was conducted. Thirty-seven students from forty responded that this method of teaching computer technology was beneficial.

Prior to the introduction of this model student previous student similar to this resulted in almost exclusively a list of hardware details copied directly from technical literature with little or no critical analysis. Most students most were able to predict the likely performance of a PC and identify nodes (devices) that would significantly handicap performance. Significantly, using this nodal model all additional progressive technical detail and complexity is additive to knowledge and understanding rather than being perceived by the student as simply a collection of factual data. Advantages to using this model include:

- Students perceive the PC as a unified collection of devices
- Node performance, measured in bandwidth (Frames/s) is a user based, easily understood measurement
- The units (Mbytes/s, Frames/s) use a decimal scaling system
- Students are able to evaluate different nodes of a PC by means of a common unit of measurement
- Students can easily determine the anticipated performance of a PC given its technical specification
- Students are able to critically analyze technical literature using this integrating concept.
- The model is suitable for students from a wide range of disciplines including Computer Science majors
- Nodes control detail and are valid for increasing levels of technical complexity.
- Nodes are independent of architectural detail
- Nodes are a diagrammatic and easy to use
- Nodes are self documenting
- Nodes can be used to provide hierarchical top down decomposition.

10. Digital Techniques

According to the report of the 1991 ACM/IEE-CS Joint Curriculum Task Force report computer science curriculum includes: digital logic and systems, machine level representation of data, assembly level machine organization, memory system organization and architecture, interfacing and communications, and alternative architectures [1]. The authors do not suggest that the bandwidth node model should replace the teaching of this traditional computer technology curriculum. Rather, it may serve as a useful introduction that provides both interesting and useful skills. Ramsden reinforces this point,

> "Material should preferably be ordered in such a way that it proceeds from common-sense and everyday experiences to abstractions and then back again to the application of theoretical knowledge in practice" [26].

11. Conclusions

This paper proposes nodes, whose performance is rated by bandwidth (frames/s), as the basis of an introduction to computer technology curriculum. Work to date indicates that modeling the PC as such a collection of nodes provides a good constructivist framework allowing technical detail to be introduced in a controlled, top-down manner that is readily understandable to students from all disciplines. This nodal model is also valid for increasing levels of technical complexity and hence may suitable for more advanced studies. The nodal model provides abstraction and hence is independent of architectural detail and can therefore accommodate rapid changes in technology.

Bibiography

1. Computing Curricula. A survey of the ACM?IEEE - CS Joint Curriculum Task Force Report. Communications of the Acm June. Vol 54, No 6. acm Press. (1991)
2. S. P. Maj, G. Robbins, D. Shaw, and K. W. Duley, "Computer and Network Installation, Maintenance and Management - A Proposed New Curriculum for Undergraduates and Postgraduates," *The Australian Computer Journal*, vol. 30, pp. 111-119, (1996).
3. S. P. Maj, G. Kohli, and D. Veal, "Teaching Computer and Network Technology to Multi-Media students - a novel approach," presented at 3rd Baltic Region Seminar on Engineering Education, Goteborg, Sweden, (1999).
4. E. T. Workforce, "Educating the workforce for the new millenium," in *Campus Review*, (1996).
5. D. J. Ewing, "Microcomputers systems 1: a computer science and engineering capstone course," *ACM SIGCSE Bulletin*, vol. 25, 155-159, (1993).
6. M. Gschwind, "Preprogrammable hardware for educational purposes," *ACM SIGCSE Bulletin*, vol. 26, 183-187, (1994).
7. W. A. Coey, "An interactive tutorial system for MC68000 assembly language using hypercard," *ACM SIGCSE Bulletin*, vol. 25, 19-23, (1993).
8. D. Magagnosc, "Simulation in computer organisations: a goals based study," *ACM SIGCSE Bulletin*, vol. 26, 178-182, (1994).
9. D. E. Searles, "An Integrated Hardware Simulator," *ACM SIGCSE Bulletin*, vol. 25, pp. 24-28, (1993).

10. S. D. Bergmann, “Simulating and Compiling a Hypothetical Microprogrammed Architecture with Projects for Computer Architecture and Compiler Design,” *ACM SIGCSE Bulletin*, vol. 25, 38-42, (1993).
11. R. J. Reid, “A laboratory for building computers,” *ACM SIGCSE Bulletin*, vol. 24, 192-196, (1992).
12. R. A. Pilgrim, “Design and construction of the very simple computer (VSC): a laboratory project for undergraduate computer architecture courses.,” *ACM SIGCSE Bulletin*, vol. 25, 151-154, (1993).
13. B. C. Parker and P. G. Drexel, “A System-Based Sequence of Closed Labs for Computer Systems Organization,” *ACM SIGCSE Bulletin*, vol. 28, pp. 53-57, (1996).
14. P. S. Coe, L. M. Williams, and R. N. Ibbett, “An Interactive Environment for the Teaching of Computer Architecture,” *ACM SIGCSE Bulletin*, vol. 28, 33-35, (1996).
15. B. L. Barnett III, “A Visual Simulator for a Simple Machine and Assembly Language,” *ACM SIGCSE Bulletin*, vol. 27, 233-237, (1995).
16. M. Ben-Ari, Constructivism in Computer Science Education, presented at Twenty Ninth SIGCSE Technical Symposium on Computer Science Education, Atlanta, Georgia, (1998).
17. N. Staggers and A. Forrcio, Mental models: Concepts for human-computer interaction research, *International Journal of Man-Machine Studies*, vol. 38, 587-605, (1993).
18. D. Scott Brandt, “Constructivism: Teaching for Understanding of the Internet,” *Communications of the ACM*, vol. 40, 112-117, (1998).
19. G. W. Scragg, Most computer organisation courses are built upside down, *ACM SIGCSE Bulletin*, vol. 23, 341-346, (1991).
20. A. Clements, *Microprocessor Interfacing and the 68000, Peripherals and Systems*. Essex: Anchor Press, 1989.
21. B. N. PCs, Brand Name PCs, in *Australian Personal Computer*, vol. 3, 122 - 122, (1998)
22. T. W. a. L. B. Results, The Winners and Losers Benchtest Results, in *PC Power Play Hot Hardware Guide*, vol. 2, 75- 75, (1999).
23. S. Mueller, Upgrading & Repairing PC s, QUE, (1997).
24. M. A. Mazidi and J. G. Mazidi, The 80x86 IBM PC & Compatible Computers, *Volumes I & II, Assembly Language Design and Interfacing*. New Jersey: Prentice Hall, 294- 295, (1995).
25. S. P. Maj, T. Fetherston, P. Charlesworth, and G. Robbins, Computer & Network Infrastructure Design, Installation, Maintenance and Management - a proposed new competency based curriculum, presented at *The Proceedings of the Third Australasian Conference on Computer Science Education*, The University of Queensland, Brisbane, Australia, (1998).
26. P. Ramsden, Learning to Teach in Higher Education. London, Routledge, (1992).

PAUL MAJ

Dr S P Maj is a recognized authority in the field of industrial and scientific information systems integration and management. He is the author of a text book, '*The Use of Computers in Laboratory Automation*', which was commissioned by the Royal Society of Chemistry (UK). His first book, *'Language Independent Design Methodology - an introduction'*, was commissioned by the National Computing Center (NCC). Dr S P Maj has organized, chaired and been invited to speak at many international conferences at the highest level. He has also served on many national and international committees and was on the editorial board of two international journals concerned with the advancement of science and technology. As Deputy Chairman and Treasurer of the *Institute of Instrumentation and Control Australia (IICA*) educational sub-committee he was responsible for successfully designing, in less than two years a new, practical degree in *Instrumentation and Control* to meet the needs of the process industries. This is the first degree of its kind in Australia with the first intake in 1996. It should be recognized that this was a major industry driven initiative.

DAVID VEAL

David Veal received his honours degree in Theoretical Physics from the University of York in England. He lecturered in Physics at South Devon college UK for 10 years. He now lives in Westrn Australia where he has taught Computing and Physics at high school level. He is studying for his PhD in Computing Science at ECU in Perth, Western Australia and is investigating competency based techniques in Computing Science as well as the modeling of computers to aid student understanding.