Journal of Computer Science and Applications.
ISSN 2231-1270 Volume 6, Number 1 (2014), pp. 47-66

http://www.irphouse.com

History of Computing in Sri Lanka and the Teaching of Software Engineering and Information Systems

S. Ratnajeevan H. Hoole

Department of Electrical and Computer Engineering,
Michigan State University, East Lansing, MI 48824, USA.
E-mail: hoole@egr.msu.edu., (www.egr.msu.edu/~hoole/)

Abstract

This paper describes the Sri Lankan system of degrees in computing in the context of the new joint ACM/IEEE Computing curricula in the five areas of computer science, computer engineering, information technology, information systems and software engineering. The University Grants Commission regulates the administration and admission of all universities and is guided by its Standing Committee on Information Technology Development. The author, a member of the Commission and Chairman of the Standing Committee during 2003-6, describes the teaching of computing in Sri Lankan universities and explains why software engineering and information systems engineering have been neglected as independent disciplines. The elite nature of the university system with engineering and medicine at the pinnacle with engineering attracting the most mathematically able has worked against a full expression of computing. The focus has been primarily on computer science and computer engineering with reasonable catering to information technology. Software engineering with little numerical content has fallen victim to the Lankan notion that engineers are superior to scientists. Information Systems, with its lack of emphasis on rigorous mathematics has generally been abandoned to foreign degree programmes conducted by companies with no academic credentials as well as to students who enter the universities for the social sciences and combine it with library science or management. SE and IS being too important to be orphaned, University of Peradeniya has rigorously taught them to engineers majoring in computer science who will engage in big software projects. This paper focuses on teaching these through an engineering faculty in a 90-hour course on

CAD Techniques for Electrical Devices. Two parallel courses, Data Structures and Practices in Software Engineering (which includes Senior Project) are used to reinforce additionally the SE/IS experience. Much emphasis is placed on participating "in the development of software to be used in earnest by others" as required by Computing Curricula 2005. In particular, the course fills the gaps felt by engineers designing devices in not having software engineers on their team and by software engineers computing designs who feel the absence of domain specific knowledge. Course methodology and empirical evaluation are also given. Because much of the work, especially in computing courses, is take-home programming work, take-home exams are studied and shown through a survey to be indefensible and therefore grading is tilted towards invigilated tests.

Keywords: Software engineering, information systems, CAD, numerical methods, interactive design, curriculum, computing education, Sri Lanka, evaluation, take-home exams, cheating.

1. Sri Lankan Universities and Elitism

Sri Lankan universities are marked by an elitist mindset, similar to that implied by the use of terms such as "red brick" and "ivy league" elsewhere. Sri Lanka had Asia's first modern occidental university in 1823 when American missionaries founded Batticotta Seminary [1]. (Although their curriculum was as secular as at any university of the time, the title Batticotta College was disallowed by the colonial authorities who reserved the word College for their intended university by the Church of England in Ceylon which college never came to be). Even though

(i) Batticotta graduates were described by the then Colonial Secretary in the words "The knowledge exhibited by the pupils was astonishing and it is no exaggeration to say that in the course of instruction and in the success of the system for communicating it, the collegiate institution of Batticotta is entitled to rank with many an European University" [2] and

(ii) Two early graduates of Batticotta simply sat Madras University's first final exams without attending classes to become its first graduates in 1857, the year of the university's founding [3],

(iii) Sri Lanka's national university system fails to date to recognise Batticotta's seminal contributions and claims that it is the state that originated universities in Sri Lanka in 1870 when a medical college was founded [4,5]. And this notwithstanding that Samuel Fiske Green of Columbia had, also under the America-Ceylon Mission, for 30 years from 1847 been training medical doctors who ably served the British empire in Ceylon and elsewhere [6] and translated 400 pages of ancient Tamil medical texts [7], a feat still not matched by the local universities. This is the beginning of the system's tendency to claim to be the first and best in everything and retards any rational assessment of shortcomings. State-run universities claiming, not always with good reason, to be the first and best is the norm today.

University of Ceylon, the first state university, grew out of University College Colombo (1921) that trained students for London's external exams. It was built in Peradeniya next to Kandy [5]; for Section 4 of The Ceylon University Ordinance No. 186 of 1942 had specified that the university shall be "in or near Kandy." Thus when the university's campuses were broken off under the Universities Act in 1978, the main campus at Peradeniya became University of Peradeniya, the university's legal successor [5].

Table 1: Enrolment in Higher Education for some Countries.

Country/Region	Percentage of 18-22 age group in tertiary education: 1985	Percentage of 18-22 age group in tertiary education: 1995	Percentage of 18-22 age group in programmes leading to a first degree: 1995
Sri Lanka	3.7	5.1	2.22
Malaysia	5.9	10.6	-
India	6.0	6.4	-
People's Republic of China	2.9	5.7	2.0
Japan	27.8	40.3	25.4
United Kingdom	21.7	48.3	24.15
United States	60.2	81.1	38.1
Australia	27.7	71.7	33.7
South Africa	-	15.9	7.3
Nigeria	3.3	4.1	4.1
Brazil	10.3	11.3	11.3
Ecuador	11.3	17.2	12.73

At these universities no tuition is charged (with nominal exceptions). Admission is done nationally from a common high school exam, the G.C.E. Advanced Level (ALs). So students are of very high academic accomplishments and come from the top 2-3% of society. Generally the best students go into the sciences and the others into the arts stream after the G.C.E. Ordinary Level (OLs or Grade 10), making belonging to the arts stream a stigma. In the sciences, bifurcated into the biosciences and the physical sciences after the OLs, the best at the ALs in the biosciences go into medicine, dental science, veterinary science, agriculture and science in that order and, from the physical sciences into engineering and science. Thus university admission is a mark of high status, and relevant to this paper, the faculty to which one is admitted a further marker of status within this elite group.

For these reasons, the idea is entrenched that only those who are selected competitively are entitled to a higher degree. Consequently the concept of a private university (where the rich can earn a degree based on wealth rather than academic merit) is anathema and can lead to violent student protests. These ideas are formalised in the Universities Act [8] which recognises only national universities and places the University Grants Commission (UGC), an independent commission of 7 members appointed by HE The President, in charge of regulating the administration including admissions to and authorisation of academic programmes of Sri Lanka's 15 universities.

It is now recognised, based on World Bank studies, that unless 8% of the 18-22 age cohort of a nation are in first-degree programmes, a country has little chance of reaching NIC status. Table 1 [4] gives figures for some countries including Sri Lanka at 2.22%. As such, government has been doing much to increase Sri Lanka's figure. However, there is little that can be done given the elitist self-image of the university system and the lack of national resources (both material and qualified doctoral level teachers), while any approval of private universities will lead to student riots and even the collapse of the government that attempts reform. Administrators pushed by political authorities to increase enrolment have rightly expanded programmes at great cost to quality, especially the cheaper arts programmes, particularly Pali, Sanskrit and Buddhism. Many high-performing AL students have been placed in these programmes and their increasing dissatisfaction has led them to become the vanguard of many youth uprisings. This further increases the stigma of being an arts student. The problem of the arts faculties is worsened because only 27% of the confirmed academics at arts faculties hold doctorates because of diminished opportunities for scholarships in these subjects in the West.

Yet another aspect of this elitist system are the 3-year general degrees and 4-year special degrees–each degree categorized into first class, second class (upper or lower division) and ordinary pass. While all professional degrees are special degrees, the others, based on first-year (sometimes later) performance, are bifurcated with the vast majority going into general degrees dealing with 3 subjects (leading to school teaching) and an elite going into special degrees where mainly one subject is studied for the remaining years. Then the best of the special degree students are recruited to the staff of the universities and the prestigious administrative service and become the guardians of elitism, ensuring for example that someone who failed to get into the university and went abroad and did well is denied a position on the academic staff because he or she was the second best around age 18.

2. History of Computing in Sri Lanka and *Computing Curricula 2005*

The idea of Computing as encompassing the five formal disciplines of Computer Engineering, Computer Science, Information Systems, Information Technology and Software Engineering, is new and has received formal definitions from the ACM and IEEE only this decade [9]. Much of the development in computing in Sri Lanka in the broad sense of the ACM/IEEE definition happened from the mid-1980s onwards and was not informed by any centralised definition as we now have. An anomaly in Sri Lankan computing arising from the university's stratification is that a computer science graduate from the engineering faculty would have a higher status than one from the science faculty with similar job prospects; indeed, a civil engineer with fewer employment opportunities would have higher social status than a science faculty computer scientist commanding an immediate, high-paying job upon graduation.

Yet another problem flowing from the stratification and specialisation of the system is that the arts students and the bioscience students have little training in mathematics and the algorithms for implementing methods. As such only those in the physical science stream have capabilities beyond OL standards to contribute to computing in an effective way unlike in many other countries. Any meaningful computing therefore is only within the province of the engineering and physical science faculties. Because the best students are usually in the sciences, the only computing programmes to come out of the arts faculties are the less mathematically challenging "Library and Information Sciences" and "Management and Information Science" programmes offered through a faculty of social sciences. Most companies however prefer to recruit computer science graduates for information systems work.

The first computers in Sri Lanka were used principally by accountants from the late 1950s–usually British firms with British machines. The first electronic computer, also British, was at the engineering firm of Walker, Sons and Co. about 1965. The Engineering Faculty at University of Peradeniya was the first academic institution to get (in 1971) a computer, an IBM 1130. The actual first computing classes were by IBM Ceylon under its staff G. Santhiran, Cecil Janze, Tony Kandaiya and Lloyd Perera (with staff from IBM India) catering both to those who had bought their computers from them and to paying students. Three certificate courses were run–Computer System Fundamentals (computer architectures, components, languages, flow diagrams), and Programming Languages and Systems Analysis (program design, flow charts, error routines, print formatting, documentation, testing, Boolean Logic).

University of Colombo, established in 1978 [8], is sited on the temporary premises of the University of Ceylon (1942) while it was built at Peradeniya. These temporary premises were those of University College Colombo that University of Ceylon had absorbed. These premises continued to be of some use even after University of Ceylon moved to its permanent site in 1952. Colombo, formed by breaking off these premises, claims to be "*the* oldest university in Sri Lanka" (www.cmb.ac.lk). Colombo indeed is in possession of the temporary, oldest, premises of University of Ceylon. Its medical faculty is the successor to the Ceylon Medical School of 1870 (which it came to through University of Ceylon). It may therefore be entitled to make such a claim, but carefully qualified and not as *the* oldest–to make this claim to the exclusion of the American institutions, and Peradeniya (the legal successor to University of Ceylon) and the other campuses that also were part of University of Ceylon is untenable. In that same spirit Colombo also claims to have "heralded the dawn of an era of computing within the Sri Lankan Universities [...] as early as in 1967" [10] using the computer at the State Engineering Corporation. Academics of that time confirm the classes in 1967 and the first acquisition of computers around 1979. However, whether they were indeed the first classes (in Sri Lankan universities) bears careful examination because, one, of the culture of claiming to be first and, two, there were other academics trained in the West working with computers, for instance Samarajeewa Karunaratne who had published a textbook on FORTRAN as early as 1968 [11].

The elitism in the university system prevented a computer science programme being established under engineering–why would someone who had performed so well to be admitted to engineering want to be called a scientist? Computing was taught under electrical engineering. The first degree in computing was launched by University of Moratuwa's Engineering Faculty in 1986, working round the status problem by calling the degree B.Sc. Eng. in Computer Science and Engineering [10].

In 1985 University of Colombo launched a Department of Statistics and Computer Science. Because the academic staff members were absorbed from the university's Computer Center, they had been hired as "Instructors" and not under the strict ordinances of the UGC for Lecturers and Professors and, as a result, most came with general degrees and no class. Flush with money from development grants and consultancy opportunities in Colombo, these Instructors were sent by Colombo on postgraduate scholarships by-passing the normally tough postgraduate admission standards in the West and Japan under foreign-aid schemes. They were then promoted to academic positions. Chairs were given without a degree or journal paper in computer science and executive Directors appointed without advertisement in contravention of the Universities Act and the Ordinances promulgated under it [8]. (The majority on the UGC declined challenging this saying that with few doctoral people remaining behind in Sri Lanka, the UGC should not make those remaining behind also run away–raising the issue if the rules apply only to universities that comply). The result has been faculty with many conference papers and often not one journal article and therefore a broad but weak curriculum lacking research insight.

The first Professor of Computer Science at Colombo was a statistician who had no degree or journal paper in computer science and was switched from Professor of Statistics to Professor of Computer Science without being put through the rigorous and mandatory rules for promotion. This would lead to a tradition of nearly all computer science professorships thereafter being filled by those with hardly a journal record and set the discipline back.

The cause of Software Engineering as a discipline was retarded when engineers on the UGC Standing Committee on Information Technology Development (that is appointed by the UGC as its advisory body on the subject) refused to accept "Software Engineering" as a discipline in putting together the scheme of recruitment because scientists would call themselves engineers. The attitude lies in part in the ambivalence of the term as a mere job label sometimes applied to programmers [9]. The argument as to whether software engineers are engineers, however, is closed in Lanka's nonuniversity sector because British engineering credentials are accepted as a result of colonial ties and Britain gives the professional engineer's title "Chartered Engineer" to those who become members of the British Computer Society! So today, even those without university degrees can take BCS membership exams and become chartered engineers. But the battle had already been lost when the Britain's IEE (now called IET), ICE, and other British professional engineering societies did the same and some persons without first degrees obtained postgraduate degrees and rose to be lecturers, professors and even Vice Chancellors. But the essential idea of superiority of those produced by the Sri Lankan degree system continues to be entrenched.

Information Technology as an academic discipline has been retarded by the idea that it is not a science. Boasts thus the University of Colombo, the largest producer of graduates in computing: "IT is the all encompassing practical manifestation of the Computing discipline whereas Computer Science is the hard science that underlies IT" [13]. While academics have this prejudice, the opposite message from the UGC is evident when its Standing Committee on IT Development is placed in charge of all computing education–approving curricula and recruitment and promotion criteria. IT therefore was taught as a three-year general degree but mostly external at University of Colombo. This meant that students would not be admitted through the normal competitive process and the teaching would be done by the Colombo staff for extra pay because it is beyond their normal duties. More seriously it meant that IT without a substantial 4-year special degree would not have academic staff trained with a first degree in IT. University of Moratuwa has since started a Faculty of Information and Communications Technology offering a 3-year IT degree and added, with Colombo, a fourth year to promising students leading to an special degree. With these special graduates coming out and staffing the programmes, IT as a discipline may yield better advocates for itself in Sri Lanka.

3. Software Engineering and Information Systems–Offshore Universities

Out of the five computing degree disciplines identified by the ACM/IEEE Computing Curriculum, we see that Software Engineering and Information Systems have fared the worst in Sri Lanka. Because private universities are not permitted, the government has sought a devious way to introduce private education and fill the gap. Under Margaret Thatcher's reforms, many polytechnics had been made into universities in the UK and asked to prove their worth by raising funds independently. It was the case that some of these new universities had to advertise as late as Summer promising to admit students with three weak passes at the ALs (a grade of E with marks from 35 to 40%). Some of these universities have also offered their programmes in Sri Lanka where locals teach and hold exams to offer degrees that are said to be the same as at the mother campus–which just cannot be, given differences in staff, and student and research cultures. Such offshore universities have been permitted by the government arguing that they are not universities in the Sri Lankan context (for they register as companies) but simply are British, Scandinavian and Australian companies offering their own degrees not Sri Lankan degrees. The reasons for the government (i.e., the regulatory UGC) looking the other way are three-fold: first the backing of these enterprises by influential businessmen; second that they offer important degree courses in scarce areas like computing for which Sri Lankans are willing to pay the high fees because they could hope to recoup their costs through gainful employment; and third that they are better able to impart English language proficiency giving their products good job prospects, unlike in many state universities where the teachers teaching so-called English-medium courses are themselves unable to speak the language and use Sinhalese thereby also causing ethnic issues.

Thus there are now several of these programmes in Software Engineering and Information Systems. By being replications of British and Australian programmes, many of them, certainly the British ones, meet the ACM/IEEE guidelines. There are, however, serious deficiencies as they are often taught by part-timers to cut costs and those with only first degrees and no research accomplishments in the subjects they teach. There have also been calamities as the partner university suddenly wants more money, the local business consortium refuses, students do not know who will issue their degree and a new university partner is found to offer its own degree despite the courses having been taught under another university's curriculum. But these being "private companies", the UGC, avoids involvement. Given the needs, the output of these offshore establishments is insignificant–for instance, the best known of these businesses in its 20 years of existence has produced about 1000 "British" graduates.

4. Software Engineering and Information Systems at University of Peradeniya

For the reasons described above, software engineering and information systems, as computing degrees, are not taught at the major science and engineering faculties. At University of Peradeniya, in the Faculty of Engineering, the department had been titled Computer Sciences in the plural and run service courses. When a B.Sc. Eng. degree with specialisation in computing was mooted in the year 2000 [14], software engineering and information systems had to be taught as subjects rather than degree programmes. While the word engineering in Software Engineering would have suited the culture, the course could not be titled Software Engineering because, as affirmed in the Computing Curricula 2005 [9], it has little emphasis on numerical methods as befits the nature of the education system in filtering the most mathematically talented in the nation for the engineering programme. Titling the degree B.Sc. Eng. Computer Science (later changed to Computer Engineering because of student wishes) and recognizing it as an engineering degree where training in formally correct software development is of essence, it was decided to incorporate software engineering in a big way with some information systems.

Given the tightness of the curriculum, these disciplines would be taught as parts of three year-long courses–CAD Techniques for Electrical Devices, Practices in Software Engineering and Data Structures. Having taught finite element field computation for several years; having developed and maintained a formal CAD program for electrical devices over three years for PA Consulting Services/PA Technology [15]; and having maintained the field offices of MAGSOFT Corporation in Southern California at Harvey Mudd College; it was decided to transfer that experience to the students through the course CAD Techniques for Electrical Devices as the primary vehicle for SE/IS. The specific skills imparted in the courses are seen in Table 2 and are seen to tally well with the subsequent Computing Curricula 2005 [9].

Table 2: The Three Courses: Vehicles for Software Engineering and Information Systems.

Software Engineering and Information Systems topics through three Courses		
Course I: CAD Techniques for Electrical Devices	Course II: Data Structures	Course III: Practices in Software Engineering
Software Design/Development Data Bases Integrative Programming Human Computer Interaction Graphics/Visualisation Scientific Computing Analysis of Technical Requirements Configuration Management Documentation	Databases Information Management Information Systems Development E-Business Software Development	Software Design/Development Project Management Software Consulting Practice Communications Legal, Ethics, Society E-Business Software Quality, Risk analysis Distributed Systems Analysis of Technical Requirements

The subject Data Structures, the title being a holdover from the past, covered Databases and also involved much software development. Practices in Software Engineering had to cover many things in a course system allowing a limited number of courses. This included the Senior Design Project which always had software development projects with requirements elicitation, software design and project management [16]. For engineers from a branch likely to lead to some consulting, the author's experience at PA Consulting Services was taught in matters like running a consulting service and how to quote for software projects and handle budgets. Many concepts of Table 2 such as the use of Gantt charts and PERT charts, project management, communications, software quality, risk analysis, requirements elicitation and software design [16] were inculcated through the Senior Project and simultaneous lectures and student presentations. Projects were from various topics and always involved extensive software development. While not all projects involved e-business, games development, risk analysis, etc., the regular student presentations of the projects that did, ensured that all students became aware of these topics and learnt from each other and honed in their writing and communications skills. In fact with student permission the best and most flawed reports were discussed publicly in a lecture and that was possible among a group of students who were socially tightly knit.

5. CAD Techniques for Electrical Devices

The above course was conceived in 1984 for electrical and computer engineering students in the US as a three-quarter graduate sequence at Drexel University with emphasis on engineering electromagnetics and the finite element method [17]. The same course was taught at Harvey Mudd College from 1987 to 1995 as a 2-semester Senior Elective/Master's course. It emphasised the development of field computation software with preprocessing, solution and postprocessing algorithms and a book was developed for the course [18]. Students were put through the theory and hands on work for developing modules and were expected to have a working finite element program at the end of the course. All along the absence of software engineering as a requirement for producing correct software was felt.

The creation of software for engineering design purposes at once involves engineering analysis, interactive computer techniques, databases and software engineering. To train personnel for such a software development exercise at a university with the necessary academic content, skills in both mathematical analysis and computer programming are required. However there are limits as electrical engineering students are usually more versatile with numerical methods and the physics of devices rather than software design and development. While indeed several electrical students are versatile programmers, they often lack formal training in the software engineering life-cycle. At the same time, working as a developer of commercial CAD programs [15], this writer has observed stakeholders in the CAD software development industry more often to be computer science graduates who are generally unable to handle the deep specialised mathematics and physics required to be suave CAD-package developers. This is confirmed in [9] in its Table 3.1 where skill-sets for IS, IT and SE personnel are given to be absolutely zero and possibly as low as zero for CS personnel! So there is a need for scientists/engineers who are well trained in both numerical engineering methods and software development skills. Indeed, Issler *et al.* [19] have identified that "[d]ue to the non-commercial, research-oriented context, software in medical informatics research projects is often developed by researchers as a proof-of-concept without applying structured software development process models." Hence the need for software engineers and electrical engineers working together.

Placed in charge of developing a computer science programme out of an electrical engineering department, several new courses had to be offered using the limited teaching staff who were tilted towards electrical engineering [14]. Attracting electrical students to be computer scientists in the early boom years for computer scientists in the last decade, the opportunity arose to emphasise software engineering in a country where software engineering was not a degree programme and to offer a course that would train computer scientists in the mathematics and physics of engineering design and engineers in the methodology of software development. Be it noted that these plans, in the event, are within the scope/range of expected skill-sets identified by Computing Curricula 2005 in it Table 3.1 [9].

6. Course Methodology

As a now given requirement that students participate "in the development of software to be used in earnest by others" [9], the three courses were centred round large scale software development. Some of the key laboratory exercises in the course CAD Techniques [18] included developing 1) The basic Finite Element algorithms (using first order triangle data from a Poisson problem, forming the local matrices for the elements, adding them to form the global equation and then solving the global equation), 2) Mesh Generation (starting the crude triangular tessellation of the device geometry and refinement thereafter using the Delaunay criterion) 3) Matrix Solvers of various types (Matrix Inversion, Cholesky Factorisation, Successive Over Relaxation, Gaussian Triangulation, and Conjugate Gradients with Cholesky Preconditioning), 4) Sparse Matrices (Sparse Storage and Profile Storage) and Renumbering 5) Plotting Equipotential Lines from graphics primitives and 6) Interactive graphics with considerations of man-machine interaction and user-friendliness (drawing a device on the screen, and the user pointing to a part of the device and having returned the field quantity there).

Not only was code development an integral part of the course but also research (here for algorithms) which Artail suggests is a must for a course on software engineering [20]. The concept of appropriateness of models and databases was gone into in the introductory lectures. Cebollero, Llamas and Dodero [21] explain why software must take into account the range of abilities, needs, and interests of different stakeholders, adapting itself to a variety of contexts and pedagogical paradigms. In this case therefore the student developer must first consider for whom the work is intended–for professional engineers and at what level or personal use or research or simply homework or, as in many cases in the projects, for students who would use it for learning a subject and at what level–and therefore see software development as "a complex process that must consider diverse perspectives and levels of abstraction" [21]. With an ambitious effort like this, the work would be heavy leading to each student having a package at the end. In the event, this has been affirmed as a sound policy by Rombach *et al.* [22] since it simulates the real world of work. As suggested by Petrenko *et al.* [23], open sources were allowed to be used, for example for mesh generation in this course after teaching the theory in class, as a means of amelioration of load. This use of open sources also naturally led to software engineering concepts of reuse, patterns and interfaces [14] that the students were forced into as a result.

To simulate the real world where the client is used for requirements elicitation (and clarification during analysis) each team was paired with another team as a client besides the author. To encourage student centred learning, students also participated in evaluating as suggested by Casallas and Lopez [24] and commenting on the user interfaces. As each module was developed and added to the main spine of the program, and submitted the then current version for grading, student-determined version numbers and their discussion taught the different aspects of configuration management [16].

7. Aspects of Information Systems

As mentioned previously the other two courses, “Practices in Software Engineering” and “Data Structures” were the main vehicles for teaching information systems in the computer sciences degree programme. Many of the Senior Projects that were incorporated involved the development of databases (for example a project for the faculty’s library developing online access and reservation of books) and development of a toolkit for games where modules were developed as software components. Another project led to a program with different modules for teaching electromagnetics–the database consisting of different solved problems–and was reported in the IASTED 2005 conference proceedings [25] and subsequently published in a journal [26]. Another undergraduate project involved an online examination system with a databank of questions and answers addressing issues of firewalls and timed exams over a slow connection so that it could be used for real exams (as well as for self-examination by students while studying). One of the better undergraduate SE/IS projects, it also led to a publication [27].

Further, because of input from the Department’s informal Board of Industry Advisors, a module on the latest commercial packages used in business was developed under Practices in Software Engineering with a view to facilitating the employment of graduates. Some of the advisors donated their own packages. Since an academic programme could not be involved with commercial packages and yet student interests demanded that they be given skills desired by industry, it was incorporated as lab work under Practices in SE and each “Instructor” (like a TA in the US but full-time) had to learn one package and then give lab exercises to the students. Thus we had Oracle^{TM}, donated by an advisor whose company was the agent, something for which the market price would have been impossible in the third world context. The incorporation of Information Systems was also effected in the CAD Techniques course in modelling various materials using different mathematical models [18] and incorporation in a database to be used by the programs. Different drawings of electric machines tagged by number of poles, slots etc .were also conceptualised in a database [28].

8. Evaluation of Students–the Validity and Political Correctness of Take Home Exams

In the way the three courses have been designed and indeed in most CS courses, much of the learning is left to coursework as programming and software development. How is this take-home work to be evaluated? Casallas and Lopez [24] have suggested evaluation by students. But then grades make careers. So is student grading fair? Are take-home exams a proper measure of student performance?

Generally in the western world–particularly America where almost 40% of the population earns college degrees (Table 1) and cannot be subjected to rigorous testing–

multiple forms of assessment are a given. These often include a) attendance (meaning marks for trying and even dissembling interest thereby punishing good students who out of preservation of self-dignity display no affectation; or who do not come to class because they may perceive no benefit from the oratory of the "Sage on the Stage") and b) participation (meaning professor's discretionary marks) making students talk often with nothing to contribute. Justification is on the grounds that there are different learning styles and take-home exams are therefore common. In fact students prefer take home exams–and that in itself is telling. Removal of the pressures of timed exams is also given as a reason for take home exams. It is simply considered politically incorrect to even suggest that students might be copying–for the very discussion would invalidate the grades we give. It is also considered politically the correct thing to not be suspicious of students. But several trials by this writer have shown that take-home exams yield higher class averages and diminish the differences between good and bad students. Take-home exams are therefore used by this writer only when the class cannot cope and need a take-home exam as an uplift to avoid problems with the administration which would call for an explanation when too many students fail.

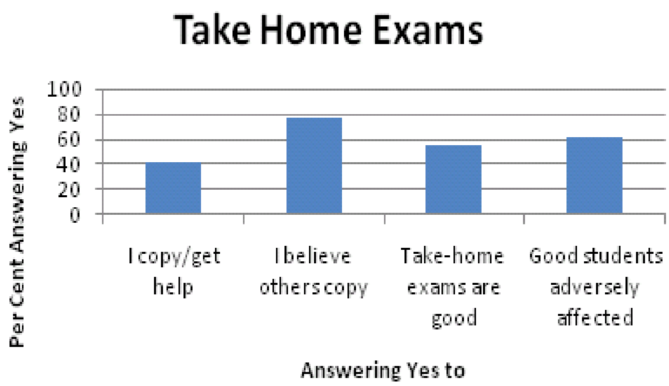

Fig. 1: Survey Responses on Take-Home Exams (Summary of 114 Responses)

As a result of political correctness we have only occasional papers such as by Underwood and Szabo [29] making findings that males cheat more freely than females–a difficult claim to make nowadays even when facts warrant such conclusions. Underwood and Szabo would have been in great trouble if their results had shown that women copy more than men. Informal observations by this writer both as a teacher and a postgraduate student that peoples from more tightly knit cultures are more likely to copy, are even more difficult and explosive to make. Apologists will attribute the better performance at take-home exams to the absence of exam stress. Especially worrisome and more difficult to explain is a higher performance by 7% scored in invigilated examinations by the half of a software engineering postgraduate class (21 students) that did face-to-face learning than the other half that consisted of 26 distance students in the same course – a difference that vanished in homework and take-home exams [30]. This was at Rensselaer Polytechnic Institute in the author's class. By and large papers gloss over such glaring discrepancies and encourage peer review among students [24]. Pare and Joordens [31], while conceding differences when students grade, speak of a software-based “grade accountability feature,” to offset the differences.

Sri Lanka too has been pushed towards considering take-home components of work for the final grade by bodies such as the World Bank and the Asian Development Bank through their experts who have a big say in the approval of loans for universities and the Ministry of Education. An early victim of this was the GCE O. Level Sewing paper where a good part of the marks came from a take-home assignment. When this writer’s daughter struggled and handed in her work at a leading high school, her teacher asked this writer privately to note that all other students got their tailor to do their stitching, and that his daughter too should do the same to do well. The advice was not heeded and she failed to get an A in a subject in which most get an A.

In fact, for the SAT examination which is a major determinant in US university admission, school teachers advised another daughter’s US high school class to carry formulae on their programmable calculators. At the test, although hand-phones were disallowed, invigilators looked the other way as students texted. Her survey of her friends showed that nearly all had their parents preparing their university applications and writing the essays that are very admissions determining. The result leaves little doubt that copying is done, particularly when it is career determining. As a parent this writer has helped his daughter with take-home exams because the teachers seemed to expect that; while asking his daughter not to carry formulae on her calculator because it seemed not to be expected by the College Board which conducts SATs. By using essays (probably by parents) and online applications, universities might be unofficially targeting upper-middle class parents whose children are more likely to succeed.

So as to draw conclusions from facts rather than informal observations, this writer conducted a blind, web-based survey among college-going or college-educated people using class lists from North America, South Asia, Europe and East Asia. Based on the 174 responses available as summarised in Fig. 1, survey reports a) 41.5% said that they copy at take-home exams b) 77.6% that they believe that others copy and interestingly c) Although 63.6 percent agreed that good students are adversely affected by take-home exams, 56.7% still thought they were a good thing. Thirty additional persons were given the survey by his students to their close friends by hand and collected them. 83.3% admitted to cheating showing that in a blind survey the full extent of cheating is not brought out.

Given these findings, there really is no basis for using take-home exams to grade some students as superior to others. It would appear that the proliferation of universities, has led to a system where a) Students must be admitted for the university to survive; b) Thereupon admitted students must be passed out for the university to have admissions applicants for the next batch–note that the student attrition rate is widely announced and is used to measure a university’s “quality”; c) A university administration run by businessmen insists that professors who fail their students do not get tenured and d) finally, professors who must pass their students to receive tenure and earn their annual raises devise take-home exams despite much evidence that students cheat at such exams.

These findings affirm the practice advocated where the homework is graded awarding almost full marks so long as the job was done. With this object students were used to evaluate software outcomes for purposes of edification as suggested by Casallas and Lopez [24]. Where a student's submissions are incomplete, the grade is "Resubmit" as a learning exercise till full marks are obtained. As a result the letter grade was in reality determined by the exams which were designed so that those who really did the software development scored well. This is because the integrity of grades must be a given and not left to trust. In any event the Sri Lankan system will not allow take-home exams and any attempt would bring disrespect to the course in the public's eye–although homework is used in part for grades on World Bank and ADB recommendations, that requirement has been by passed by making marks differentials be based only on the supervised exams. Good students have a right to the demonstrable integrity of their grades.

9. Empirical Evaluation of Courses

The curriculum for the computer science degree was carefully planned to meet ABET and Washington Accord guidelines. In developing the particular course with a class of about 30 students with different skills, the system of course moderation was used to make improvements and be in line with international standards. An electrical engineering professor from Rensselaer Polytechnic Institute in the US marketing his own software products for electrical devices was used as a moderator. Computer science professors from University of Birmingham and Rhodes University were also asked to critique the syllabi and the examination papers. Their suggestions were fully incorporated. These international moderators also looked through the graded final exams and made their input. The most useful outcome was in settling whether software development homework fell within the ambit of the faculty-required accompanying laboratory classes in engineering. An affirmative answer from the moderators settled the raging debate within the faculty unfamiliar with standards in computer science. Another useful outcome was the incorporation of the software engineering life-cycle [16] into the course syllabus. As mentioned, industry leaders who were consulted by the department also helped improve the courses and the programme while international friends of the faculty were surveyed by the Faculty for an overall assessment of the computer science programme with very positive findings [14].

At the end of the course, each student had his or her own finite element program with extensive capabilities. Perhaps the best proof of the course in the period where this writer was in charge (2000 to 2002) is that many of the undergraduate projects led to ISI-indexed journal articles [26-28, 32-34]. Another metric is that over two-thirds of the first two batches have gone through western graduate programmes–perhaps nearly all, but getting the exact number is difficult given the scattering overseas.

10. Is it a Replicable Course? Experience at other Universities

The course was very ambitious. The students were the crème de la crème of Sri Lanka, having been selected to the faculty from the very competitive national high school examination, the G.C.E. Advance Level, where about 2000 high school students are selected from the 150,000 who sit the examination for the most competitive seats in medicine and engineering. And these 150,000 are already weaned from the national G.C.E. Ordinary Level Examination. It was therefore possible to emphasise a disciplined, hard-working approach to the course as suggested by Romabach *et al.* . . . . [22] to software engineering so as to replicate the real situation students would face when they leave the ivory tower. With such a motivated and highly intelligent class, the teaching experience was a pleasure. The students did a lot of research on their own leading to several publications from undergraduate research [25-28, 32-34] and added features that were not taught in class to their programs.

The question then arises whether the course experience is worth sharing through a paper with other universities where the students may come from a less selective process. The answer is a firm yes, but with some qualifications. In the period 1984-87, this writer set up and taught a very similar course at Drexel University, but for graduate electrical students, titled "Computer Aided Design of Electromagnetic Products." More than a half of the enrolment of 12-16 students was constituted by this writer's graduate thesis students. They were committed to the topic in a way undergraduates are not and with the instructor's students leading the way the course was a success as judged by this writer and the continued enrolment by students of other professors. In the period 1987-1995, a similar course to the one at Drexel was set up as a senior elective/first year master's course at Harvey Mudd College, one of America's most selective institutions. The course was hugely successful, with mostly undergraduates.

The course was offered again at Drexel University in 2007/8 as a graduate course while this writer was a Visiting Professor there with no graduate students. The course was a disaster in that if the students fall behind (as they did) say in setting up the finite element code, there is no way in which the class can proceed to the next stage of say equipotential plotting; for there is nothing to plot. A part of the reason was that students were taking the course not because it was in their area of research but because they needed certain credits for their degree and they happened to pick this. In the US system today, unlike in the 1980s and before, student centered teaching is the norm where the customer is right. Thus if students fail, it is regarded as the instructor's fault. For example, when an assignment was made to senior electrical engineers requiring programming, based on a student complaint, the Provost queried why the assignment was made when the previous year no computer assignments had been made by the then professor. Similarly late homework has had to be accepted on the Dean's orders. Thus, given the culture, in the course CAD Techniques, when students did not turn in their homework, there was no question of carrying on regardless. The course therefore had to slow down to a pace the students determined by handing in homework and only a few of the learning outcomes were realised. Students know that most of them have to be passed in the present culture.

The course therefore will work and be a very positive force in student careers where they are invested in the course and their programme of study with appropriate intellectual endowment.

11. Conclusion

Since the Sri Lankan system will not support SE and IS as separate programmes because of the elitist nature of Sri Lankan education explained in the paper, they have had to be incorporated through a regular computer science programme using three course as vehicles. One of the courses, CAD Techniques, as developed and delivered over a career of 30 years and adapted to teach software engineering, is ambitious and useful and indeed successfully implementable with highly accomplished undergraduates as has been demonstrated at University of Peradeniya. For accomplished students the three courses have led to seminal publications by undergraduates. The model used is sound for training computer science personnel with a good balance between software engineering and domain specific knowledge. It is also a useful and successful exercise when the students are highly motivated as at Drexel University where thesis students were involved at graduate level and at Harvey Mudd where undergraduate students are highly accomplished. However, when the class is neither motivated nor particularly intelligent, the course just takes us nowhere in an academic culture pandering to students.

The course has been shown to be in line with the ACM/IEEE model for software engineering that was developed after the event insofar as the emphasis on learning outcomes is as suggested by ACM/IEEE. An exception is the numerical methods which have been incorporated to include the domain-specific knowledge required for development. Moreover, the course has been shown to be in line with the suggestions of many SE/IS educators in terms of emphasizing difficult software development work, research, use of open source software, and student evaluations (limited to coursework). Grading has been tilted towards supervised exams because take-home exams having been found through a survey to involve copying and to work against good students by diminishing their superior accomplishments. A future paper will take up the survey’s finding that Asians have less faith in take-home exams–a matter of importance in programmes where Asians are greatly present.

References

[1] American Board of Commissioners for Foreign Missions, *Report of the Deputation to the India Missions made to the American Board of Commissioners for Foreign Missions at a Special Meeting Held in Albany, NY, March 4, 1856*, Press of T.R. Marvin, Boston, 1856.

[2] Sir James Emerson Tennent, *Christianity in Ceylon*, John Murray, London, 1850.

[3] S.R.H. Hoole and D. Hoole, "Theocracy, History and Historiography: an Exploration in Professional Ethics," *Current Science* Vol. 85, No. 12, pp. 1681-1684, 25 Dec., 2003. Also *The Hindu*, Aug. 09, 2004. www.hinduonnet.com/thehindu/mp/2004/08/09/stories/2004080900190300.htm. Downloaded July 20,2009.

[4] University Grants Commission, *National Plan for the University Sector in Sri Lanka*, UGC, Colombo, April 2004. Also see http://www.sjp.ac.lk/aboutus/history.html Downloaded June 20, 2009. Downloaded July 20, 2009.

[5] H. Abeyagunawardene, "University o f Peradeniya – More open than Usual, *Daily News*, Dec. 23, 2002.

[6] Samuel Fisk Green, *Life and Letters of Samuel Fisk Green, M.D., of Green Hill*, New York, 1891.

[7] US Govt.: www.slembassyusa.org/srilanka_us_relations/historical_context.html. Downloaded June 19, 2009.

[8] Sri Lanka Parliament, "The Universities Act – No. 16 of 1978 and amendments thereto." Also see http://www.ugc.ac.lk/policy/university%20act/preface.html

[9] The Joint Task Force for Computing Curricula 2005, *Computing Curricula 2005: The Overview Report*, ACM/AIS/IEEE-CS, 30 Sept. 2005.

[10] http://www.cmb.ac.lk/academic/Science/Departments/Statistics/history/. Downloaded June 20, 2009.

[11] Samarajeewa Karunaratne, *Fortran Programming*, Frewin, Colombo, 1968.

[12] http://www.cse.mrt.ac.lk/index.php?option=com_content&task=view&id=53&Itemid=139. Downloaded June 20, 2009.

[13] www.ucsc.cmb.ac.lk/wiki/images/1/1d/Article_dnr.doc. downloaded June 20, 2009.

[14] S.R.H. Hoole, "Establishing a Computer Science Program with Staff Shortages in a Developing Country through an Electrical Engineering Department," *Int. J. Eng. Educ.*, Accepted.

[15] P. Campbell, S.R.H. Hoole and I. Tsals, "Finite Element Analysis in 2-D and 3-D on a Personal Computer", *IEEE Trans. Magn.*, Vol. MAG-20, NO. 5, pp. 1903-1905, Sept. 1984.

[16] Bernd Bruegge and and H. Dutoit, *Object-oriented Software Engineering – Using UML, Patterns and Java*, 2nd edn., Pearson-Pentice Hall, Upper Saddle River, NJ, 2004.

[17] S. Ratnajeevan H. Hoole and P. Ratnamahilan P. Hoole, "Finite Element Programs for Teaching Electromagnetics", *IEEE Trans. Educ.*, Vol. E-29, No. 1, pp. 21-26, Feb. 1986

[18] S.R.H. Hoole, *Computer Aided Analysis and Design of Electromagnetic Devices*, Elsevier, New York, 1989.

[19] L. Issler, C. Sprekelson and C. Wessel, "Implementing software development guidelines in a medical informatics research project," *Methods of Information in Medicine*, Vol. 46, No. 6, pp. 641-645, 2007.

[20] H. Artail, "A methodology for combining development and research in teaching undergraduate software engineering," *Int. J. Eng. Educ.*, vol. 24, No. 3, pp. 567-580, 2008.

[21] D.D. Cebollero, C.F. Llamas and J.M. Dodero, "Towards an effective instructional engineering analysis method," in W. Nejdl and K. Tochtermann (ed.), *Innovative Approaches for Learning and knowledge Sharing*, Lecture Notes in Computer Science, Vol. 4227, 2006.

[22] D. Rombach, J. Munch, A. Ocampo, W.S. Humphrey and D. Burton, "Teaching disciplined software development," *J. of Syst. and Software*, Vol. 81, No. 5, pp. 747-763, 2008.

[23] M. Petrenko, D. Poshyvanyk, V. Rajlich and J. Buchta, "Teaching software evolution in open source," *Computer*, Vol. 40, No. 11, pp. 25-, Nov. 2007

[24] 24. R. Casallas and N. Lopez, "An environment to help develop professional software engineering skills for undergraduate students," *Int. J. Eng. Educ.*, Vol. 24, No. 4, pp. 648-658, 2008.

[25] S.R.H.Hoole, S. Aravinthan, T. Kuganeswaran, R. Vasanthakumar, and T. Gnanakumar "Data-Band of Electromagnetic Still and Animated Graphics for Teaching over the Internet" Proc.*IASTED Education and Technology Conference. IASTED*, Canada, July 2005

[26] S.R.H. Hoole, S. Aravinthan and T. Kuganeswaran, "CAI of Electromagnetics with Flicker-Free Time Varying Fields Powerfully used to Enhance Learning, *IEEE Trans. Magn.*, Vol. 45, March 2009.

[27] Dushyanthi Hoole, N. Yogendran, S. Thavachandran, P. Pryatharshan and S.R.H. Hoole, "A Bank of Chemistry Questions on an On-line Server," *Journal of Science Educ. & Techn.* Vol. 11, No. 1, pp. 9-13, 2002.

[28] S.R.H. Hoole and A. Mascrenghe, K. Navukkarasu, and K. Sivasubramaniam, "An Expert Design Environment for Electrical Devices and its Engineering Assistant," *IEEE Trans. Magn.* Vol. 39, No. 3, pp. 1693 –1696, May 2003.

[29] J. Underwood and A. Szabo, "Academic offences and e-learning: individual propensities in cheating," *British J. of Educational Technology*, Vol. 34, No. 4, pp. 467-477, Sept. 2003.

[30] S.R.H. Hoole, "Distance Education using RPI's Learning Management System and Model of Lesson Delivery," *Int. J. Elect. Eng. Educ.*, Under review. Manuscript available from author.

[31] D.E. Pare and S. Joordens, "Peering into large lectures: examining peer and expert mark agreement using peerScholar, an online peer assessment tool," *J. of Comp. Assisted Learning*, Vol. 24, No. 6, pp. 526-540, 2008.

[32] S.R.H. Hoole, A. Rahunanthan, T. Sivapriya and S. Sutharsan, "MacLean's Model of Flux Penetration: Addressing Stability," *IEEE Trans. Magn.* Vol. 39, No. 3, pp. 1187 –1190, May 2003

[33] M.R. Udawalpola, K.R.C. Wijesinghe and S.R.H. Hoole, "Features and Classes for an Optimisation Tool for Electrical Device Designs and Computer-Aided Instruction," *J. of Mat. Proc. Techn*, Vol. 161, pp. 327-333, 2005.

[34] K.R.C. Wijesinghe, M.R. Udawalpola and S.R.H. Hoole, "Towards Object Oriented Finite Element Preprocessors Exploiting Modern Computer Technology," *J. of Mat. Proc. Techn*, Vol. 161, pp. 247-252, 2005.

About the Author

S. Ratnajeevan H. Hoole, B.Sc. Eng. Hons Cey., M.Sc. with Mark of Distinction London, Ph.D. Carnegie Mellon, is Professor of Electrical and Computer Engineering at Michigan State University in the US. For his accomplishments in electromagnetic product synthesis the University of London awarded him its higher doctorate, the D.Sc. (Eng.) degree, in 1993, and the IEEE elevated him to the grade of Fellow in 1995. Prof. Hoole has been Vice Chancellor of University of Jaffna in Sri Lanka, and as Member of the University Grants Commission there, was responsible with six others for the regulation of the administration of all 15 Sri Lankan universities and their academic standards, admissions, funding and regulation of administration.

Prof. Hoole has contributed widely to the learned literature on Tamil studies and been a regular columnist in newspapers. He has been trained in Human Rights Research and Teaching at The René Cassin International Institute of Human Rights, Strasbourg, France, and has pioneered teaching human rights in the engineering curriculum. In the area of education research he has worked on flipped teaching to condense optimization and finite elements into one course and advanced teaching ethics for engineers viewing the IEEE Code as an ideal rather than an inviolable deontological standard that engineers always fall short of.

In his continuing technical research work for the US Army's Tank Automotive Research, Development and Engineering Center (TARDEC), Prof. Hoole has been applying his design optimization methodology to identifying defects such as corrosion and post-IED damage in the armor of ground vehicles through Michigan State University's Nondestructive Evaluation Laboratory. The system he is building inverts eddy current testing data on a graphics processing unit for quick characterization of defects in the theatre.