Are Computer Engineering Students Getting Enough Design?

Dennis K. Peters
Electrical and Computer Engineering
Faculty of Engineering and Applied Science
Memorial University of Newfoundland
St. John's, Newfoundland
Canada
dpeters@engr.mun.ca
http://www.engr.mun.ca/~dpeters/

June 1, 2001

Abstract

In this paper I will discuss several aspects relating to the Engineering Design component of Computer Engineering programmes, with particular reference to the structure of our programme at Memorial. I will consider both the quantity of design instruction, with respect to both CEAB accreditation criteria and our perception of the needs of our graduates, and the distribution of it through the programme, both by year and technical area.

1 Introduction

Like all Canadian engineering faculties, we at Memorial regularly review the make-up of our undergraduate programmes for many reasons, the most obvious of which is to ensure that they satisfy the accreditation criteria stipulated by the Canadian Council of Professional Engineers (CCPE)[1]. One of these criteria that can cause some concern is the requirement that the programme contain a minimum of 225 Accreditation Units (AUs) of "Engineering Design," which is defined as follows.

> Engineering design integrates mathematics, basic sciences, engineering sciences and complementary studies in developing elements, systems and processes to meet specific needs. It is a creative, iterative and often open-ended process subject to constraints which may be governed by standards or legislation to

> varying degrees depending upon the discipline. These constraints may relate to economic, health, safety, environmental, social or other pertinent factors.[1]

One reason that engineering design content can be of some concern is that it is not always obvious exactly what portion of a particular course *is* design — while some courses, for example those based around senior design projects, are clearly entirely, or almost entirely, design, others are not so obvious. For example, many programmes, ours included, have early courses with titles such as "Engineering Design," which from their title seem to be substantially about design, but, examined more closely, often include many non-design topics (e.g., graphics, measurement, effective communication), and hence cannot be reasonably claimed as entirely design. The amount of design in technical courses may also be difficult to judge, and may vary significantly depending on how the course material is presented — where one instructor may simply present the technical content, another may lead the students in solving design problems using it, and hence could reasonably claim substantially more design content in the course.

In this paper I will pose several questions regarding the Engineering Design content of Computer Engineering programmes, and offer some of my own insight gained from a recent review of our programme at Memorial. Some of the questions to be considered are:

- How much design can, or should, be taught in junior level courses?
- Should we trade off analysis in favour of design?
- Is programming design?
- Do labs $\rightarrow$ design or design $\rightarrow$ labs?
- Is "building it" necessary?
- How do we assess the amount of design in a course?

The next section presents a brief overview of the Computer Engineering programme at Memorial and a summary of the distribution, both chronologically and by technical area, of design instruction in the programme. The following section addresses each of the above questions, in turn. Section 4 draws some conclusions.

2 Computer Engineering at Memorial

We have recently created a new programme in Computer Engineering based on our former "Computing and Communications" option in Electrical Engineering. The first class of graduates from this programme will graduate in 2002, and so we are planning for an accreditation visit that year. The following information is extracted from our analysis in preparation for that visit.

Table 1: Core Programme Chart

A	B	1	2
Calculus I	Calculus II	Eng. Math I	Eng. Math II
Chemistry I	Chemistry II	C.S.	Eng. Materials I
Physics I	Physics II	Mechanics I (statics)	Mechanics II (dynamics)
C.S.[2]	C.S.	Circuits	Structured Programming
English	Free Elective	Eng. Graphics	Eng. Design

Table 2: Programme Chart for Computer Engineering

3	4	5	6	7	8
Probability & Statistics	Economics	C.S.	Assessment of Technology	T.E.	T.E.
Circuits II	Elec. Devices	Analog Elec.	T.E.[3]		
Discrete Math	Numerical Methods	Digital Systems	Project	Operating Sys.	Project
Adv. Programming	Data Structures	Algorithms	Software Design	Software Eng.	T.E.
Digital Logic	Microproc.	Comp. Architecture	Controls I	T.E.	T.E.
	Systems & Signals I	Systems & Signals II	Comm. Principles	Voice & Data Comm.	Digital Comm.

All engineering students at Memorial follow a common "core" programme for the first two years[1] of their study as outlined in Table 1. Since this portion of the programme forms the foundation for further study, it naturally emphasizes mathematics and the basic sciences, with a limited amount of instruction in engineering sciences and design. This is discussed further in Section 3.1.

The non-core portion of the Computer Engineering programme is summarized in Table 2. Terms 3 and 4 of the programme are taken in common with students in the Electrical Engineering programme. The program is structured to ensure that the essential content is covered early so that the senior year can be used for specialization based on student interest. Hence there is very little choice prior to term 7, and a fair amount of freedom in terms 7 and 8.

In the project course in term 6 the students work in teams consisting of both Computer

[1]For primarily historical reasons, Memorial has an introductory "first year", during which the students take courses that are mostly not specific to engineering, we refer to this portion as terms A and B. Some particularly well prepared students are admitted directly into term 1 through the "Fast Track" option.

[2]Complementary Studies

[3]Technical Elective

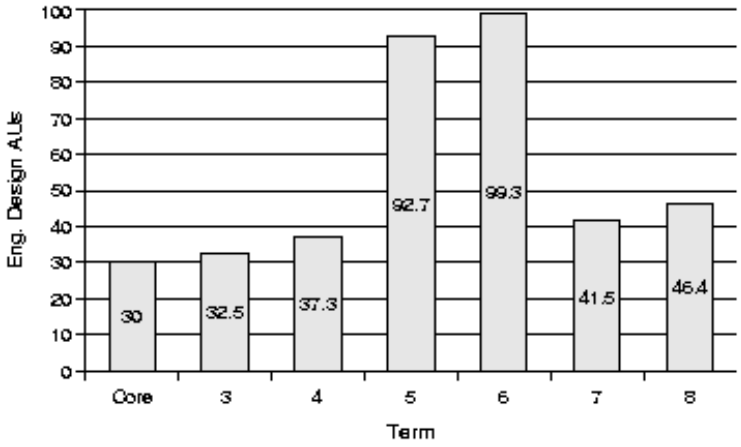

Figure 1: Design AUs by Term

and Electrical engineering students to design and construct (using standard components) a system satisfying given requirements and optimized for given criteria. The relative success of the systems is judged by a competition between teams. The systems typically involve software, digital and analog components. For example, in recent offerings the project involved adding 'intelligence' to a toy motorized front-end loader so that it could autonomously navigate through a simulated underground mine and could respond to high-level commands from a remote computer via wireless link (see `http://www.engr.mun.ca/~dpeters/6806/`).

In term 7 the "Software Engineering" course is also based around a team design project in which the teams each design and implement a software system to satisfy the given criteria. Here, too, the success is judged partially by a competition between the teams (e.g., see `http://www.engr.mun.ca/~theo/Courses/se/index.html`).

The "capstone" design project in term 8 is an individual design project in which each student designs a different system to satisfy some need expressed by a client — typically a professor or industry representative. At the end of the term the students demonstrate their projects for their fellow students, professors and industry representatives. The demonstrations are judged by a team of academic and industrial judges.

2.1 Distribution of Design Instruction

The distribution of design instruction by academic term (with all the core considered together) is illustrated in Figure 1. To simplify the analysis, technical electives — electives where students choose from a small number of specialized engineering courses — are neglected in this figure. However, since half of the courses taken in the senior year are technical electives, the quantities given for these terms are somewhat low.

Figure 1 clearly shows a pattern: early courses emphasize non-design topics (e.g., engineering sciences, math), with a small amount of design — typically less than 25%. This increases through the programme to peak in terms 5 and 6 (third year of a four year programme) where most courses have significant design components, and then falls off again

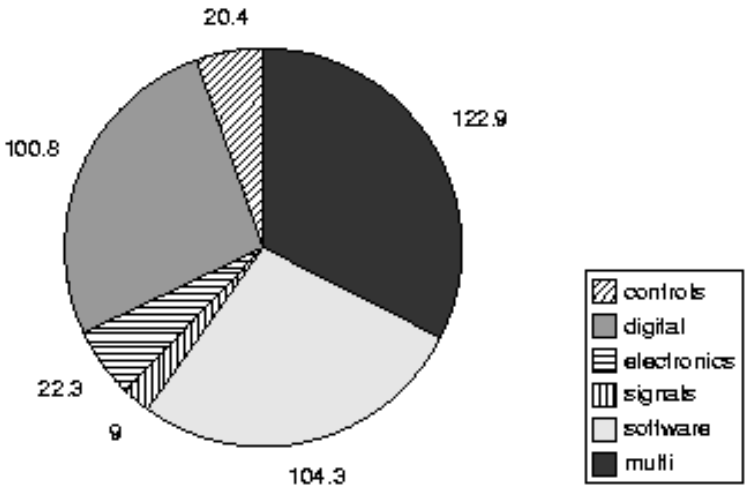

Figure 2: Design AUs by Technical Area

in the senior year as students select specialized courses, which typically do not contain a significant design component.

The distribution of design instruction by technical area is illustrated in Figure 2. Not surprisingly, given the above definition of engineering design, the largest single component is multi-disciplinary, primarily due to the design project courses in terms 6 and 8. The two other significant blocks come from the two main broad technical areas of Computer Engineering: digital electronics and software.

2.2 Planned Changes

Our programme continues to evolve as demands change or we find new ways to improve it. The following are some changes that are currently going through the formal approval process.

Extended Capstone Project We recognize that the four-month period dictated by an academic term is quite short for substantial design projects, but the co-op nature of our programme doesn't allow us to extend a course over two terms. In the past we have attempted to overcome this with regard to the term 8 project by asking the students to choose their project before the end of term 7, and encouraging them to do some initial research and planning while on their work term. To formalize this process, and to recognize the substantial effort required to complete the project, we are introducing a one credit (one hour per week) course in academic term 7, which will act as a precursor to the term 8 project. In the new course the students will be expected to produce an initial document detailing their project requirements and a high level functional design for their system.

"Offshore Oil & Gas" Option The oil and gas industry in Newfoundland is rapidly growing and there is a perceived demand for engineers of all disciplines with specialized skills

relevant to this industry. To help satisfy this demand we plan to offer options in "Offshore Oil & Gas" in each of our engineering disciplines. These options are implemented by selection of specific technical electives and so do not impact the analysis presented in this paper.

Engineering Graphics & Engineering Design As discussed further in Section 3.1, our introductory engineering design courses are being re-organized and re-designed to ensure the design content is clearly laid out and consistently presented.

3 Design Content

This section considers some of the questions posed earlier with regards to where, when and how design should be taught in Engineering programmes.

3.1 Design in Junior Courses

Many engineering programmes include introductory course with titles such as "Engineering Design," which serve to introduce students to the process of design, including such topics as design criteria, solution generation, solution evaluation and feasibility analysis, and may also include graphics (design and assembly drawings etc.). In such courses it is difficult to "integrate" other topics, as the definition in Section 1 requires, since the students have yet to be exposed to very much of those other topics. However, it is possible to guide the students through the *process* of design using examples where the technical details are either familiar to the students or are abstracted away for the purpose of the exercise. This is appropriate at this stage, I think, because it ensures that the students get early exposure to the design process so that they can be aware of how future material may fit into it, and it allows students to understand the process itself without it being lost in the details of the technology, which can tend to dominate in later courses.

The structure of such courses should clearly distinguish techniques and technology from the design process itself. At Memorial we have recently re-organized our introductory courses, which were formerly known as "Engineering Design I" and "Engineering Design II," to make this distinction clear. The new courses are "Engineering Graphics," which concentrates on the techniques (graphical projections etc.) and technology (CAD systems), and "Engineering Design," which leads the students through the design process using realistic examples. With one class through these new courses, early indications are that this move was successful — the students seemed to gain a solid appreciation for the design process and were able to complete and present impressive projects (of course they did not implement the projects).

3.2 Analysis vs. Design

The "iterative step" in the engineering design process typically involves analysis — does a proposed solution satisfy the requirements and is it appropriately optimized? A solid

understanding of the necessary analysis techniques is therefore an essential precursor to any detailed design exercise. In teaching, and examining, technical courses there is often a trade-off between instruction in analysis techniques and design, and it is important to strike the right balance. Early courses in a particular subject area should have a strong emphasis on analysis techniques so that the students develop a sound understanding of the techniques and the behaviour of the systems in question. More senior courses can, and should, put more emphasis on exercising design skills, which will include using the analysis techniques as described above. Effecting this emphasis can be as simple as posing questions in a different way: instead of presenting a circuit and asking "what is the signal at point A?", ask the students to design a circuit that produces the required results. Of course it is not always possible to pose questions in such a way, and such questions typically require more time both for the student to answer and the instructor to evaluate. However, design is not something that can be learned by rote, so if we wish to graduate engineers skilled in design it is essential that they get this practice.

3.3 Programming

If we agree, which I think we will, that software is an appropriate "system" for a computer engineer to design, and that software design is engineering design, then there is a temptation to claim that any course in programming has a significant design component. We should be very cautious in yielding to this temptation, I think, since it can quickly lead to unfounded claims. While, designing a software *system* is certainly engineering design, implementing a given algorithm in a particular programming language certainly is not. An essential aspect of a design exercise is that the students must be required to make non-arbitrary decisions to ensure that their design satisfies the given requirements.

In early programming courses the students will make many decisions in solving a problem, e.g., identifier names or order of unrelated statements, but these are typically not design decisions in that their choice will not significantly effect the product. In order for a problem to be considered a design problem, it must be substantial enough that the student can make some choices and, through analysis, evaluate the success of those choices. Thus early programming courses typically contain a very small design component, but more advanced courses in which students design data structures, module interfaces, and complete software systems clearly have a larger design component.

3.4 Laboratories

There is a temptation in electrical and computer engineering to equate laboratories with design, since these are the periods when the students get to work with the "real thing." However, just because the students are getting hands-on experience doesn't mean that they are doing design. On the contrary, the component of a laboratory exercise that is truly design is typically in the preparation for the lab. A student who comes into a lab and starts putting together components without having planned the circuit beforehand is 'hacking' (to borrow a software term), not designing. In the same sense, having done the preparation, it is

beneficial, but not essential to go into the lab and try the circuit/system to see if it behaves as expected (see the next section).

3.5 Synthesis

Because computer engineering typically involves software and/or small, inexpensive electronic components, there is a tendency to consider implementation of a system as part of the design exercise. While it is undoubtedly beneficial for the students to be exposed to the idiosyncrasies of real systems, this is not an essential part of design *if* other suitable means are available to assess the design (so that the process can iterate). For systems that include software, however, current state of the art dictates that implementing the system is perhaps the most cost-effective means of assessing a design.

3.6 Assessing Design Content

As stated earlier, judging the quantity of design in a particular course can be difficult. Two clearly defensible sources of information that can be helpful are:

Students Gathering feedback from students is fraught with errors, but if it can be effectively gathered, it can be a very effective mechanism for judging the success of any component of the instruction. If the students think they learn about design in a particular course, then in all likelihood they do. If on the whole they don't, then it can reasonably be concluded that the content is not present or is not effectively communicated.

Evaluation materials A more objective measure, and one that can be easily gathered, is the evaluation materials for the course: assignments, labs, quizzes and examinations. These also have the advantage of giving a simple numerical measure: if 20% of the student's grade is due to design questions, then it can be reasonably concluded that at least 20% of the course content is design.

4 Conclusions

The answer to the question posed in the title, "are computer engineering students getting enough design?" is, I think, "yes," at least at Memorial. The analysis in Section 2.1 shows that our programme clearly satisfies the CEAB criteria in this respect, and in my opinion, which is shared by my colleagues, our graduates have had sufficient exposure to design to adequately prepare them to be effective computer engineers. This paper showns one way — the Memorial way — that engineering design can be effectively integrated into a computer engineering programmes without compromising any other aspect of the programme.

References

[1] Canadian Council of Professional Engineers, 80 Elgin St., Suite 1100, Ottawa, ON K2P 2K3, *Canadian Engineering Accreditation Board Accreditation Criteria and Proceedures for the Year Ending June 30, 2000.* Available from `www.ccpe.ca`.