ABSTRACT

Subjects are the principle visible unit in most degree courses. Students tend to focus on passing subjects, sometimes to the detriment of understanding the concepts, their context and relative importance to the subject, degree and profession. The material being taught is effectively fragmented by the mechanism used to deliver it. This paper outlines a view of degree courses that promotes *threads* as a visible unit, to put back into place the context and importance of material, by drawing the focus beyond the subject boundaries to the body of knowledge or theory that the material belongs to. Links between these threads are considered to promote learning efficacy and to more easily facilitate course changes.

INTRODUCTION

There is a tension in technology-based degree courses between maintaining relevance and course stability. Whilst potentially new material grows rapidly, existing material does not become irrelevant at nearly the same rate. Careful selection of material to be included is therefore required, and when it is included it needs to be integrated into the course in a way that causes minimal disturbance, and provides maximum linkage to existing material. More importantly, the removal or replacement of material can not be done in isolation from the remainder of the course, and the impact on the remaining material must be carefully assessed. Furthermore the reordering of material within the course may have an impact on the learning efficacy of students even if there are no prerequisite relationships involved.

Traditionally dependencies between material within a curriculum are represented at the subject level, using prerequisite and co-requisite relationships. This is a reasonable representation for relatively static courses, and one may plot a path through a degree course by drawing a graph of the required subjects and possible orders that they may be taken in. These relationships are of less use when planning changes to a course however, as they are tied to particular subjects, and not the concepts within subjects which are likely to be redistributed. Nor do they provide an overview of how specific material relates to a larger body of knowledge but only show assumed knowledge or skills at discrete points within a course.

We propose a method of representing dependencies within a curriculum based on “threads” which show the relationship between modules rather than subjects, allowing material to be moved between subjects without loss of continuity. Furthermore, this view of a course also provides a representation of the relationship between concepts and the bodies of knowledge that they belong to. This paper describes our course structuring terminology and shows how it can be used as a model of the structure and progression through a degree course. The application of this model is discussed, as well as the method of using it to improve the ability to incorporate incremental refinement into courses.

A PROBLEM OF FRAGMENTATION

In part, our thread model has been driven by a major course restructuring that our school endured. In the review of the degree leading up to restructuring, and in the restructuring process itself, there was considerable concern about how effectively key ideas were cemented into the minds of students. Furthermore considerable concern was expressed by the teachers of later year subjects that key, basic information had not been fully understood and had not been retained. This has been expressed as being caused by fragmentation (Leaney 1993), which leads to poor learning.
In the UTS subject Data Acquisition and Distribution Systems (DADS) in 1991, students complained that they could not solve a Wheatstone bridge during the time of an examination because they weren’t told that it was going to be specifically in the examination. This clearly shows a poor understanding of whether information is important and why any particular information should be learnt. Although in an engineering degree we cannot (and nor should we try to) teach students everything they will need during their career, there is certain key information which they must always know. Information is often learnt because it will be in the exam, without any thought to how it fits into a body of knowledge, or whether it is foundation material that will support learning in subsequent semesters.

**THEORY VIEWPOINTS**

Viewpoint analysis is used in Systems Engineering to establish the requirements of each of the groups of people (or "things") interested or involved in a system. It requires that each group's view is established, recorded in a meaningful fashion, and ultimately integrated into the total system solution.

For example, in the specification of a control system for an elevator system, viewpoints which are relevant are those of the passengers, those of the owners of the building, those of the people required to maintain the lift system, those of the people who are required to operate the lift system.

A degree can be considered as having viewpoints according to the various theories being taught in the degree. An Electrical Engineering degree can be seen in terms of network theory, field theory, software theory and signal theory amongst others. Each of these theories provides a different view of the same degree. The concepts that make up each of these theories will be spread throughout the degree, and delivered within specific subjects.

Unfortunately, the packaging of the concepts into subjects often obscures their context within the theory. Students deal with subjects as obvious entities, but they exist as discrete units, taught by different academics and the overview of the theory that runs through the degree is not explicit or visible.

**Threads and Sub-Threads**

By tracing the concepts in a theory through the degree, a thread is identified. Threads show how a theory is introduced, how it may be split up, how it is reinforced and how it is used throughout all the subjects. Threads are an explicit representation of concepts existing within a theory.

Examples of threads within the Electrical Engineering degree at UTS include a software thread (Figure 1), a computer hardware thread, a control thread, an analogue electronics thread etc. Each of these threads can be traced from the beginning of the degree through each subject that builds on the theory or body of knowledge that the thread represents. A thread terminates when no further subjects teach modules that belong to that theory. Other subjects may still require application of knowledge that was taught in the thread, but they do not teach concepts from that body of knowledge.

![Figure 1. Part of the software thread in the School of Electrical Engineering at UTS.](image-url)
Threads may be broken down into smaller threads or sub-threads which trace one specific aspect of a thread. For example in the software thread in the electrical engineering degree, development methodologies, programming languages and software crafting are all component threads.

Strands

A number of threads may be grouped as a path through a degree to form a strand\(^1\). In this context, a strand is similar to the notion of “majoring” in a particular area of an engineering discipline. The School of Electrical Engineering at UTS offers students three paths through the electrical engineering degree. There is a telecommunications strand, an instrumentation and control strand and a power and machines strand. The Computer Systems Engineering (CSE) degree and Telecommunications Engineering (TE) degree are also considered to be strands (in this context), as they are specific paths through the pool of subjects offered by the same school leading to a degree.

Terminating Threads

A strand need not include an entire thread. (Indeed it could be said that an undergraduate degree could not possibly contain all the concepts that exist within a body of knowledge that forms a theory). Therefore threads may terminate at different points within different strands. For example, for students doing the electrical engineering degree the software thread terminates at the completion of the subject Real Time Software and Interfacing (RTSI), however CSE students complete several more subjects which teach modules in the software thread (Figure 2). The thread is said to terminate at RTSI for Electrical Engineering students.

![Figure 2. An example of a terminating thread.](image)

Modules

If threads trace concepts within theories running through the degree, then modules are the units which they pass through (Figure 3). A module is the teaching unit in which a concept is delivered, and is contained within a subject. Subjects contain one or more modules, possibly from more than one thread.

\(^1\) A strand is made up of threads and not the other way around because of the historical precedence of the term “strand” within our school.
Figure 3. Modules in Software Development One, on the Software Thread.

Modules are the smallest unit that can be delivered as “stand alone” and cannot be further divided. A module may run across several lectures, tutorials or workshops, but the individual lectures are not complete, and only cover the concept properly when part of the package.

Because modules are a teaching entity rather than just a representation of content, it is possible for a module to have more than one thread pass through it.

Nodes

Nodes are conceptual points outside the modules. Ie, they are milestones on a thread that show progression through the theory or body of knowledge which the thread represents. Nodes can be expressed in terms of preconditions or post-conditions on modules, and are useful for identifying the relationships and dependencies between modules and therefore, the prerequisites and co-requisites that exist between subjects.

Nodes may also be useful in helping to tie the thread to other threads, and synchronising the timing of material in the thread to the remainder of the degree. Threads provide an ordering of material, because they show a largely sequential relationship. They do not however provide the timing of that material relative to the other threads in the degree. Nodes provide one place for binding the separate threads.

BINDING THREADS

Threads are important as the vertical traces through a degree, however the horizontal links at various stages in the degree are also very important. Just as a woven piece of fabric contains both a warp (vertical threads) and a weft (horizontal threads woven into the warp), so too a degree must provide linkages between the various theories within it, so that it is more than a collection of loose threads. These linkages take two main forms.

Linkages at Nodes

If a subject contains modules from more than one thread then there are nodes from each thread immediately before it. These nodes represent the prerequisites for the subject, and this can be viewed as a horizontal link joining the threads at the nodes (Figure 4). These are special points in the degree because they represent a merging of the different views of the body of knowledge that the degree represents.
Linkages Between Modules

The second kind of horizontal link is more subtle. Links between modules may be (but not always) identified in a degree by co-requisites. These are the links that enhance learning because one module refers to another module in a different thread, providing a different view of the material. These modules may be within a single subject or they may not be.

Links between mathematics and the subjects using the maths techniques are obvious examples, such as the explanation of the way Laplace transforms are used in network theory. A less obvious example is a subject from the professional development strand, Engineering Discovery (ED), which uses the design and production of an electromagnet and an educational game about electromagnetics to teach project life-cycle and management at the same time electromagnetics principles are taught in another subject, Electrical Engineering Two (EE2) Figure 5.

The important aspect here is timing. When students take these modules concurrently, the learning of each module may be reinforced because of the different context provided by the other. If the modules are taken consecutively, as is often the case where one subject is a co-requisite of another but not vice-versa, then at best only one module will benefit from this reinforcement, and possibly students will not even connect these fragments because of the semester or more gap between them.

The electromagnet example above is interesting because it highlights the need for a mutual co-requisite. ED requires the subject EE2 because it does not teach electromagnetics but allows for the application of that theory, however EE2 benefits because of the reinforcement gained in the practical application in ED and the different context (viewpoint) provided by the professional development thread.

BALANCING ANALYSIS AND SYNTHESIS

Since an engineering degree must go beyond mere engineering science, there must be a balance within each thread between analysis and synthesis applications of the concepts being taught. If we consider a system life-
cycle model, synthesis is the forward path between phases in the life-cycle and analysis is the generation of
the earlier stage from the proceeding one (Figure 6).

![Figure 6. A thread will involve both analysing and synthesising.](image)

**THREADS AND COURSE CHANGES**

Threads may be used to ensure that integrity and continuity are maintained when courses are restructured and
that new material may be integrated smoothly into a course.

To do this one must identify the threads that exist implicitly in the course. This is done by considering the
different viewpoints that exist of the discipline. These may be found by considering the major theories or
bodies of knowledge that are used by the profession. This is the **vertical** grouping of material.

Having identified threads, the modules should be isolated so that the boundaries between modules are clear.
This will allow for modules to be redefined and redistributed amongst subjects if necessary, but minimise the
impact on the concepts being taught. Specifically it should help ensure that material is not lost when changing
the subjects in a course because the modules are explicit.

Finally, one looks for the weft in the degree by identifying the **horizontal** links. In doing this the nodes are
used to identify prerequisites or one way co-requisites which arise for subjects that cover material in more
than one thread. Desired learning reinforcement may be considered when (re)designing modules, which will
provide another timing constraint due to links between modules. This can be implemented using mutual co-
requisites.

**CONCLUSION**

We have presented a method for structuring courses based on threads and modules that aims to address
problems of fragmentation caused by the packaging of material into subjects which distracts from the context
of the material and focuses on the goal of passing the subject. The threads run vertically through the degree
from start to finish, and horizontal links between the threads may be used to improve the learning efficacy of
students within each module.

**REFERENCES**

Engineering, University of Technology, Sydney.