Working with Changing Knowledge:

a case study of Computing Science.

How a cohort of established academics

at a Scottish 1990’s entitled university

responded to the forces of

change, development and innovation

in teaching Computing Science


John Norman Sutherland

Bachelor of Science (Glasgow) 1978

Master of Science in Computational Science (St Andrews) 1987

Doctor of Education

The University of Edinburgh

August, 2009
Abstract

keywords: Computer Science, knowledge change, university, Scotland.

This is the report of a case study which was an investigation of how a group of long-established, in long-term and close working relationships, academic Computer Scientists working at a 1990's Scottish university have understood the many changes that have taken place in their field over their careers. It is a study that was performed by one of these Computer Scientist who had found it increasingly difficult to keep a grasp of the expanding, evolving and transforming knowledge-base that is at the core of being a Computing teacher in academia today, in the hope that performing the study would shed some light upon the nature of these changes, the forces that cause these changes, and how other Computer Scientists handle their changing field.

The study was primarily performed through open conversations that took place in the group, one-to-one between the author and his then colleagues. As such, the study is based on analysis of subjective expressions of the personal experiences of the academics involved. As teachers in a new university, previously a Scottish Central Institution (akin to an English polytechnic), their teaching was applied Software Engineering rather than theoretical Computer Science, but a part of the group were originally educated as Computer Scientists.

The study reviews the growth of Computer Science as an academic field in Scotland and compares the participants' experiences with those in other changing academic fields. The principal findings of the study are that knowledge in applied Computer Science originates entirely from outwith the academy. Commercial companies, philanthropic groups, end users and students all bring Computer Science knowledge into the academy. In order to teach the subject, these Computer Scientists must actively seek to gather in this knowledge, filter it, and apply it in their teaching.

The knowledge is volatile, difficult to provenance, only partially knowable, and time-stamped. It is not be found in books or other traditional academic sources. The one role that these Computer Scientists bring to knowledge creation in the field is in their formulation of new degree programmes which produce the field’s new graduates and so affect the renewal and direction of the applied field of CS.
Declaration

This thesis has been prepared solely by the candidate and the work has been entirely that of the candidate. This work has not been submitted for any other degree or professional qualification.

The author reserves copyright on all and any part of the thesis and its abstract. However, the author grants The University of Edinburgh permission to publish the abstract and to authorise its publication for any scholarly purpose with proper acknowledgement of authorship. Other individuals or organisations, unless employed by a commercial concern, who wish to use the publication may quote freely from it on the condition that proper acknowledgement is made of its authorship. Commercial individuals or companies should approach the author to obtain permission to quote or use any part of this work.

[Signature]
Table of Contents

Acknowledgements
Dedication

1. Introduction

2. Literature review
   2.1 Introduction
   2.2 Academics and The Forces of Change in Higher Education
      2.2.1 Introduction
      2.2.2 The Traditional Role of The Academic
      2.2.3 The New Status of The Academic
      2.2.4 Academics and the World outside the Academy
      2.2.5 Conclusions
   2.3 Pressures on Universities
      2.3.1 Introduction
      2.3.2 Tradition
      2.3.3 The Scottish Universities
      2.3.4 Universities and Economic Development
      2.3.5 Expansion
      2.3.6 Managerialism
      2.3.7 Private Training and Knowledge Production
      2.3.8 Conclusions
   2.4 Pressures on Knowledge in Higher Education
      2.4.1 Introduction
      2.4.2 Changing Knowledge Theory
2.4.3 Changing Knowledge Practice and Use

2.4.4 Conclusions

2.5 Pressures on Subjects

2.5.1 Introduction

2.5.2 Multidisciplinary, Transdisciplinary and Cross-Disciplinary

2.5.3 Management Sciences

2.5.4 Environmental Sciences

2.5.5 Conclusions

3. Computing Science Studies at Scottish Universities

3.1 Introduction

3.2 The Developing Discipline

3.3 The Computer Science Faculty and Students

3.4 Changing 'Computing' in Scottish Universities

3.5 Computing Science as Viewed from Outwith the Academy

3.6 Summary

4. Methodology

4.1 Introduction

4.2 Case Study

4.3 The Setting

4.4 The Researcher's Presence in the Research

4.5 Gathering the Data

4.6 Analysing the Data

4.7 Ethics

4.8 Drawing Conclusions & Reporting Results
5. The Method

5.1 Introduction 134
5.2 Data Sources 137
5.3 Undergraduate Studies in Computer Science at Scottish Universities 138
5.4 The Pilot Study 140
5.5 Some Principles of Design that Arose from the Pilot Study 142
5.6 Data Gathering 143
5.7 Data Analysis 145
5.8 Reporting 147

6. Findings 148

6.1 Computer Science as a Knowledge Field 149
6.1.1 The Use of Technical Terminology 150
6.1.2 Abstraction 153
6.1.3 The Currency of Terminology 155
6.1.4 Theory versus Application 158
6.1.5 Rate of Change in Computer Science 161
6.1.6 Computer Science in Relation to Other Academic Fields 163
6.1.7 Conclusion 166

6.2 New Knowledge Production in Computer Science 168
6.2.1 Knowledge Change 169
6.2.2 The Academy 174
6.2.3 Commerce 177
6.2.4 Students 182
6.2.5 Lay and End-Users 185
6.2.6 Conclusions 188
6.3 Teaching and Learning in Computer Science

6.3.1 Programming 191
6.3.2 Staffing 196
6.3.3 Management and Control 201
6.3.4 Degrees 206
6.3.5 Students 209
6.3.6 Conclusions 213

7. Conclusions 215

7.1 Introduction 215
7.2 What is happening to knowledge production in Computer Science? 216
7.3 What kind of knowledge field is Computer Science? 217
7.4 What underpinning beliefs about Computer Science do academics at this site hold? 219
7.5 What processes of teaching and learning follow from these beliefs? 221
7.6 How far have things changed over the years? 223

Glossary of Terms 227

Bibliography 230

Appendix A - The Original, Accepted Research Proposal (edited) 271
Appendix B – Source notes for Secondary Data Gathering on Undergraduate Studies at Scottish Universities (selected) 292
Appendix C - Exemplars of Requests and Replies for Data from Scottish Universities on Computing Applications (selected) 303
Appendix D - Interview Transcript for 'Dr Blue' (edited) 306
Appendix E - Data Coding Examplars (selected) 322
Acknowledgements

Five years is a long time, and I have been grateful for the support of many during this period of study, reflection and writing. Firstly, I must acknowledge the central place of four people who have been there for me whenever I got, well, stuck and needed encouragement, inspiration and hope that one day this study would see the light of day in this form. Firstly to my Lord Jesus Christ, who ever comes first in my life. Secondly to my wife Hazel, who only ever takes the place directly behind him; she gave me time to talk and listened; she allowed me to keep on saying, 'just one more chapter and it'll be over (by Christmas?)'. Thirdly to my supervisors, to Professor Jenny Ozga who ensured the study was a properly bounded case, and the Dr Charles Anderson who ensured the report was properly done.

I must acknowledge the support of two universities who part-paid my fees but who must remain anonymous, as to name one would be to reduce the anonymity of the other. To Ian and Colin who supported me during often difficult times, and the now eponymous cast of Reservoir Dogs who were my colleagues and co-subjects in the study. To Professor Mary Simpson who persuaded me to enrol on an EdD, not a PhD, and who was right, it has taught me far more than a purely research degree. To Professor Lindsay Paterson who encouraged right-thinking and corrected error. To Professor Ray Land who made postmodernism understandable. To the students of Cohort 3 who emailed, eat, drank, laughed and encouraged each other, with a particular word for Josephine and Brian who were both prepared to be silly, and Alex who was prepared to dissent from the norm.

As a Computer Scientist I must give a special word for Apple Computers for creating the Mac Mini, MacBook, OSX and Pages, to Mozilla for Firefox my essential webserving buddy, which all work, and by which this report was crafted. To the Scottish universities which opened their archives and St Andrews University in particular which allowed me free deskspace to sit, think, read their books and copy their journals in that lovely space which is their South Street library by The Scores.
Dedication

I dedicate this report to two professors who have been such a real influence upon my career as a Computer Scientist. Firstly to Professor Ojika Takeo (小倉武雄市) of the Faculty of Engineering at Gifu University (岐阜大学) and also to the late Professor A 'Jack' Cole, founder of the Department of Computational Science at the University of St Andrews.

Ojika-sensei is a bundle of energy who propelled me through virtual reality into video games and said: John, get a PhD.

Jack was an inspiration and sensei to so many of us. He taught us Αἶν αἰσθεῖν - always to seek to be the best.
1. Introduction

This study was driven by a combination of personal curiosity and opportunity. The personal curiosity was prompted by my own career-long experiences as a Computer Scientist\(^1\) and as a teacher, in particular by the difficulties I experience in staying on top of a field where knowledge change appeared to be increasingly difficult, even impossible to keep up with.

The opportunity was that, for a brief window of time, a group of similar academics could be gathered together who between them had over 200 years of experience in the short-lived field which is Computer Science (CS); for, although we worked together for much of the previous two decades, we have now been scattered by retirement, severance and redeployment.

The study is therefore one of a particular time in the life of a field, with a particular group of academics, reflecting on their career experiences in open one-to-one conversations. The aim of the study is to bring out some understandings from these reflections and to air how these experienced academics in this young field have made sense of their professional calling.

In over thirty years of working in CS, from the mid-1970’s to today – a field which I, like many of my generation of CS academics, fell into more by accident than by design - much has changed and continues to change. My earliest experiences were punching short Algol-W programs onto Hollerith cards, placing them in a pigeonhole in a building in Glasgow, from where they were modulated to be sent by phone line, demodulated in Newcastle, compiled and run on a computer, the output then

\(^1\) the terms, ‘Computer Science’ and ‘Computer Scientist’ are difficult to define precisely. I studied ‘Computing Science’ at bachelor’s degree level, read for a master’s degree in ‘Computational Science’ and first taught as a ‘Lecturer in Computer Studies’ in a ‘Department of Mathematical and Computer Sciences’. 9
to be modulated and demodulated again via the GPO, printed on fan-fold paper, thence to appear in the same Glaswegian pigeon-hole wrapped round the original cards by elastic band twenty-four hours later. I took to this immediately, I confess; I fell in love with CS from the start and put my Natural Science studies to one side.

On graduating, my commercial programming was mainly for finance systems: thousands of lines of ANSI COBOL running on an IBM mainframe which filled a huge room with its boxes, cables, industrious people, noisy printers and flashing lights. Then a short bit of economic modelling in FORTRAN-IV for another IBM mainframe. Next I customised BASIC suites of Accountancy code that ran on Data General minicomputers. Then I created stock control systems in RPG II and COBOL for ICL mainframes and IBM minicomputers. All this in around three years of work across the United Kingdom.

I then researched relational database fourth generation languages (4GL’s) before starting my first teaching job as a Lecturer in Computer Studies teaching Pascal programming on DEC VAX minicomputers to Electronic Engineering students. I moved on to do research in Virtual Reality on PC’s at a university laboratory in Japan before moving into teaching Video Games design and development which, ten years on is still my CS subfield of work.

As things have continued to change around me in my professional life, so they must be changing around others. This study, I hope, will be of interest not merely in and of itself as a study of a particularity, but also to those who are seeking to create understandings of work-life change. I do not prescribe what these future readers’ interests or concerns may be. I simply present a case study of change in acadæmia as it was understood at a particular moment in time and by one concerned researcher/practitioner.

The dissertation presents, albeit in an academic format, the story of this research project in seven chapters.
Chapter 1 – Introduction sets the tone for the study and for the report. It should help you understand who I am, why I wanted to do the study, why it might be of interest to the reader, and how the rest of the report flows on to the end.

Chapter 2 – Literature Review places the study within its academic context and seeks to ground the study in current knowledge. The chapter considers first the position of the academic, often as an embattled species having to work with a range of forces acting upon him (or her); this was useful as it contextualises the direct stresses upon the author which drove the need for the study. As the case study site is one of an establishment, which - like much of Scottish and UK acadæmia - has undergone considerable restructuring and sectoral repositioning in recent years, so the chapter continues with a consideration of how universities, in general, are reported as having been affected by recent changes.

A significant issue, which only became clear once data gathering had begun and analysis started, was that there was something inherently unstable about knowledge in CS, at least as it was being understood by those involved in the study. It was decided to investigate what has been said about different kinds of knowledge that could be applied to the CS as it was perceived to be at the study site.

Lastly, in this chapter, the ever-changing nature of degree offerings in CS (as also evidenced in chapter 3) needed to be put in a context of other academic subjects being stressed and changed in ways that are particularly relevant to applied CS as taught in this particular part of the CS academy.

Chapter 3 – Computing Science Studies at Scottish Universities is a secondary data review on how CS has been and continues to change at Scotland’s universities. I was given access to the archives of several universities and data are presented and interpreted on how the past prospectuses of these institutions show evidence of CS as a growing and changing taught field. A feature of undergraduate
taught CS is that its arrival in Scottish academia postdates the archiving of prospectuses and so the analysis is close to complete. This is consolidated for analysis with data on numbers of students applying to study and studying CS at a range of institutions and over a period of years.

Chapter 4 – Methodology describes the pre-study thought processes. By academic training I could consider myself to be a Computer Scientist, a particular kind of Mathematician or a Software Engineer. As a newbie in qualitative educational studies I thought it best to be up front about this and to allow the reader to be reassured that the study proposed, performed and reported is one that is valid. However, I still think as a Computer Scientist even if the study is a qualitative one so any errors in thought are, I hope, founded in my interpretation of the meanings of educational studies rather than in poorly understood or applied techniques.

The chapter contains particular considerations of the nature of the case study, the significance of the study’s setting, the role I would play as an actor in the study (it was my workplace and the other actors were my then colleagues), the means of gathering in data, analyzing and reporting from it, and the importance of dealing with all partners in the study in an ethical way.

Chapter 5 – The Method is a detailed and honest account of the performance of the study and its reporting. Most importantly it is at this stage that the two important issues of the case and the research questions are mature enough to be revealed. The chapter details the sources of data which was gathered and their types, the piloting that was tried out and what came from these, the techniques that were used to perform data analysis (which was principally from interviews), and the care taken in reporting to ensure all parties would be happy with what is said in this report.
Chapter 6 - Findings is an extensive account of the meanings found within the data, which was gathered and analysed. The first aspect is what the academics had to say about CS as a field of knowledge. I have tried to keep this as non-CS technical as possible whilst keeping it entirely based on an analysis of CS as related by Computer Scientists, so there are reports on the field as communicated in words and pictures, a consideration of how these terms come into and fall out of use and what this says about the realities behind these words, the significance of CS as a field with theory and application issues, the rate of change of knowledge in CS and, finally, how CS relates in this site closely to other fields.

The second part of the findings is that CS knowledge is being continually reviewed and renewed. But, just who is doing this, from where are they creating this new knowledge, what kinds of knowledge is this, and how does this affect the work of those teaching in the academy?

The last part of the findings chapter is given over to reporting how these academics reported the stresses upon the field as they attempted to teach it and ensure their students learnt it. It is here that the essence of what CS is to these academics is revealed. Consideration is given to the roles of the different parties in the working of the academy: the academics themselves, their interfaces to management, the degrees they work on, and the students they work with.

Chapter 7 - Conclusions these are offered as a bringing together of what the academics had said, as analysed and reported, and now brought together under the headings of the research questions. As said elsewhere, this is an opportunistic case study and not intended to be an exemplary one. It is, I believe, a good and valid case and so it is worth my while reporting how the study answered the questions that concerned me, and I hope worth your while reading it to find out some useful pointers for whatever has led you to read it.
The bits at the back - I have attempted not to overload the reader with excessive and paper-wasting padding, but to provide enough extra information in order to be helpful. There is a **glossary of terms** covering specific CS terminology (less than 30) which arise in the main body of the document but are perhaps not of obvious meaning to the reader, an extensive **bibliography** detailing everything cited in the main body of the report (except some web url’s which are given as inline text) and a very few extra references that proved useful to the study, a copy of the **accepted research proposal** which is included for academic completeness, some notes in two appendices on **secondary data gathering** which form the basis for much of chapter 3 but are also used in chapter 6 as required, a copy of one **interview transcript** chosen as it is the one from which extracts were most regularly taken for the findings, and, extracts from the **encoding process**.
2. Literature Review

2.1 Introduction

The essential core of this study is the meanings held in the conversations of a group of long-experienced academic Computer Scientists discussing their work, field and workplace. These meanings remain anecdotal until related to how, in the published literature, other academics are being affected by changes. To place what this study’s CS academics were saying there are two settings included in this report. The first is presented here: a literature review on what has been written about the forces affecting academia today and, where relevant, in the past. The second is recorded in the following chapter as an analysis of secondary data on the rise, fall and changes in academic CS, principally in relation to Scotland.

There is considerable evidence in published discussions and reports on how various fields within academia have been affected by changes from within and outwith the academic community. This chapter considers a wide selection of the literature concerning change in academia, from four related angles that when taken together provide a depth and breadth of view on the topic and allow it to be related to this particular case of the perceptions of change in CS.

§2.2, Academics and the Forces of Change in Higher Education, is a consideration of how the rôle of the academic is reported as having being impacted upon by outside forces.

§2.3, Pressures on Universities, relates how universities have been affected by recent changes, some of which are highly visible and objectively real (such as the removal of the binary line), others which are more perceived from a range of viewpoints and stances.
§2.4, *Pressures on Knowledge in Higher Education*, is an investigation of discussions on the theory of knowledge, particularly as related to propositions on tacit and explicit (or modes 1 and 2) knowledge which arise regularly in the wider discussions on issues affecting university education.

§2.5, *Pressures on Subjects* looks at how other subject areas under stress in the academy are seen to be coping with change, in order to allow the general case of CS to be put in a context in order to avoid a monocular skew.
2.2 Academics and The Forces of Change in Higher Education

2.2.1 Introduction

Academics in universities operate within their own discourse communities, the membership of which defines a large part of their lives. It is effortful to become part of such a community, but once a member, you have an insider’s access to mutual knowledge and shared understandings. Academics within their disciplinary discourse communities are constantly engaged in their own research, or a scholarship of discovery.

(Fraser, 2006)

Dramatic changes have been taking place in higher education in recent years which are disrupting the traditional identities of place, of time and of the scholarly and student communities

(Bridges, 2000)

Nixon et al (1998) tell an apocryphal, illustrative tale of a colloquium where four academics are discussing their understandings of the meaning of being an academic today. To the first academic the essential core of the profession is primarily built around the academic freedoms of personal autonomy and self-governance; however, he sees these freedoms as being increasing bridled by the ‘painful’ modern transition to academic professionalism, where the academic is being forced to acknowledge his existence in a milieu that includes wider institutional and public interests. To the second academic the essence is in being a knowing teacher,
the master who teaches his wisdom to his students. The third academic sees the development of team teaching as the new route to knowledge delivery, which is in itself a moral, not purely an epistemological question answered in acknowledging the diversity of human beings and their place in a society and in a world in which we would all hope and aim to live well. The fourth is pessimistic, seeing only job casualisation / insecurity, teaching / managing / research overwork, the widespread use of simplistic measures of quantity over quality, a lack of external forward planning, and the ever-present problem of sectoral elitism.

These are presented as being characteristic views of modern academics as they consciously reflect in a conversational way on their practice and place, and many academics would recognise the issues Nixon et al raise by proxy. Schön (1987) foresaw the need for the kind of conversation Nixon et al fictionally illustrated when he talked of their being an epistemology of education establishments in the knowledge they are expected to hold and to impart. Academics - the core knowledge workers in the academy - need to take time and space to reflect. A lack of such reflection is, to Schön, 'why all educational reform fails to reform.' In other words, the academics themselves have a significant responsibility to ensure that they are reacting positively to the worlds of change around them as active agents rather than passive spectators. However, as Dovey et al (2001) note, this is a challenging task as universities too are, like the academics themselves, including those in this study, both changing and conservative institutions.

It is the purpose of this study to illustrate one such set of academics’ conversations.
2.2.2 The Traditional Role of the Academic

The academic can be viewed as a knowing actor (Askling et al, 2001) with a set of skills that are personal, theoretically based in the field¹, applicable to test research findings against accepted theory, used for prediction, based on accepted forms of knowledge, related to those of academics in similar fields, used for training students, greater than textbook knowledge, practised and acquirable (Thomson, 1987, pp9-10). Academics are essentially, 'individuals [who] understand themselves, with their sense of identity ..., with their being in the world; this is a world order which is characterised by ontological dispositions' (Barnett, 2004).

There is a deep cultural value to such academic autonomy and the seeking of truth for its own stake (Bleiklie and Byrkeflot, 2002) in the purposes and instantiation of academic life, however this has ever varied across institutions depending on wider contexts such as the politics of the day (Henkel, 2007). There is much evidence that attempts are now being made to make academics become less thinkers and have them convert more into being knowledge practitioners. Wensley (2002) sees this transformation project's success as being unlikely, and Ospina and Dodge (2002) see the fundamental divide between theory and practice as factors that thwart easy attempts for academics and practitioners to connect due to the significant difficulties in linking the parallel lines of practical relevance and theoretical / methodological rigour (Pettigrew, 2001).

Barnett (2004) discussed both the past and present world of academics and, on reflection, he saw one where the control of many factors which drive their work has now been lost by the academics. This, he proposed, has created new areas of uncertainty, incompleteness, insecurity, anxiety, fragility, chaos, destabilisation, multiple descriptive conflicts and paradox in the academic's life.

¹ The word 'field' is used in this study to indicate the field of study, that is, Computing Science, rather than the field of practice. Where 'field of practice' is the intended meaning this will be made explicit.
In a wider society of rapidly-changing boundaries the academics have found their work being increasingly perceived and presented as esoteric, obscure rule/logic based, and isolated from the realities of modern society. This cap may fit too well for many established CS academics, originally Mathematicians who view CS through $\lambda$ Calculus.

Academics have to respond to demands for workplace innovation in order to retain the traditional claim to be the holders and producers of reliable knowledge set in a societally recognised and supported, powerful social institution (Thorlindsson and Vihjalmsson, 2003). Traditional boundaries have shifted with respect to knowledge and to institutional settings in ways that are both geomorphic and structural (Becher and Trowler, 2001, pp45-57), challenging the academic on several fronts. CS, a late 20th century field, has hardly had time to bed in before the tectonic plates moved.

Teaching has always been an important part of academic life. Norton et al (2005) reported a wide variety of teaching approaches used, depending on the academic's stance with respect to subject knowledge, views as to the aims of teaching, material to be taught, the discipline, teaching media available, grasp of field-knowledge, workload, institution, sector, teaching experience and gender. Academics are traditionally a self-replicating (Lave and Wenger, 1991), even a tribal (Becher and Trowler, 2001) community, but one that is notably fragile (Bourdieu, 1977) in which facts and behaviour are passed on from generation to generation (Sorlin, 2002). In a field as new and volatile as CS, based in a changing world, the CS academic has to work in a complex environment where, as Cliff & Woodward (2004) stated, accepted 'knowledge is built both academically and professionally' in many subject fields. Further, academia is a profession based around knowledge, but where the knowledge-base is eclectic, grows and will ever grow out of 'a complex and often contradictory mixture of natural observation and human constructs.' So, they continued, 'academics view their knowledge as being structured by professional, historical and philosophical contexts ... [and] ... that knowledge is relational, time and context-bound, and values laden.' It is perhaps in uniquely accepting these knowledge complexities that academics are, after many years, free to validate their own 'intellectual maturity' to operate in the profession.
As such, being an academic remains one in which traditional, Newmanesque values are still important (Newman, 1976; Hill, 2004; Stiles, 2004).
2.2.3 The New Status of The Academic

Academics are no longer granted a status of unalloyed authority; and their frames of knowing are felt to be inadequate for a fast-changing world, replete with its own and proliferating frames for comprehending the world - the rules of which are at least a matter of process as of substance.

(Barnett, 2000a)

Halsey (1982) noted that,

when adversity came to the British universities at the end of the 1970’s, it was remarkable as much because it was as unfamiliar as because it was unwelcome. Dons could perhaps be forgiven for failing to notice that there is no necessary law of permanent university expansion.’

In CS the years of plenty which rose throughout the 1990’s appeared to be unending, an inevitable consequence of the paradigm-shift towards mass-education (Becher and Trowler, 2001, pp4-6) and the growth of IT applications that was taking place. They were unprepared for the sudden dot com crash of the new millennium which rebounded on the subject. Halsey (1982) stated that the existence of external pressures slowly dawned on academics to awake to the need to be aware of: the future, outsiders, finance, job security and the new status and kinds of academic work.

Although CS had grown rapidly through the 1980’s and 1990’s, there is a widely reported academics’ perception that more isn’t better. As Bleikie and Byrkerflot (2002) put it, ‘The number of higher education faculty grew as well, and university professors in particular have felt considerably less exclusive than before, as they
have experienced a declining income in relative terms ...’ and, as Bleiklie (2005a) said, academics have ‘less social elite status than they used to’ even, as Barnett (2000a) added becoming marginalised and dislocated both institutionally and societally. The arrival of new universities and new fields, such as CS, jostled with the established academy for position and recognition (Becher and Trowler, 2001, p75ff) as the new academics and their field are absorbed into the wider academic culture and attain respectability (Simon, 1996, p112).

Governments have sought to control, assess and propagate new measures of academic status through such means as RAE ratings and a passively encouraged culture of league tables. These measures directly affect the academic’s self-validating status, and have become distractions as time is spent in consideration of how to play the ‘ratings game’ (Starkey and Tempest, 2005) of where the local institution and academic subject stands on public-facing measures which are contradictory, simplistic, opaque and not open to easy challenge.

To Barnett (2004), such pressures have generated a Higher Education world where the place of the academic has become radically unknowable, unclear, contested, destabilised and unstable. However, this also may be an inherent feature of our profession, as illustrated by Becher & Trowler’s (2001, p41) influential thesis that the concept of an academic discipline itself is not straightforward. However there is evidence of a recent, significant breakdown of knowledge boundaries that has undermined the idealistic professional self-government that may exist in the mind of the academic (Henkel, 2007). While Henkel generally agrees with Becher and Trowler, she also cautioned that professional academia is inherently unstable. She would remind us that what is for many the core professional strength - academic freedom - is not something that was ever given as of right nor when won was it ever been won for all time.

Recent changes to modes of learning have changed the teaching roles of academics too. Barnett (2004) reported a nursing lecturer as having said,

One day we [the teachers] turned up half an hour later than them [the students] and they were doing exactly what they would have been doing with us. They were organizing a discussion and ... they didn’t need us
for that. We became more their resource.

For today’s academic, at least in applied fields such as CS and Nursing, his status and authority as an expert is neither automatically given nor is it self-evident to students (Thorlindsson and Vílhjalmsson, 2003). This lost élite societal status of the creative knowledge user, which was as much imposed upon him by others as expressed by himself, has now shifted elsewhere in popular eyes to such as the entrepreneur (Carlaw et al, 2006). Academics have no TV shows to compare with Dragons’ Den.

Askling et al (2001) and Garraway (2006) noted that both students and employers are now also involved in curriculum design, challenging and fragmenting the nature of knowledge (Clark, 1996). These forces further seriously challenge the traditionalist academic’s claims to unique epistemic authority and responsibility.

As a third blow to the academic’s power to control knowledge definition and delivery, Delanty (2003) detailed how university managements have used managerialist packaging of modular, consumer-driven education where the pedagogical labels, structures and content are defined by administrators. It can be particularly frustrating for a CS academic to be told, for example, that a module title can only be 28 characters long, as he can see that this arbitrary limit is caused by poor knowledge of CS.

As a result of a raft of pressures, change is taking place today in academia in the “how” and the “what” of academic work. Academics are being seen as becoming, in some cases, mere mentors, practice assessors or programme managers rather than traditional knowledge-holders (McLoughlin, 2004). They are likely to be seen more as ‘facilitators ... [as] learning becomes much more experiential and is complemented by reflective processes that help students to enhance their capacities to deal with life and work’ (Askling et al, 2001). Such changes are a power-shift from knowledge to practice, from knowing-and-doing to reflection-on-action, from sensei-gaksei to coach-and-student, from knowledge base to reflective practicum, from marking-and-correcting to counseling-and-consulting (Schön, 1987a). After a long discussion of what was happening to the role of the academic,
Taylor (1999) saw this change from theoriser/researcher to teacher as the primary and significant trend in the role of the academic.

Kogan (1999) criticised this emergence of a new kind of academic life where few have an interest in theory testing and paradigm building as instead we merely react to the impact of externally-imposed changes in workplace policy and power. He saw a deeper loss as these mutely accepted changes affect the way in which academics think about themselves and their areas of academic work. On the other hand, many academics are responding inventively to these forces by becoming, simply, better teachers, an essential within CS as it is based around the core and hard-to-teach/learn skill of programming. As McInnis (2005) stated,

There is a growing awareness of the importance of evidence-based approaches to the organisation of the learning process. That means universities and academics routinely collecting evidence about how much their students have learned and modifying their approaches accordingly.

Notwithstanding the gains from keeping a better eye on skills teaching, Carlow et al (2006) reported a perception of a loss of professional standing by academics themselves, the creation of a mill mentality and a workplace where we have slipped into becoming mere human capital in a post-industrial proletariat consisting of poorly-unionised, part-time and casual workers. This has created new patterns of social inequality for new academics, such as casualisation, and a loss of legitimacy in a society which is increasingly understanding less, questioning more and skeptical of experts. Such a state is a long way from the idealised world where tenure was the guarantor for the academic to set his own agenda. We instead have often become part of an industry of academic capitalists (Henkel, 2005) who marketise knowledge (Becher and Trowler, 2001, pp8-10).

The life of an academic continues to be one of considerable variety. Blaxter et al (1998) interviewed many academics and noted a workload that included such activities as marking exams, encouraging former students, writing references, trying to gain the title 'Professor', finding obscure references for students, comforting distressed students, fighting with academic and non-academic
colleagues, living in fear of league tables and RAE scores, enduring bullying, attending meetings, breaching the Fourth Commandment ('keep the Sabbath day holy') by working every hour God gives - and yet they also note that this wide range of tasks is not in itself seen by many academics as a problem to be solved, more as the humdrum grind of academic life. This plethora of tasks can be stimulating and a boon, but they also note that, in the academics they interviewed, there seemed to be a loss of happiness and satisfaction in this world of work overload (see also, Barnett, 2000). It is difficult to attempt to square the triangle of time, workload and job satisfaction and also to remain the academic who is, according to Garraway (2006) and Gunasakera (2006), flexible, responsive, technology aware, knowledge updating, self-programmable, able to work in a complex network, responsive to globalising influences, etc. This range of tasks is a lot to place on anyone's shoulders, and yet these tasks are often ones the academic has volunteered to carry rather than simply had imposed upon him by others.

Dunkin (2002) encouraged the academy to periodically revisit the meaning of the academic as created both by the academic himself and by those with similar rôles outwith the HE-based academy, and to remember that,

It has long been understood that academics are less motivated by absolute pay than by access to research time, resources and students. What matters to them is: access to resources for research, access to time for scholarship, a degree of autonomy, an environment of debate, peer-reviewed status, an input to decision-making, and the absence of administration.

In the light of the previous paragraph's discussions on tasks, it could be added, perhaps sarcastically: Oh, is that all? Perhaps even the now widely-reported buzz of complaint in academia is itself an essential core to being an academic? The now-gone computer company, Digital Equipment Corporation (aka, DEC) used to issue a list of reported errors in their software and actions taken. This was a 'feature list' as, they said, there were no bugs in their software, only features. Overwork and a dislike for external controls could be seen more as a feature rather than as a bug within academia.
2.2.4 Academics and the World outside the Academy

The research scientist functions differently in each of the systems in which he can generate some knowledge: his culture, political system, legal/economic system, libraries & journals, professional group, work organization, shared reference group, preferred colleague group, work associates and his own head.

(Short, 1973)

Academics are not the majority of employees in universities, constituting, for example 43% at University of Glasgow (2008) and 47% at Napier University (2008). There are now in the universities many new actors with different interests, forging new alliances that tilt the inherited power balance within universities (Bleiklie and Byrkeflof, 2002). From being at one time the drivers of academia, the recent input of managerialism (Becher and Trowler, 2001, pp10-14) has caused a loss of power and influence from the academy in absolute terms. This can be seen as academic appointments are now often made with non-academics involved in decision-making. So those within the university but outwith the academy are influencing the renewal of the academy. Although it continues to be the case that academic appointments are primarily or sometimes entirely academically made, in many cases the academy’s monopoly of responsibility in its renewal is less clear than it once was and its boundaries are now also more permeable to non-academics (Tapper and Salter, 1995; Henkel, 2007). Halsey (1992, p13) noted that this was part of a wider change where the academic was becoming increasingly a salaried or even a piece-work labourer in the service of the administrators and technologists who increasingly run the universities. Bleiklie (2005b) noted that,

One of the major shifts in power relationships in and around universities is that ... the individual academic [is] supposed to serve the expressed
needs of stakeholders for research and educational services.

This has created an institutional rôle confusion for academic staff, which has caused feelings little short of dismay among many academics (Winter et al, 2000).

There are other stakeholders closely involved from outwith the academy. Politicians, who control funding, have taken a more direct rôle in recent years in the running of the universities (Bleiklie, 2005b). This has caused resentment, and a feeling within academia of being under attack from political authorities (Bleiklie and Byrkeflot, 2002; Bleiklie, 2005b).

The interface with industry stakeholders, so popular with government, also involves changes in academic workload. The research process has been and continues to be altered from one of enquiry for purposes defined by the enquiring academic to that of the possible gain to those outwith the academy. As Bleiklie (2005) put it, it is a change from a,

basic and disciplinary research mode in which the researcher defines the research problem, directs the research process and communicates the findings to the public through scientific publication ... [into ] the direction of a transdisciplinary mode in which the research problem may be defined by wider teams of people and where the customer or end-user takes part in the definition of the research problem, monitors and takes part in the research process and may influence when and how results are communicated.

Ospina and Dodge (2005) discussed the necessity of such an industry-academia relationship, particularly in fields where there is a recognised, parallel professional practice such as would apply in CS. They also discussed the difficulties that arise in the gaps in such a practice-academia relationship as practice becomes increasingly specialised, causing both disconnections and an ‘asymmetry of power relationships’ between the academic and practice partners, in which the practitioner ‘over-influences’ the other causing the ‘weaker’ academic partner’s needs and claims to be ignored. Amabile et al (2001), in detailing just such a difficult relationship in one particular project, went as far as to delineate practitioners and academics into
two quite separate professions, which, if it were to become true, would be
disastrous for CS in a new university, such as in this study. Looking at the
weakness of both parties, Ospina and Dodge (2005) disparaged the academic’s
penchant for what they call dysfunctional practices of high methodological and
theoretical rigour that have no connection with practice, which are written-up in
highly technical language and aimed at publication in obscure journals, and they
contrasted this with the practitioner’s motivation for faddy, narrow, practical needs
which are quickly satisfied. However, Becher and Trowler (2001, pp95ff) noted that
practitioners are not alone in these tendencies, as academics are also prone to
faddism, not least CS academics who are regularly inundated with news on the
latest thing. It is all too easy for each side of a field, the academics and the
practitioners, to tend to paint the other’s drives as pointless; this may help to
illustrate where their respective areas of work in the same field do not coincide, but
it could also exacerbate an unhelpful, widening gap between the academic and the
practitioner. Where there are examplars of knowledge transfer in and between
universities and practice these are often due to personalised and informal
relationships between the partners which themselves depend on the personal
chemistry and empathy (Krucken, 2003) of those involved, underpinned by a good,
long-term working relationship between the corporate and institutional partners.

The wider public (Kogan, 2005) can also stake a claim to own and affect what was
once the academic’s exclusive knowledge resource. This alters the thinking space
and the working boundaries of academic experts (Maranta et al, 2003) as
contemporary society can now speak knowledgably with academics (Henkel, 2007).
This has also challenged academic field ownership, to which some in the academy
have retaliated by calling lay-sourced knowledge mere ‘Heathrow Literature’
(Burrell, 1989). Van Aken (2005) sees this as a contest over ownership and
possession of the field – one, which is particularly felt by those academics in the
newer, softer fields which may include applied CS. This creates an uncomfortable
cognitive dissonance – and what could be worse than that for an academic? - as
one of the requirements for being able to utilise knowledge is that one must have
access to it and to its creators (Short, 1973). And so, if it isn’t now where it was
traditionally expected to be - in the mind and on the shelves of academics - new
ways must be found to locate, seek, organise and manage new kinds of relocated
knowledge. There is a further danger when, as the field expands and is
reinterpreted and reclassified by a powerful lay community, the number and extent of issues to be tackled exceeds the available headcount of the relevant academic community (Becher and Trowler, 2001, p104). This can further embed a feeling of powerlessness and lack of authority in the traditional epistemic community of the academic (Kogan, 2005). For a field in retreat, at least in terms of full-time academic staff, but in advance, in terms of innovation and change, such as CS, this becomes a very real problem for the remaining full-time academics. Evidence of this is found in this study’s findings (see chapter 6.)
2.2.5 Conclusions

'I love my job, I love [my discipline], I love teaching ... but what I want is the job as we used to know it, not as it is now. That's why I would like to get out.'

_An academic_ (Trowler, 1998)

It is perhaps not too difficult to become depressed by what has been illustrated above of the battle which surrounds and involves today's academic. Halsey (1992), and Tapper and Salter (1992) bemoaned the passing of a grander era when academics were the powerful, influential, core knowledge workers. Change has been painful for 20th century academics and the evidence is that it will most probably continue to be so for academics into the short to medium term (Nixon et al, 1998), which can be disheartening too for experienced academics with only a few working years ahead before planned retirement or unplanned severance.
2.3 Pressures on Universities

2.3.1 Introduction

Post-modern society has a tendency towards pluralism, diversity, volatility, uncertainty of outcomes and transgressivity. Previously separate institutions such as science, politics, culture and industry now more readily penetrate one another.

(Nowotny et al, 2001)

Higher Education Systems, institutions and practitioners across the world have been in ferment during the last three decades. Now that we are in an era of decline in the authority of institutions of all kinds and we embrace multiple conceptions of knowledge, ideas about the governance of higher education, about its functions in society and about the values and assumptions that have informed them are being reappraised.

(Bleiklie and Henkel, 2005)

Having studied at three universities and worked for six, with my children having studied at a further five, I have an awareness of the breadth of sector. Universities are set in today’s world, the UK and, in particularly for the purposes of this study,
Scotland. There are real dangers of drawing easy conclusions about where the university sector lies (which in Scotland politically now includes the further education colleges) or what it means to be a university in Scotland today. It is a sector that continues to change; the university in this study gained title in the 1990’s, the University of The Highlands and Islands project continues to attempt to create a single university from 14 separate institutions (UHI, 2008), in 2007 The University of The West of Scotland (UWS, 2008) formed the country’s third largest university student body, and recently Glasgow’s Metropolitan and Stow colleges announced a merger to become Scotland’s largest Further Education college (BBC, 2008). There are clearly huge forces acting upon Scotland’s universities today.

Kogan’s statement places today’s ferment of the universities in some context when he stated,

   Education is a social artifact embodying aspirations about the good life for the individual and the best arrangements for the whole of society. It is, therefore, particularly prone to change as social and economic circumstances change.’ (Kogan, 1985)

Universities exist today in a milieu of epistemological and organisational uncertainties (Lidskog et al, 2005), wandering (Enders, 2005), where external pressures (Halsey, 1982) act upon them, creating and instigating changes (Bleiklie, 2005a). They are changing as the forces of globalisation, economics, competition and work-life act upon them (Garraway, 2006). It is a time of constant change causing enormous upheavals (Pæchter, 2000).
2.3.2 Tradition

The traditional university is parodied by Bourdieu (1999) as,

... the august array of insignia adorning persons of "capacity" and "competence" - the red robes and ermine, gowns and mortar-boards of magistrates and scholars in the past, the academic distinctions and scientific qualifications of modern researchers, all this social fiction.

This has been called an impotence of vague abstractions of neutral effect with no associated activity, which at worst can be self-styled centres of intellect, both extraordinarily optimistic and surprisingly stupid. (Delanty, 2003; Jamieson, 2003; Darling, 1983). It is one of the quirks of the new universities, such as the one in this study, that they have been so quick to ape Bourdieu’s pseudo-antiquity of robes, maces, titles and other paraphernalia of the perceived *ancien academy*. This old university stuff is catching, for example the 1990’s university entitled The Robert Gordon University claims a heritage back to 1668 (RGU, 2008).

Ennals (2004) noted the university sector’s tendency to try to play safe despite the fact that it was increasingly recognised that the traditional academic orthodoxy was not sustainable and derided. In a report for Government, Wilsdon and Willis (2004) place in the past tense such university activities as basic research promoted by a spirit of enquiry.

Traditionally, what were universities for? The literature has revealed a wide variety of opinion:

- Knowledge production (Dearing, 1997, §5.11; Askling et al, 2001; Bleiklie and Byrkjeflot, 2002; Gumport, 2005; Teichler, 2005; Constable, 2006);

- Knowledge transmission (Dearing, 1997, §5.11; Bleiklie and Byrkjeflot, 2002; Leslie, 2002; Cliff and Woodward, 2004; Norton et al, 2005; Teichler, 2005; Constable (2006); Jones and Abraham, 2007);
- Creating mature, moral and civically-aware new intellectuals (Dearing, 1997, §5.11; McInnis, 2002; Cliff and Woodward, 2004; Teichler, 2005);

- Creating the next generation of academics (Teichler and Kehm, 1995);

- Creating useful graduates (Ryan et al, 1996; Toohey et al, 1996; Dearing, 1997, §5.11; O'Brien and Hart, 1999; Teichler, 2005; Fraser, 2006), or failing to do so (Teichler and Kehm, 1995);

- Preserving knowledge (Teichler, 2005; Constable, 2006);

- Categorising knowledge (Tight, 2003; Gumport, 2005; Kogan, 2005; Constable, 2006); and,

- Societal transformation (Dearing, 1997, §5.11; Askling et al, 2001; Horlick-Jones and Sime, 2004; Sall, 2004; Starkey and Tempest, 2005).

There is a traditionalist tendency in universities to delimit knowledge by extant departmental boundaries and disciplinary domains, an approach increasingly seen as being unable to address real-world issues (Winberg, 2006). Established departmental and epistemological boundaries can create what Stiles (2004) calls 'conceptual lenses' that avoid 'fundamental self-questioning' by working within safe, if obsolete, knowledge boundaries as the universities wish to be seen as the places which possess the hegemony of knowledge production (Winberg, 2006).

Universities also want to control the purpose of the education they provide and see this as a sovereign right, free, and bounded territory to be defended (Henkel, 2007). Universities nowadays are increasingly pressured by industry’s demands for work-ready skilled graduates over thinking graduates (Jones and Abraham, 2007). There have also been societal trends as work has become more knowledge-based, as graduates fill an increasing slice of the economy, and the costs of training through the expanding mass-education system becomes ever more a burden both for society (Enders, 2005) and individuals (Teichler and Kehm, 1995).
However, the university tradition of being places of argument, debate and discussion does potentially place them well to continue to play a discursive role as an environment which can guarantee free research and debate (Kivinen and Ristela, 2002) in such issues as, video games, pervasive computing, interactive technologies and other ethical and social dilemmas (Wilsdon and Willis, 2004, pp15-6). Our universities exists in a time of tension between such forces as the societal relevance of them and their graduates, their grasp of field excellence and their decreasing power over knowledge definition (Rip and Eijkel, 2004.)
2.3.3 The Scottish Universities

Government steering (Teichler and Kehm, 1995; Kogan, 1999; Bleiklie and Byrkjeflot, 2002; McInnis, 2005; Henkel, 2007) and demands for accountability (Fraser, 2006) and applicability (Horlick-Jones and Sime, 2004; Biekleie and Powell, 2005) are at the heart of many of the changes being directly imposed on its servants (Henkel, 2005), the universities, as they become ever more visible and relevant to the lives of voters (Bleiklie, 2005b). Scott (1998,2000) saw Scottish Higher Education as subject to much the same forces as were acting on other Western universities, i.e. massification, globalisation and local market economics, with a local/national instantiation of a particular Scottish problem of educational elitism which, contrary to (Smith and Webster, 1997)’s hope that massification would eliminate it, still causes, as is noted elsewhere, ‘lower socio-economic groups to continue to be under-represented’ in universities as students (Hutchings and Archer, 2000,2001; Archer et al, 2001a,2001b,2001c; BBC, 2004).

In particular, it is notable that the newer, lower-status universities such as the one in this study, are places for students from less privileged backgrounds, perhaps creating particular problems for the CS academics teaching these lower-entry qualified students in a highly technical field.

Sizer (1997) noted the autonomy and potential money-generating capabilities of the Scottish universities. Post devolution, with the re-integration of the universities into Scottish education, the Scottish Higher Education Funding Council (SHEFC), under instruction from the Scottish Executive, commissioned a detailed and wide-ranging report from PA Consulting on the future direction of Scottish Higher Education (Atkinson et al, 2006). This comprehensive report was open both in being upbeat and critical in its review of how government, universities and industry could and do/do not work together for greater national economic prosperity in Scotland, and how fit the universities are for higher education’s new proposed rôle as national economic saviour (Toohey, 1999, p7). The report began by quoting the Financial Times,
The past decade has witnessed a major shift in Scotland’s role and position in the new global economy. No longer can it be characterised as a mix of dying old manufacturing, fragile technology branch plants and a thriving financial services sector. The new picture is of a flexible, high value adding economy that is a leading player in the European and Atlantic knowledge economy.

What and who has been behind this transformation?

Fundamental to this shift has been the renaissance of a Scottish higher education sector that now provides the foundation for growing a flexible human capital, building a world-class teaching and research base and catalysing an active knowledge market.

... cementing the university-led knowledge revolution.

(Silverman, 2006)

If Silverman’s piece and the Atkinson et al report are read carefully, they actually state more of a proposed future position rather than a current reality; a hope for Scotland’s university-led, knowledge-fuelled, economically-strong future. Yet the main Atkinson et al report pulls few punches, stating that they believed that Scotland’s knowledge economy, ‘does not match its potential’. They saw significant gaps between what universities teach and what students and employers want, University knowledge often fails to make the leap to the real-world of business, government and society, ‘Universities’ wish for self-determination is contrasted with the public’s wishes to see public money used for public benefit. To these authors there is a post-modern malaise in Scotland and ‘there is little consensus over the nature and roles of higher education in the early 21st century.’

The report’s authors proposed the creation of a new consensus for the purposes of Scottish Higher Education: to become a well-spring of learning and research, a significant employment and wealth-generating sector in its own right, and a place where new ideas can be thought through. Krucken (2003) - thinking in a global arena, not just Scotland - caught this mismatch between government aspiration and university reality,
The pace of change is remarkable ... [but] the shift at the discursive and policy level is hardly accompanied by an equally dramatic change at the level of practice of central knowledge producing organizations like universities.

One academic he interviewed stated the embeddedness of the problem, 'It would be nonsense to expect a university professor to teach practical knowledge.'

Atkinson et al described a hand-to-mouth mindset in all the Scottish universities and a drive to take from the economy rather than add to it. They reported that in practice there is far less industry-government-acacdaemia transfer taking place than should be the case if the economic models proposed are right, often leaving the Scottish universities in the slow lane in a world of rapidly moving newcomers that include consultancies, publishers, the media, software providers, internet providers, and private learning providers. The direct measurement of the economic creation of new intellectual property by Scottish universities was 'relatively low for their size', yet they were turning over more government finance per unit cost than their generally bigger English counterparts. The Scottish universities were also seen to be 'highly risk averse' and showed 'enormous untapped potential ... to grow involvement in knowledge exchanges of all kinds.'

The report argued that the status quo – elitism, low morale and inefficiency – cannot continue and must be changed. The authors proposed such changes as fewer universities, more mergers of the Paisley-Bell kind, new funding regimes, capacity reduction in some disciplines, and even a single combined University of Scotland may be required in order see the economic benefit of the huge public investments in Scottish Higher Education.
2.3.4 Universities and Economic Development

Companies can form beneficial relationships with customers, suppliers, research institutes, education and the wider business community. These relationships, properly organised and focussed, improve company performance, increase business birth rate, generate innovation and attract knowledge-based inward investment.

(Botham and Downes, 1999)

Leydesdorff (2005) stated that,

The knowledge base of an economy can be considered as a second-order interaction effect among interactions at interfaces between institutions and functions in different spheres,

where the interfaces between universities, government and industry combined in a model he called the triple helix (see also Leydesdorff and Etzkowitz, 1988; Leydesdorff, 2003; Leydesdorff and Meyer, 2003; Leydesdorff, 2006.) He described a model (see figure 2.1) where,

... [the] university [acts] as the main carrier of the knowledge production function, industry as a proxy for the economic function, and the relevant level of government as responsible for the interfacing and organization of the two other functions ... (Leydesdorff, 2005).

Although this was a simplistic model (later in the 2005 paper he viewed the three main actors as all being potential proxies for the functionalities he had earlier outlined), it does outline how universities, such as the one where this study is located, increasingly could be seen as an industry combined (Sall, 2004; Gumport, 2005), capable of re-generating the local economy, with the academics as active
agents in this regeneration in a form of hybrid coalition which is a synergistic, complex tripartite relationship (Stiles, 2004).

```
public <--------> private
  \         /  \\    / \\
          R&D    Markets
               \\
       Academia    Industry
```

figure 2.1 – Leydesdorff’s (2006) model of the vertical and horizontal interfaces around today’s universities.

These rôles - mediator and co-worker within and between private industry and government - are not a new function for universities which ‘have long been recognised as providers of basic scientific knowledge for industrial innovation through their research and related activities’ (Gunasekara, 2006). Universities are places where,

innovative entrepreneurship is [itself] a significant engine of technological knowledge creation ... a fundamental engine for long-run economic growth. Innovative, typically profit-seeking entrepreneurs have been historically responsible for a large proportion of the innovation necessary to make new technologies commercially viable

(Leydesdorff, 2006).

It is increasingly an expectation (Department of Education, Science and Training, 2002; Commonwealth of Australia, 2004; Winberg, 2006) around the world’s governments to direct research and other university funding in this direction, where 'research is business' (Ennals, 2004) and knowledge is created for economic ends (Bleiklie and Henkel, 2005b). Echoing Leydesdorff's thesis, this is often an enforced relationship in a package of, 'trilateral research grants, where government funding to either universities or business is linked to cash and/or in-kind contributions by the other sector' (Gunasakera, 2006); not that this relationship

41
has always been as effective in practice for either the business or university partners (Cyert and Goodman, 1997).

This is not an easy tripartite relationship. It is one of ‘tensions’ (McLoughlin, 2004; Gumport, 2005), as, ‘corporations and governments ... regard them [universities] as a cost to be cut, and not an investment to be sustained’ (Ennals, 2004), creating irreconcilable differences (Stiles, 2004). Krucken (2003) noted that whilst on the surface academics and industrialists often speak different languages, the issue can go much deeper, and Ennals said, ‘when governments and corporations do not understand universities or research, their [the universities’] survival is in danger.’

Is it really so serious that industry and government may gang up to destroy a sector which they both need? This view was balanced by Kogan’s (1999) comment that politicians may be Philistines, but they have a duty to question how usefully universities have spent large sums of public money on finding purposes for using new knowledge in the globalised world economy (Carlaw et al., 2006).
2.3.5 Expansion

There has been a double pressure on universities since at least the 1960’s:

In the UK, for example, there has been a major expansion of the number of students in higher education in recent years, but with a significant reduction in the level of funding available to support each student. (Ennals, 2004)

Ennals further reported that, ‘students are piled high, taught cheaply ... the system is producing increasing numbers of graduates educated and trained for a past era’. This is a worrying statement that haunts a CS academic trying to create a four-year-long education/training programme in a field where much can change in one or two years, even daily. Sall (2002) called the production of unemployable graduates ‘the proof of failure of traditional higher education’ and saw this as impetus for further change. This demand for change is also fuelled by the differing demands (Askling et al, 2001) from new students (Gumport, 2004). Today’s students are, when compared to those of the recent past, more mature, younger and are socially and educationally more diverse (Scott, 1995; Reay, 1998).

The government-driven (Henkel, 2007) expansion has created a marketplace for students in which universities increasingly compete with each other and with a range of education providers (McInnis, 2002) often using latest advertising technologies such as YouTube (youtube.com). The decline in unit teaching resource has been paralleled by perceptions of a decline in student engagement in their studies (McInnis, 2002), softening of study programmes (Huisman, 1995) and an ongoing increase in graduate employment (Teichler and Kehm, 1995). It is difficult to square these perceptions with the statement of Lesley Wagner, Chair of the Higher Education Academy, that, ‘the student experience is the main function of higher education’ (McInnis, 2005).
The expansion of universities in terms of institutions and student numbers has been paralleled by a decline by merger and abolition of academic departments and fields (Henkel, 2007) with a visible tendency towards increasing specialisation in degrees and subjects being offered by institutions seeking to capture slices of the student and graduate marketplace (Bleiklie, 2005b). Funding for teaching is today calculated via a simplistic government *bums-on-seats* algorithm; and it is the lack of funds which is creating a drive in the universities to seek more students (Becher and Parry, 2005).
2.3.6 Managerialism

Organizationally speaking, we have AIDS! This new disease that is sweeping through higher education in this country takes the form of corporate management, a particularly nasty virus ... that attacks our collective immune system by seeking to destroy the democratic ways of working that are so crucial to our survival.

(Smyth, 1989)

Universities face changing conditions ... [their] internal structures and processes seem to be inadequate for meeting new challenges.

(Sporn, 1996)

Kogan et al (1994, p116) noted the drift from traditional academic leadership of the academy towards systematic management, and this with a remarkable 'zeal and speed' (Fairley and Paterson, 1995). It was hoped that this would resolve into a combined horizontal and vertical structure combining the best of external novel and internal traditional practices (Smith and Bath, 2006). Blaxter et al (1998) believed it was the very expansion of the university system, which was the primary driving force for the need to put in place better management structures. Gumport (2005) detailed how the aim was for a sector-tuned, Management Science inspired academic management of best practices. Instead the expanded and centralised role of the new managers has, she believed, created a fundamental clash of managerial neo-conservatism with traditional academic methods to the overall detriment of universities. A house divided will have trouble standing (Matt. 12:25).

Ennals (2004) talked of the tensions created by the academic's need to be a knowledge generator within 'the frequent top-down command and control culture of
a modern university’ where ‘the rhetoric of business, line management, competitiveness and globalisation has taken away the universities’ concern for academic matters.’ This tendency to Taylorism is almost universally and strongly rebuked in the academic press. Dunkin (2002) derided,

systems and practices [that] are often at least 20 years old and are based on command-control bureaucracies that dominated the [defunct] manufacturing and service industries ... alien to universities but are increasingly seen as inappropriate to knowledge-based professional organisations.

Kogan (1999) called their effects pathological and Delanty (2003) saw it as a process creating a Fordist production line within academia where students are intake, labour is casualised and the mantra is efficiency. As Anthony (1994). remarked dryly, ‘poor Wittgenstein: only two short books in his whole career’.

Rhoades (2005) raised the possibility that these new professional managers are perhaps better skilled to help universities act as new knowledge vendors. Parker and Jary (1995), whilst recognising the bureaucratic and surveillant tendencies of management, warned against becoming nostalgic, for, as (Leavitt, 1962) noted universities are organisations which are essentially built upon their human academic capital, yet they have ever tended to undervalue this humanity and devalue the person, becoming inhuman organisations. For academics such as in this study, 1962 is too far back even for them to know that life for a ‘50’s academic could also be ruthless.
2.3.7 Private Training & Knowledge-Production

The Universities’ aim for a hegemony of control within their sector of the new knowledge economy has sometimes leaked to private research laboratories (Winberg, 2006) in parallel with a loss of control over the creation of knowledge (Barnett, 2000; Leydesdorff and Curran, 2003). Indeed, in many fields, such as CS, the leading edge has long been outside the universities (Adelman, 1997) and in other fields, such as Environmental Science, external and professional organisations are leading the creation of communicable knowledge (Jamison, 2003.)

In this, the privatisation of knowledge, there may be an argument for the retention of the independent intellectual such as the traditional academic, but as Jacoby (1997) found, intellectuals seem increasingly attached, affiliated, institutionalised, commercialised and professionalised. He continued that ‘no-one can argue any longer that intellectuals form an independent group outside major institutions.’

There is a long track record of major companies having their own in-house research and development. In CS such major companies as IBM, Rank Xerox and Sony spend considerable effort pushing boundaries of knowledge and in other academic fields there are similar private sector knowledge generators (Jacob, 2000). Companies have been gaining on the roundabouts – by outsourcing research to the universities as a cheap option – and gaining on the swings too – by taking control of the knowledge created within cash-strapped universities. Further, in teaching Tennant (2000) noted that,

Universities have lost their monopoly over the accreditation, production and distribution of knowledge ... because business and industry are becoming more adept at providing learning opportunities in the workplace.

47
2.3.8 Conclusions

The University is a key institution of modernity, perhaps the major site in which knowledge, culture and society interconnect.

(Starkey and Tempest, 2005)

There are many stakeholders (Henkel, 2005) working in the life of a modern university (Jones and Abraham, 2007): students, academics, employers, industry, government and society (Blieklie and Powell, 2005). It is a world of rapid change (Hostaker and Vabo, 2005), where the older identities and structures that were with us as recently as when many of our current academics, for example those in this study, entered academia. Universities have been restructured, reorganised, rationalised, merged and reformulated (Henkel, 2005) into new knowledge networks (Sall, 2004) with new players working with them in creating and using new kinds of knowledge (Gibbons et al., 1994; Blieklie and Byrkjeflot, 2002).

The universities’ world is globalised and managerialist (Blieklie, 2005a), a place of academic capitalism (Slaughter and Leslie, 1997; Bliiklie, 2005b), cognitive capitalism (Hostaker and Vabo, 2005) and, more positively, enterprise (Clark, 1998). Universities co-exist with many other players (Sall, 2004) uncomfortably, perhaps even schismatically (Starkey and Tempest, 2005). The echo of the academy’s recent and longer-term past remains strong. Together these combine to create a reality of academic existence, which is more complicated than can easily be formulated or assumed (Bliiklie, 2005a; Bliiklie, 2005b).

However, with Barnett (2000a), I do not see ‘the end of the university’ nor ‘the university in ruins’ (Readings, 1996). It is difficult not to embrace his notion (if not necessarily his enthusiasm) for a new kind of university that can come from the ‘death’ of the old (Barnett, 2000), that has, as Wright (1988) added, ‘come through a period of transformation which involves not simply its restructuring but also a re-evaluation of its aims and social worth’. The university’s rôle in society remains crucial, but it often lags behind the discourse remaining yet too school-like, bureaucratic, institutional and administration-bound (Kivinen and Ristela, 2002).
The jury remains out as to whether universities have either as a sector or as organisations 'outlived their usefulness' (Bleiklie and Byrkjeflot, 2002). As Bleiklie (2005a) put it, 'such tensions mean that it is not easy to predict how institutions will respond to reforms'. Yet the need for more change will, so it seems, remain one of the central issue for universities into the future.
2.4 Pressures on Knowledge in Higher Education

2.4.1 Introduction

Be warned, my son ... of the making of many books there is no end and much study wearies the body.

King Solomon, c300BC (Eccl. 12:1)

He did many other things as well. If every one of them were written down, I suppose that even the whole world would not have room for the books that would be written.

St John the Great, cAD100 (Jn. 31:25)

I am a third generation graduate of The University of Glasgow. My father used to talk to me about his studies in the 1930’s, and his late father’s studies in the 1880’s. In some ways little had changed at this ancient seat of learning - we had identical Latin matriculation cards, we matriculated by writing our names in a huge book, we were both classified as Transforthana being born in the Scottish Highlands, and we both have huge vellum graduation scrolls also in Latin (of course). But my father could not understand how I could study Pure Science (as it was then called) without also reading Philosophy. He used to ask, “How can you
possibly understand the nature of what passes as knowledge in your field if you haven’t also studied Philosophy?" In the 40-year gap between our studies much had changed.

Acadaemia is an ancient institution dating back to ancient times in Greece, China and India. The first universities were founded towards the end of the first Christian millennium in Europe and North Africa. Much has changed in the knowledge they have stored, developed and imparted in this time. The discussions presented (see also §2.2, §2.3 and §2.5) about the changing nature of knowledge in the academy relate to a tendency away from knowledge-as-theory towards knowledge-as-practice, encapsulated by Gibbons et al (1994) influential text on the two modalities of knowledge which has captured much of academic curricular analysis and design imagination1.

The rest of this section of the literature review is a consideration of what has been said about this tendency to divide knowledge into two camps which might be called theory and application. This was a significant issue which arose early on in the pilot study and in the early data gathering through interviews (see §§5.1). By examining what has been said on the nature and causes of epistemological change and tensions this section should provide underpinning to a study which is itself concerned with knowledge-in-change in one particular field, university and time as it was perceived to affect a particular group of CS academics.

---

1 This book has been regularly evident in much of the material read for this study. A partial listings of authors who prominently cite it include: Barnett, 2000a; Askling et al, 2001; Bleiklie and Byrkjeflot, 2002; Kivenen and Ristela, 2002; Wensley, 2002; Delanty, 2003; Krucken, 2003; Maranta et al, 2003; Thorlindsson and Vilhjalmsson, 2003; Tight, 2003; Barnett, 2004; Horlick-Jones and Sime, 2004; Sall, 2004; Bleiklie, 2005; Ospina and Dodge, 2005; Starkey and Tempest, 2005; Van Aken, 2005; Carlaw et al, 2006; and, Henkel, 2007.
2.4.2 Changing Knowledge Theory

Man’s capacity to think is his most outstanding attribute. Whoever speaks of man will therefore have to speak at some stage of human knowledge. This is a troublesome prospect.

(Polanyi, 1958, p11)

Above all, the advancement, transmission and exploitation of multiple forms of knowledge ... have become of prime importance not only to nation states, but also to many other interests, including industrial, commercial, voluntary organizations, families and individuals.

(Henkel, 2007)

Polanyi (1967,1969), in my view, best described and explained the nature of the problem that is currently facing academia. He stated that:

what is usually described as knowledge is of two kinds. What is usually described as knowledge, as set out in written words or maps, while unformulated knowledge, such a something we are in the act of doing, is another form of knowledge (Polanyi, 1958, p12).

He called these kinds of knowledge ‘explicit knowledge’ and ‘tacit knowledge’. So Polanyi begins to elucidate the problem faced today by traditional academics such as our CS teachers in this study,

...tacit knowing appears to be a doing of our own, lacking the public, objective, character of explicit knowledge. It would appear therefore to lack the essential quality of knowledge

and,
the essential logical difference between the two kinds of knowledge lies in the fact that we can critically reflect on something explicitly stated, in a way in which we cannot reflect on our tacit awareness of experience.

Is this tacit knowledge therefore of a lesser kind than explicit knowledge? If it is then the knowledge of how to program computers is of less value than a textbook on λ Calculus and Computational Theory, which is clearly absurd. Polanyi (1958, pp14ff) also argued that this knowledge hierarchy could not be true. He compared a map with a journey - interestingly, in much the same way that the CS academics in this study also used similes and metaphors. The map is objective, real, can be discussed, can be explicitly updated, can be created by a second party and is external to us. The journey is personal, experiential, immensely richer than the map, and is the only way to check the accuracy of the map. Indeed, Polanyi (1958, p20) argued that explicit knowledge is useless unless it becomes useful tacit knowledge through the process of understanding.

There is weakness in tacit knowledge: it needs to be created by a process of understanding. Personal experience does not become tacit knowledge unless the person reflects upon the experience in the light of explicit knowledge. So, the skill to create a widget is not of the same order as the understanding of the meaning of the process of the creation of the widget. As one of the interviewees in the study said,

it is because I studied Computational Science at St Andrews University that I have been able to cope with all the changes which have taken place in Computing. People with HND's in Computing I met treated every new language as something totally new, whereas I understood what was going on, so learning the new language was easier for me (see appendix D).

I return to this as a theme in the Findings (chapter 6).

The teaching of mere skills falls well short of what might be thought of as a university education. Becher and Parry (2005) noted that practice-oriented
knowledge is know-how but not know-why; it cannot of itself easily be used to generate guidance, to structure ideas, or to innovate.

Polanyi (1962) also argued against the universal applicability of explicit knowledge stating it could not, on its own, answer the questions we are actually interested in answering, even for highly technical subjects. However, Polanyi's argument is at its weakest when he spoke as a naïve, positivist, modernist with no explanation of the problems of today's post-modern knowledge. He asked 'whether knowledge, admittedly shaped by the knower, can be determined as he thinks fit?', and replies that, 'a passionate search for the correct solution of a task leaves no arbitrary choice open to the seeker' for 'the shaping of knowledge [is] a responsible act, free from subjective predilections. And it endows, by the same token, the results of such acts with a claim to universal validity.' (Polanyi, 1958, p36)

Short (1973) also divided knowledge into two camps, which he called 'production' and 'utilisation'. He talked of the problem of accessing usable knowledge as it is generated post factum. This resonates with Polanyi's discussions on personal and tacit knowledges for knowledge may both not have been generated and/or not be capable of generation. In either way, it is knowledge that exists in the head of someone else rather than being explicitly available in a 3rd party form such as on paper. He also discussed the divide between the practitioner who wishes practical knowledge and advice, and the researcher who can only produce theoretical knowledge without advice. He concluded,

knowledge utilization can no longer be viewed simplistically as a matter of moving knowledge from one place to another. It involves the user's internal knowledge structure, his motivation, his estimate of the source's credibility, the social sanctions that exist for or against use, the availability and appropriate form of knowledge, as well as the existence of adequate linkage agents or systems.

Interestingly, this definition can still be equally applied to theoretical or practical kinds of knowledge in CS and other fields.

Foucault (1969) (see also Foucalt, 2002, p201) defined a field as a 'discursive practice ... a group of rules that are immanent in a practice', defining field knowledge as communicable, debateable, conceptual and usable. Kogan (2005)
saw the two modes of knowledge as being explanands of power: patterns of power and assumptions about power. The applied CS knowledge of the new universities could thus be viewed as being of a lesser kind than the pure, theoretical knowledge of CS in the old universities. Kogan also saw a drift away from internalised knowledge, controlled by academics and intellectuals, towards socially embedded knowledge of a more democratic kind (see also Delanty, 2001). Bleiklie et al (2003) saw this change as a serious threat to such established epistemic centres as Medicine and Education.

Rammert (2004) outlined a wider focus on a series of changes in knowledge over time: from oral to written, from myth to religion, from magic to theory-based reasoning. He continued,

similarly, the media revolution from written to printed knowledge turned secret and local knowing into public and universal knowledge. This change gave rise to modern knowledge regimes, which substituted formal rationality for routine and tradition and from scientific knowledge to experience.

He called the effects which this ‘post-normal science’ of new knowledge has on the old knowledge, as fragmenting and disunifying the latter much as is perceived by CS academics as they lose control of their field as it slips from controlled theory to uncontrolled practice. Boudourides (2003) saw this post-normal science as one that is not controllable by the traditional scientific community, but one that requires ‘civic participation as a policy response to an increasing uncertainty.’ This parallels the rôle of such public cyber-spaces as forums, Usenet and Wikipedia, places of new knowledge creation and storage. He caricatured the old knowledge regimes as being merely ‘puzzle solving practices’ and contrasted them with new knowledge with its uncertain facts, disputed values, high stakes and fast decision-making.

If new universities are places primarily concerned with the transmission of the lower academic status applied knowledge, and the older universities places to contain traditional theoretical knowledge – a broad brush, but quite likely to be true in CS – then both kinds of establishment have their knowledge hegemony threatened by the location and use of new knowledge today. As one of my colleagues said recently, ‘why would I want to own a dictionary – it would be out of date.’
Hildreth and Kimble (2002) are knowledge management scientists who came across at first hand the difficulties in co-handling these two kinds of knowledge. One kind, they said, 'comes in books' and 'can be quantified, counted, organised and measured' and the other kind 'cannot be articulated, abstracted, codified, captured and stored.' Others have struggled when trying to combine the two kinds of knowledge: tacit knowledge is hard to manage (Buckingham-Shum, 1998) and it too can be further subdivided into personal and corporate knowledge (Rulke et al, 1998). However, Leonard and Sensiper (1998) saw not separate kinds of knowledge but a spectrum which ranges from tacit knowledge at one end to explicit knowledge at the other. In an exemplar well understandable to a Computer Scientist, Hildreth and Kimble (2002) explained their idea of a scale of knowledge in proposing we consider human language as an example: some of the knowledge (such as grammar and spelling) is explicitly codifiable, whereas other rules are less so (such as pronunciation) and other almost not at all (such as meaningful grunts.) However, they saw Leonard and Sensiper's idea of a continuum from tacit (soft) to explicit (hard) as inadequate:

both hard and soft knowledge are created and shared. The implication is that it is pointless to seek soft knowledge on its own ... [this] is a false dichotomy. Rather than seeing knowledge as opposites, perhaps we should think of it as consisting of two complementary facets: a duality consisting simultaneously and inextricably of both.

In the academy there has long been a variety of knowledge. Bleiklie (2005a) compared degree studies in the Liberal Arts and Sciences with those in Medicine, Law and Engineering, the former pair providing a 'complete education' with the latter three aiming 'to educate students [for] specific occupations.' However, CS retains both sides of this coin: a strong foundation in CS theory is founded in $\lambda$ Calculus, and it has a strong need to be applied in practice. The very young field of CS has been created in parallel with major changes in knowledge,

by the 1970's the understanding of the changes taking place shifted from information alone to a greater emphasis on knowledge. This occurred in the 1980's and 1990's at a time when the institutional environment was one of deregulation and liberalization. (Carlaw et al, 2006)
This, they added, was also occurring at a time of great technological, economic, political and social changes, such as we are still seeing today.
2.4.3 Changing Knowledge Practice and Use

Jacob (2000a) reported ‘a structural shift in the economies of industrialized countries to the post-industrial or knowledge society phase of development, in which knowledge is the prime motor of economic growth.’ Scott (1995, p101) echoed this belief that ‘knowledge has succeeded energy as the basic resource.’ The university wishing to teach in such an applied knowledge society must seek to understand its practices and structures, ‘as a site of knowledge and as a site of learning’ (McIntyre and Solomon, 2000). One of the difficulties in the de-technologising of the post-industrial workplace and its humanisation is in making people work with technology rather than for it in less rote ways. This involves making the skills and knowledge softer and so less visible to the teacher, learner and assessor (Thorlindsson and Vihjalmsson, 2003). Applied knowledge is created and defined in practice, and it is knowledge which is often uncertain in nature, being constructed and deconstructed continuously (Lidskog et al, 2005) on a range and at a speed not recognizable to the explicit knowledge theorist.

The change to a knowledge-based economy impacts what scientific knowledge is, and so also the rôle of the university and its academics in this society (Jacob, 2000b). Jacob also showed his concern that the creation of mode 2 knowledge, which is by definition practical, requires a commercial partner, one who will inevitably claim some ownership knowledge generated from public funds (Carlow et al, 2006), and who may seek to use the university as a cheap means of accessing high powered minds for its own capitalist ends leaving ‘the university stripped of its broader knowledge mission’ (Starkey and Tempest, 2005). If there is a drift towards mode-2 knowledge in the academy, Rhoades (2005) saw it not as a sudden sectoral or institutional shift, but one that occurs more strongly within certain academic departments and in particular universities.

Creating curricula that support the greater duality of knowledge is a major challenge. Toohey (1999, p40) reported a workshop of academics who reported
such problems in course design as [Toohey's comments are in the open and mine in italic-parentheses]: our students come from a different country than we work in (is it the same kind of knowledge there?), some academics or departments don't want to change (could the field knowledge be rebalanced?), it takes time and effort to do properly (finding out the knowledge structures and mapping them to teaching), what does the new kind of course look like? (there may be no easy exemplars to copy), mixing full-time and part-time staff (there may be a change of power towards applied-knowledge part-time staff (Symes and McIntyre, 2000)), does this help us with existing teaching problems? (such as multi-campus delivery), what do we cut out? (to add more applied knowledge some theoretical knowledge must go.)

Symes and McIntyre wondered 'whether such [applied] courses will ever enjoy parity of esteem with more conventional degrees?' Boud and Symes (2000) noted that 'some of the élite institutions ... are staggeringly untouched by these developments and have resisted modernising trends.' Are these new kinds of applied degrees children of a lesser god to be sneered at for 'putting utility ahead of moral enrichment as the principal goal of education (Symes, 2000)'? Teichler (2005) wished to debunk this hierarchy of the theoretical over the practical education by referring – I believe more hopefully than realistically here - to the Bologna Agreement which does not explicitly refer to the type of knowledge in a degree study and instead explicitly affirms degrees aimed at the labour market.

Contrariwise, some view the traditionalist forces in favour of academic theoretical respectability as having a detrimental affect on the knowledge taught in a field. For some subjects it is believed that it is actually the 'scientization of our field [which] has greatly diminished ... respectability' (Van Aken, 2005). There is a particular problem in theory-laden subjects as to how an academic who wishes to remain relevant and powerful within a traditional explicit-knowledge mode field handles the 'uncertainty that arises out of the sheer multiplication of pertinent evidence or relevant knowledge ... as the libraries of Western universities groan under the weight of volumes and journals, not to mention the arrival of electronic forms of scholarly communications' (Barnett, 2004). We are in a 'knowledge explosion' (McInnis, 2002) that is itself creating field fragmentation which pressurises 'an official curriculum that is fully packed' (Kivinen and Ristela, 2002) to select or diminish taught content, made difficult by the loss of established anchors to attach to changing knowledge (McInnis, 2002).
Standing still cannot be an option, even if the move to tacit/personal knowledge is also deemed not to be an acceptable option. Much of this debate is recognisable to the Computer Scientist who must teach his students using real programming tools, for an ever wider range of applications, whilst trying to hold onto underpinning teaching that is perhaps – although there is no absolute agreement as to what this is in the academy – ever important.

Middlehurst and Barnett (1994) had similar concerns over how new knowledge affects the structure of academic departments and over who drives these changes: academics, students, employers, managers, professional bodies, etc. Barnett (2006c) later asked ‘can such forms of knowing be securely assessed?’ CS has been in this quandary for some time. Computer Scientists can assess explicit knowledge (e.g. ‘are there are 8 bits in a byte’) and tacit knowledge (e.g. ‘can you write a program to play pac-man?’). However, personal knowledge (e.g. ‘how did you use your mind to write that program?’) or practices (e.g. ‘how did you use the SDK?’) cannot be easily expressed or assessed.

The wider academy continues to grow to include such as ‘catering, nursing, hotel management, health, tourism and leisure studies’ (Symes, 2000), all naturally-occurring applied knowledge domains. The nature of assessment in such newer fields is quite different from subjects such as Physics, Mathematics and Anatomy as ‘one cannot understand knowledge development without understanding the differentiated cultural machinery in which the disciplines are embedded’ (Bleiklie and Byrkjeflot, 2002). This is an increasing challenge for university Registry regimes with their mind-sets of tick-boxes for assessment rules/marking and curriculum definition/delivery forms.

In these applied knowledge fields there are new kinds of criteria to be assessed: judgment (Hagar, 2000), critical decision-making (Horlick-Jones and Sime, 2004), presentation skills, teamwork and ethical behaviour (McLoughlin, 2004). To these Tennant (2000) would add that the tacit and personal knowledge to be assessed is generated by the learner and not to an easily pre-defined script from the teacher. There is also the issue that this knowledge is not produced and consumed within traditional university structures, whether physical or organizational (Usher, 2000). This can be seen in university CS teaching which is often created, located and
delivered outside the traditional real campus using such means as open source and cloud computing.

Further thought may also be required in student selection for degree programmes (Hagar, 2000). Perhaps traditional academic SQA Highers or Edexcel A-Levels may not be a best-fit for a more applied, tacit-knowledge based degree programme. In a way this debate may already be too late, as 'if we do not acknowledge the habits that students have already acquired, all attempts to redirect their actions' - perhaps to traditional explicit knowledge studies - 'to make them learn, are likely to fail' (Dewey, 1980, p30). There is a close echo of this problem, perhaps, in CS which, unlike Mathematics, Chemistry and Physics, rarely requires any entry qualifications in the subject from school studies, indeed even rejecting such awards as Higher Computing as being irrelevant.

A major challenge in teaching in an in-flux knowledge field is also working with monolithic, anti-innovative, non-responsive and Taylorist university management structures (Symes and McIntyre, 2000) which are designed to stifle and control change precisely when there is a need for more flexible provision and less institutional conservatism (Taylor, 1999, p155). PG Taylor saw the corporatist university as a hopeless case and called for academics to strike out alone as self-interested activists as some in CS do. This is a message mirrored by Smith and Webster (1997) who called for the academy to distrust the sectoral directing of 'philistine' politicians.

Such anti-Taylorist outbursts could be seen as a result of a classic stress situation as the inability to respond to university managerialism combines with the 'changing nature of knowledge domains over time' (Tight, 2003) trapping the knowledge worker in the middle. There is hope, though, as, as Tight continued, 'the vocational role of the university has always, of course, been strong.' It can be a relief to the harried applied CS academic to think that 'higher education should aim to a lesser extent at providing a certain body of knowledge' (Teichler and Kehm, 1995) as such knowledge becomes increasingly difficult to source and grasp. Jacob (2000a) noted that governments prefer universities to be involved in mode 2 knowledge production as this is a more clearly identifiable product for stakeholders. Involving industry in knowledge production also saves the government up-front money easing 'the burden of the universities on the public purse' (Jacob, 2000b).
Rip (2000) warned about getting stuck in a small understanding of what is going on, such as the current mode-1-versus-mode-2 of Gibbons et al (1994). This may be a fashion which may lock-in our thinking in a wrong way as ‘the world may well evolve in such a way that present-day exemplars are left behind’. Barnett (2004) warned that any solution proposed is itself time-limited and perhaps even simply expeditious as the problem is at a meta-level of increasing complexity, where even skills plus knowledge may not be enough in a super-complex (Barnett, 2000a) world but skills in handling change may be the most pertinent. It is interesting to note that the mode-1 versus mode-2 debate may itself be stultifying and mode-1 in nature, reflecting old ways of thinking whereas, to Delanty (2001) this debate is itself an opportunity for a university’s traditional core critical analysis and discourse skills.

In the non-intellectual world, where much of the tacit and personal knowledge is deployed, it is free-range and roaming, ‘rarely subjected to serious academic scrutiny’ even defying academic classification in having its own labels (Jamison, 2003), part of the knowledge belonging to ‘a steadily more informed and better educated public’ (Blieklie and Byrkjeflot, 2002) in a world of contested ‘knowledge politics’ (Healy, 2003). Maranta et al (2003) challenged the established notion put forward by the traditional theoreticians that lay people are ignorant because they lack theoretical knowledge. Lay expertise turns out, they said, to be a ‘thoroughly demanding setting for the experts after closer inspection.’ However, they acknowledge that there is an ‘epistemic divide’ between experts and lay people and so they caution against involving lay people in defining academic studies. Lay people act in a different way from the academic as the former are ever active agents and so not skilled as the experts are to step back and reflect upon their knowledge.

And so, it can be seen that a field’s knowledge is heterogenous in form and plural in type. Society has expectations for and from this knowledge, with its ownership and creation, and is showing a wider will that sites of knowledge creation should be more open and public, driving acadaemia closer to wider society (Krucken, 2003), calling upon academics to recognise that ‘knowledge is increasingly produced in the context of application by hybrid communities working on multiple sites, by no means all of which are academic institutions’ (Henkel, 2005).
Perhaps the university can remain that place with 'a major role to play in knowledge generation and reconfiguration by providing a meeting place in which the different discourses of business and society can confront each other' (Starkey and Tempest, 2005) and where the knowledge generated by the professions, vocations, industry and community groups is legitimated (Becher and Parry, 2005). However, if the university, in a pro-theory hissy fit abdicates this rôle then, (Hall, 1998) reminded us that there are other places where this can take place, and perhaps not just in the real world places which he envisaged, but perhaps also in the virtual worlds of websites including Google, MySpace, Messenger, Facebook, Second Life, Wikipedia, etc. It is perhaps the university which seeks to hold onto that which it has already lost which has most to lose further. It is the one which recognizes that knowledge is created and engaged with by a multiplicity of partners which can engage with wider expectations of knowledge definition, creation, validation and use.
2.4.4 Conclusions

Let’s work without thinking about it. It is the only way to make life bearable

*Voltaire*

The academic, or the many others working in today’s ever-changing, expanding and developing knowledge economy, could be forgiven for emulating Voltaire. However, Polanyi made it clear that working is thinking, and, thinking is working. Knowledge is varied, eclectic, changing, partially-knowable, time-based, itinerant, never fully known and rarely fully unknown. And that is the joy of being an academic in any field, even if the unknowability is also our pain.

There is a great deal of consideration in the universities, government and industry over the nature of the knowledge economy, knowledge production, knowledge instability, knowledge ownership and the very nature of that knowledge and where it is held. At the same time academics, professionals and the wider public are engulfed in a tsunami of knowledge, whether explicit (through the ever increasing number of books, journals, etc.) or personal (through websites, television channels, Internet chat rooms, email, txt, etc.)

For some fields within some institutions there may be greater degrees of possibility of ignoring this knowledge revolution and to seek to stand still. However, students incoming knowledge, experiences and expectations are moulded by the knowledge society they are formed in and may not find this looking back acceptable. Some fields will wither as their relevance to new knowledge is challenged, as did my father’s Philosophy. Still other fields will need to bite this bullet and seek ways of teaching and researching in new, applied ways. As will be seen later in this study, CS is at the heart of this storm.

Over time some parts of our knowledge and our knowledge structures will be lost and others will be gained. Possibly more theory will be lost from academia as the
pressure to orientate studies towards applications increases. Some subjects will
disappear as new ones appear. Some academics, departments and institutions will
rise to the challenge and others will fade away. Governments will continue to
pressure the universities to expand their teaching to wider groups in the
community, and to be more graduate-employment orientated. Industry will also
want better graduates and access to cheap, smart researchers. CS is no exception
to these forces and changes.

Whatever happens, it will be more change for knowledge workers - academics and
practitioners - as they seek to remain relevant and in touch with the modern
knowledge economy.
2.5 Pressures on Subjects

2.5.1 Introduction

The growth of knowledge and the consequent explosive growth in disciplines and their fragmentation into sub-disciplines is probably the most important, but often-overlooked, change affecting HE in recent years, adding to its complexity and proving more powerful in its effects [than] even the system’s ‘massification’ and marketization.

(Becher and Trowler, 2001, p14)

My experience of the changes in academic studies is focussed by what my own children studied for their degree awards at university in the 1990’s and to date: Contemporary History and Politics, Environmental Biogeochemistry, Criminology, Waste Management, and, Physiotherapy; my younger daughter is considering reading for a BSc in Midwifery. My studies in Computer Science and my wife’s in Social Sciences in the 1970’s were then novelties. As, quite likely, were my grandfather’s MbChB studies in the 1880’s. Perhaps Becher and Trowler (in the opening quote to this section, above) were being short-sighted in not seeing the river of epistemic change that ever flows through our universities.
CS is one of many subjects that are taught and researched in universities. It has had, thus far, a relatively short life-span, dating only from the mid-twentieth century at earliest, and appearing as a major university subject only in the 1980's. It would be useful to consider how CS relates to other academic fields which are under pressure and to seek to understand from where these pressures emanate. It is a world where many academic fields struggle with the new world of exploding, restructuring and reprioritizing knowledge.

Three particular fields are considered here: Management Science (like CS, another relative newcomer to academic life in the UK), the Environmental Sciences (again, like CS, a combination of several different sciences and philosophies under a single heading) and, first, the general case of the emergence of multidisciplinary, cross-disciplinary and trans-disciplinary studies in universities (such as CS has been purported to be).
2.5.2 Multidisciplinary, Transdisciplinary, Cross-Disciplinary

BSc Environmental Biogeochemistry focuses on the chemical and earth science aspects of the natural environment with an emphasis on the role of water and its interaction with rocks, soil and living organisms.

(University of Glasgow, 2007)

We explore the current definitions of science, ... homeopathy, continental drift, autism, MMR vaccinations, Gulf War syndrome and the Loch Ness Monster

(University of St Andrews, 2007)

I have taken Horlik-Jones and Sime’s (2003) line that the three terms used in the title above are generic and ambiguous, and so can be taken as having a single meaning. All are pertinent to the case of CS as a multidisciplinary / transdisciplinary / cross-disciplinary field. For example my own area of Computer Games includes such teaching as programming, mathematics, physics, design, audio, camerawork, narrative, characterization, enterprise and marketing in order to graduate a technical C++ programmer skilled to create video games. This combination of skills is validated by Skillset (skillset.org.uk) which is established and run by the major UK video games developers.

Subjects combine in related, cross-disciplinary ways, indeed, ‘knowledge is generated in the context of application. It is transdisciplinary’ (Askling et al, 2001). Lidskog et al (2005) noted the long-established overlaps of regulation (Law), food
(Biochemistry), toxic chemicals (Inorganic Chemistry) and health risks (Medicine) when the mass media, the public and politicians consider Health.

Transdisciplinarity is not always easy to handle, as they acknowledge when they continue to discuss the as-yet unresolved issues relating to mobile phone transmitter safety. Hostaker and Vabo also noted how even powerful and established fields can come under public pressure to loosen their grip on what academics wish to see as their knowledge, 'the knowledge-monopoly of the medical profession has been challenged due to the easy access for the public to medical information on the World Wide Web' (Hostaker and Vabo, 2005).

There is a real danger in over-stepping competence boundaries in a field of study that crosses known epistemological boundaries, as I am aware as I write this section of the study. Becher (1989) believed that there is a close coupling of the discipline and the academic, where the academic is nurtured and developed within his academic field. He saw the internal validating processes of peer review and validation as critically important. Bourdieu (1977) said, there is a 'struggle for scientific authority' where a scientist,

cannot expect recognition of the value of his products ("reputation", "prestige", "authority", "competence", etc.) from anyone except other producers [field-specific scientists] who ... are the least inclined to grant recognition without discussion and scrutiny.

Traditionally, a field is validated by its experts, who assure the provenance of knowledge by careful scrutiny against their individual and combined knowledge in the field. Bourdieu continued, 'this is true de facto: only scientists working in an area have the means of symbolically appropriating his [the other scientist's] work and assessing its merits.' The real danger for an academic acting outwith his traditional area of field knowledge in a transdisciplinary emerging field is that, 'this is also true de jure; the scientist who appeals to an authority outside the field cannot fail to incur discredit.' This acts as a double-edged sword: the scientist cannot make claims for areas that fall outwith his field of competence, nor can such claims be cited by others.

Yet, the drive towards multiple-knowledge approaches is being powered by the expansion of the application and understanding of science in hybrid communities
(Henkel, 2007). Those who work in these transdisciplinary fields, such as solution-designers can see themselves as having “a knowledge-base [which] is eclectic [multi-sourced]” (Cliff and Woodward, 2004) and they are often comfortable in the eclecticism of handling many different forms of knowledge (Henkel, 2007). But, is this comfort zone justifiable?

Nowotny et al (2001, p115), in discussing the problem of researching the field of AIDS vaccines stated, ‘choices emerge in the course of a project because of many different factors, scientific, economic, political, even cultural’. In this there are what Thorlindsson et al (2003) called ‘boundary challenges’ between such different academic bedfellows as Biomedical Sciences, Technology, Sociology, Anthropology and Ethics.

The academic field of Chemistry has had to react to ‘the push for relevance’ which ‘will work out differently for different fields and disciplines’ (Rip and Eijkel, 2004). Henkel (2005) concluded that this has had tragic consequences for the breadth of academic studies as ‘pure of foundational disciplines, such as physics, chemistry and economics, may be less well represented in the basic units across the range of higher education institutions’.

It is not all loss, however. Leydesdorff (2006) said that the nature of transdisciplinary cooperation is also expansive as it, ‘provides us with more choices than hitherto.’ Carlaw et al (2006) were positive about the newnesses that transdisciplinarity creates and, when taken in combination with global capitalism, there are significant driving forces in this cross-disciplinary movement as ‘the economic value of the individual pieces of knowledge is only generated when they are combined with other pieces of knowledge to form commercially valuable products and services.’

Whereas fields such as Physics and Chemistry are long-established, this is not so for the Management Sciences or Computing Science. The short-timescale nature of interdisciplinary fields is indicated in the Rip and Eijkel (2004) paper on research centres. In new cross-over fields they also noted that for significant British and American interdisciplinary research centres a ‘key point is that they are time-limited in terms of funding.’ It follows that these are not fields that are as deep as they are broad. Bleiklie and Byrkjeflot (2002) could only use the word ‘thematic’ to
describe how these centres congregate around a common topic. Kogan (2005) wrote of these new themes as being ‘domain-based or policy-oriented’ and ‘essentially interdisciplinary’ involving ‘the crossing of new boundaries and the creation of new syntheses [which] may advance both knowledge and human betterment.’

The trans-disciplinary nature of new knowledge is a real challenge to the academic community. Not only does it combine different forms of academic knowledge, but it also absorbs forms of knowledge more traditionally found outwith the academic domain, for example, from where it is located in workplaces and local communities (Winberg, 2006). These kinds of knowledge have to be combined and a sensible structure put upon them, as Winberg (2006) put it,

emergent transdisciplinary curricula structure processes of interaction within overarching frameworks that organise knowledge in new ways, address new problems, and develop new discursive practices, with the co-operation of diverse sectors of society.

To Horlick-Jones and Sime (2003) this transdisciplinarity is inherently unstable and unpredictable. It is ever a ‘living on the border’, a place of conflicting ownerships and identities. Here there is also a need to spend less time creating theory and more time interfacing with the knowledge of practice. They see the discussion of mode 1 versus mode 2 knowledge (Gibbons et al, 1994) as being descriptive and analytic rather than creative; it describes what is happening, not how it happens. The creative force is the need - or ‘crisis’- to create better solutions in a highly complex world. They even called for a new way of thinking that goes beyond the traditional bounds of the domains of natural and social sciences, caricaturing both meta-viewpoints as being too reductionist.
2.5.3 Management Sciences

Management Science covers a wide range of subjects. This section of the study will consider a few of these in an attempt to gain an understanding of the wider issues affecting this overarching field and attempt to draw some parallels with the situation in CS.

Accountancy is inherently a field of application, driven by the needs of and developments in the workplace (Jones and Abraham, 2007). The changing nature of Accountancy, driven by new corporate and legal-political realities has necessitated change in the training of new entrants to the profession (Siegel, 2000; Simpson, 2000). This series of changes has been hard for universities to keep up with, with some smaller and/or newer universities recently giving up the struggle and dropping the once-popular subject from undergraduate studies. The field of Accountancy is one which is naturally multi-disciplinary, covering such areas as Information Technology, Finance, Ethics, Law, Anthropology, Sociology, Politics and Economics (Jones and Abraham, 2007).

Public Administration is a field that continues to develop. Ospina and Dodge (2005) saw it as one deeply involved in the search for connectedness between acadaemia and practice in a changing landscape. The field, like Accountancy, has a natural pluralism with which it is struggling as its academics try hard to be both academically sound (through traditional scholarship and publication) and practitioner-connected (through applied relevance and rigour.) They noted how traditional academics take an approach which naturally and uncomfortably lifts the researcher above the practitioner, and so makes the connections between theory and practice difficult to make in such an academic organisation. Academics in Public Administration also have to cope with a field of practice that is becoming increasingly specialised, and balance this with a tendency to want to ape the approaches of established academic disciplines in a search for scholarly respectability. The authors cited case studies where academics and practitioners
have come to a total impasse over approaches to knowledge production, in what Ospina and Dodge see as yet another mode 1 versus mode 2 knowledge battle. They concluded that there was a 'gap between academic researchers and practitioners [which itself] stems from a lack of appreciation for the other's perspective' and conclude sadly that 'there are no bridges to connect the two worlds’ in this field.

Figure 2.2 - an example of the popular business press, or knowledge which exists outside academia (Anon, 2007)

Wensley (2002) considered the case of Marketing as a subfield in the Management Science academy. He started in much the same vein as Ospina and Dodge (2005) had concluded:

much of the discussion about the relationship between marketing academe and practice assumes that there is a wide, and indeed sometimes widening, gap between these two domains which is itself fraught with problems.
He called for a building of new bridges between the two opposing sides - academia and practice - of Marketing. He also noted instances of mud-throwing between the two sides, with the academics being accused by the practitioners of having an obsession with ‘journals of obscure analysis’ while practitioners are slated by academics for having a practice that repeats known errors and pays no attention to theory. In a field that is so closely connected to capitalism, Wensley (2002) noted the dichotomy between the individual in practice for personal profit, and the academic in research purely for the greater public good. He discussed other instances of ontological clashes between the two communities and argued for a new way of thinking that would allow a shared handling of the dichotomous nature of the field, which he deemed is essential if it is to have any chance of surviving as the same field in academia and practice. If this doesn’t come about, it is possible to perceive a similar outcome to the one described by Kogan (2005) where years of battles within the applied and theoretical Social Sciences have caused politicians (as applied Social Scientists) and academics to become almost entirely unstuck.

Van Aken (2005) considered the more general case of Management Science in the academy and called for a combination of validity (academic) and relevance (practice) particularly for the good of the students who pass through the academy. This is, he confessed, ‘problematic’ and ‘a fairly old problem.’ Again, it is Gibbons et al (1994) who have either precipitated or focussed the debate as being one of battles in and around modes of knowledge. Van Aken (2005) called for Management Science to abandon its obsession with mode 1 knowledge and to move to mode 2 knowledge, citing Medicine, Engineering, Natural Sciences and Sociology as exemplars of fields which have done this successfully. However, the discussion below on Environmental Sciences, (see §2.5.4) shows this to be a highly simplistic proposal.

Van Aken went on to list refereed work since 1978 which criticised academic activity in the field of Management Science. He proposed, firstly, better communication of good research from the academy to practitioners, secondly, to make research more practically-based, and thirdly, to entirely embrace mode-2-ism and abandon all theory in favour of short-term practical applications. In doing so he echoed the Environmental Scientist, Healy, who ridiculed the traditionalist academy and called for the precedence of mode 2 ‘context, process and procedure’ over mode 1 ‘metaphysical or ontological abstractions’ (Healy, 2003). Van Aken seriously
proposed Healy’s option. He would propose steering Management Science away from heavy theory as he thought it was simply not ready to combine both the mode 1 and mode 2 routes as the established fields of Medicine and Engineering do. To him this is route which is both too difficult to take for Management Science and of doubtful relevance to the field.

Universities tend to bring the Management Sciences, and increasingly other related fields such as Information Technology, together under the umbrella of a Business School. Bolton (2001) saw the Business School as a ‘ cuckoo in the nest’ of academe, a centre that does not want to be part of the enclosed university community which it, the Business School, sees as inherently statist and shackled. He said,

  the challenges for business schools is so great that, in order to succeed
  ... they need incentives and a measure of autonomy to achieve
  improvements in staffing, services and infrastructure.

In response, wider acadaemia often views Business Schools as ‘shallow ... epitomised by non-conceptual texts such as [The] One Minute Manager’ (Blanchard and Johnson, 1983), where business is seen as a ‘theme’ rather than a ‘field’. Bolton noted that this relationship became so fraught in the early 1990’s that the then right-wing UK government considered passing legislation to divorce business schools entirely from universities.

Starkey and Tempest (2005) also considered the case of the Business School. They concluded that the current instantiation of the business school is ‘fatally compromised’. They saw the problem in it being situated in the privileged knowledge-space of the university, which accredits its biggest degree programme, the MBA, whilst practice does not value this qualification which, they stated, has not had any significant impact upon management practices. As Pfeffer and Fong (2002) damningly put it, Business Schools are, ‘teaching the wrong things in the wrong ways to the wrong people ... at the wrong time in their careers.’ Starkey and Tempest’s (2005) criticism continued, describing the academic Business School as being an incestuous closed loop, hermetically and hermeneutically divorced from the real world, amoral and corrosive. They were not against the existence of Business Schools, but saw a real danger of the current university-based Business
School becoming marginalised and increasingly contested in the eyes of real business by the provision of more relevant-to-them private training. In a kind of masochism they even criticise their own academic work and co-workers in their field for helping create the managerialist nightmare that is modern acadaemia. They offer little in way of respite, except a hope that, again, mode 2 knowledge production will become the future of the business academy.
2.5.4 Environmental Sciences

I am indebted to my younger son, Dr. Malcolm Alexander Sutherland, BSc (Glasgow)(upper second class honours), PgDip Waste Management (Abertay)(with distinction), PhD (Dundee) for his help and advice in understanding the issues that are affecting the Environmental Sciences.

The task is to make visible the invisible, to expose to public scrutiny the assumptions, values and visions that drive science.

(Wilsdon and Willis, 2004)

Few, if any, fields are in such a high public profile as the Environmental Sciences. Lidskog (2005) noted the highly politicized and public nature of the field which often acts as a focal point in the passing of 'the warning bells' from scientists, via the media to the wider public. The BBC News website contains many journalistic illustrations of the kind of Environmental Science related stories which catch the public imagination, with titles such as: 'Icebergs are Ecological Hotspots' (BBC News, 2007), 'Climate Fears for Heritage Sites' (BBC News, 2007) and 'Sir Edmond [Hillary] Urges Climate Care' (BBC News, 2005).

In the first of these reported studies, 'Icebergs are Ecological Hotspots', there were academic field interests expressed from as wide a range as Meteorology, Microbiology, Biochemistry, Inorganic Chemistry, Organic Chemistry, Photography, Space Science, Ornithology and Oceanography. There was a modern, almost
The mantra-like feel to the embedded quote, 'The number of icebergs found in waters around Antarctica had increased in recent decades as a result of global warming, the researchers wrote in the paper' (author's italics.)


A Demos applied Environmental Science report for the UK government, 'See-through Science' (Willson and Willis, 2004), covered a similarly wide range of subject areas. In just one exemplar paragraph a vast range of subjects were covered (see table 2.1.)
<table>
<thead>
<tr>
<th>quote</th>
<th>academic field(s)</th>
</tr>
</thead>
</table>
| 'Technological innovation [a] is vital both to the economy [b] and to the environment [c]. Innovation [d] drives much of our economic growth [e]. And new technologies can deliver solutions to environmental problems. Devices to save water [f] will protect river catchments [g] and sensitive ecosystems [h]. Biological waste treatments and soil remediation [i] can reduce pressure on landfill sites [j]. And hydrogen fuel cells [k] will cut air pollution [l].' | [a] Engineering, Computing Science  
[b] Macro-Economics  
[c] Sociology, Biology  
[d] Management Science  
[e] Politics  
[f] Civil Engineering, Meteorology  
[g] Hydrology  
[h] Biology  
[i] Soil Science  
[j] Local Politics  
[k] Inorganic Chemistry  
[l] Organic Chemistry |

Figure 2.1 shows what is a very worrying trend in the Environmental Sciences to some Scientists working within and outwith the field: the use of definitive forms of verbs, showing an unscientific and unsupportably high degree of confidence in findings and in proposals. Despite this being such a multi-disciplinary report, covering over a dozen fields in just over 50 words, such definitives were used as: 'is vital', 'innovation drives', 'can deliver', 'will help us', 'will protect' and 'will cut'. There is only one verb proposed with caution, 'can reduce.' This echoes the
concern of (Bleiklie and Byrkjofot, 2002) that in these cross-disciplinary areas 'the positivistic belief in the scientific method, in knowledge as procedure has weakened' and they further quoted Feyerabend's 'anything goes' statement.

Jamison (2003) discussed the appearance of new forms of knowledge used in the field of the Environmental Sciences which have originated entirely outside the academy: those of the environmental activist, such as 'Green Knowledge'. These new kinds of knowledge are to be found in areas noted for being more bound to communities, professionals, militants and the individual as a person. He added, 'there could be more' kinds and locations of knowledge in the Environmental Sciences. He also accepted the existence of a Theological line in this knowledge, identifying some kinds knowledge as being 'secular ... [focussed] on changing policies and political decisions rather than on changing beliefs' and also that some activists retain 'something of the spiritual' in their activities, promulgating 'mystical teachings of new age philosophy.'

There is also a problematic political sphere (Kogan, 2005) in this new Environmental Science knowledge, which, as Jamison (2003) noted, creates clashes between such extremes as the Neo-liberals and militant NGO's. Political militancy in an academic field and the knowledge changes created have also challenged the sometimes longer-established NGO's traditionally associated with the academy, of which there are many in the Environmental Sciences. Jamison concluded that the field is now one which is 'filled with ambiguities'. Notably, in an echo of the above criticisms of academic Management Scientists made by external practitioners (see §2.5.3), he identified the universities with those opposed to radical Green Knowledge, as being allied to Capitalists, and so creating a similarly deep epistemological (if not already ontological) divide within the Environmental Sciences as appears already to exist in the Management Sciences.
the specialised, subject-based curriculum is hopelessly out of date, still maintained by powerful groups, but passed over by changes in the economy, technology and contemporary employment

(Ahier, 2000)

It is hard not to conclude that many subjects in the academy are finding it difficult to deal with the changing nature, source and location of knowledge. Traditionally, academics have been careful, peer-reviewed and field-specific. The new knowledge is quick, applied and crosses traditional field boundaries. Whilst Environmental Science has actively embraced this new knowledge and its loci of creation and use, this has been at a price of scientific standing and the independent direction of the academy free from the will of outside forces. But, few would doubt that, contrariwise, as a result the field has prospered on many other measures.

On the other hand, the Management Sciences appear to be 'not drowning, but waving' (Smith, 1972). They are caught in a major pull between the perceived need for respectability within the established wider world of Higher Education, and the need for relevance in the world of business. It is almost possible to hear the sinews breaking as the university-bound wider field and its subfields struggle for coherence and meaning from an applied world that is rushing away from the taught and researched world of the academy. It is difficult not to agree with Starkey and Tempest (2005) of the need to do something, even totally embrace mode-2-ism, the Management Scientists then could be as radical as the Environmental Scientists and, throw off everything that hinders and run the race with perseverance.
Academics themselves are divided over major changes to epistemological boundaries and content in their field. Becher (1989, pp68-9) noted that some ‘considered this a healthy feature ... allowing individuals to move in new directions and ensuring ideas do not stagnate.’ Others took a far less positive view, deploring the tendency of some colleagues to ‘behave like a flock of lemmings’ in pursuit of novelty, noting the adverse effects on the systematic development of the field’. It would seem that the negative academics are tilting at windmills as the issue is not their fellow academics so much as inherent field instability caused by fundamental forces also affecting governments and education systems (Hill, 1997).

CS as a field within the academy also competes for its knowledge in terms of kind, location, renewal, curriculum, ownership and research with a wide and broad range of business and lay communities. In this struggle the CS field academics also engage with (both positively and negatively) other academics, university management and politicians. However, unlike the Management Scientists CS has not disengaged with its applied arm. And, unlike the Environmental Scientists CS still fights to retain its roots in inherited academic good practice. For a new field it is, comparatively, doing rather better as an academic field, even if it does continue to be stressed by great pressures.
3. Computing Science Studies at Scottish Universities

Computer Science is Mathematics and Logic ...

Computer Science is Engineering ...

Computer Science is Communication ...

Computer Science is Interdisciplinary ...

The Problems solved by computer scientists come from a variety of disciplines — mathematics, physics, chemistry, biology, geology, economics, business, engineering, linguistics & psychology, to name a few ...

A computer scientist must be knowledgeable about much more than just the computer ...

The successful computer scientist must be able to communicate, to learn new ideas quickly, and to adapt to ever-changing conditions.


Having discussed how the academic literature considers the general background to this case, I now want to look specifically into the situation at the Scottish universities in order to place the site of the study in a clear, real-world setting.
3.1 Introduction

In the introductory quotation (previous page) taken from their set book, 'Introduction to Computer Science', Nance and Naps capture the breadth of studies that is Computing Science (CS). This chapter presents evidence of the changing nature of CS as a taught subject in the academy. It is grounded in data gathered about Scottish universities, plus relevant data on British (i.e. the entire UK) CS studies, with reference to what American (USA) and Australian institutions consider CS to be. The development of computer hardware, software, software development and applications are also considered briefly to help the non-CS reader understand some of the concepts of CS.

The core of the chapter is an investigation of the changing offerings of, primarily, undergraduate CS degree programmes at five Scottish universities¹: The University of Abertay Dundee, Glasgow Caledonian University, The University of Dundee, The University of Glasgow and The University of St Andrews². This is an evaluation of the teaching of CS in Scotland over the past forty years, as it was presented in undergraduate prospectuses, from when the subject began to appear in these universities’ prospectuses.

The chapter also aims to lay a foundation of understanding of the taught field of CS which, firstly, will allow the reader to place this case study in relation to the more general area of taught CS, and, secondly, to allow an informed analysis, later in this study, of the discussions with those in the particular CS department on their coping strategies with this ever changing discipline, CS.

---

¹ For justification of this selection, please refer to §5.2

² For brevity, these universities will be referred to below as Abertay, Caledonian, Dundee, Glasgow and St Andrews
3.2 The Developing Discipline

Computing Science (CS) as a discipline can be viewed as dating back to three phases of development during the early and mid 20th century (Guinee, 2006; Cringely, 2006). The subject roots can be understood as growing out of Philosophy (logic structures and abstraction) and Mathematics (problem definition and solving), as defined by Von Neumann, Turing, Church, et al (Smith, 2004; Copeland, 2004). The first electronic computers were developed during World War II, principally by the Americans and British, for defence applications using developments in analogue electronic engineering (radar) (Copeland, 2006) and digital mechanical engineering (data encryption and code-breaking) (Bletchley Park National Codes Centre, 2006). As a digital electronic engineering discipline CS grew in the 1950s from the work of the US Department of Defense’s COBOL (Sammet, 2000) and FORTRAN (Backus, 1998), IBM’s PL/1 (en.wikipedia.org/wiki/PL/1) and OS/360 (Jardine, 1972) and The University of Manchester’s Leo I (leo-computers.org.uk) projects.

Computer science can be defined as the body of knowledge concerned with the use of the digital computer (Gilmour, 2003). The digital computer itself has changed very little since the 1960s. It is still a stored program2 device that processes a data stream through the central processing unit (CPU). Data are held in binary digits (bits) either in fast, volatile memory (usually called RAM) that is lost when the program stops, or on slow, non-volatile memory (e.g. disks, tapes) that continue to exist after the program stops. This continues to be true of such devices as the

---

1 for brevity, all URLs have been shortened to remove http://www. or http://prefixes. All inline web references are to hypertext markup pages (http).

2 In the UK CS keywords are generally spelled in American English, so ‘disk’ rather than ‘disc’ and ‘program’ rather than ‘programme’. 85
Sony Playstation 3¹, the Apple iMac, a Nokia mobile phone, or the Cray supercomputer in the National e-Science Centre.

Much else of CS, including the creation of programs, has been in an almost constant state of upheaval. The earliest digital computers were programmed directly in the binary language of the computer’s CPU (e.g. ‘10011011’). In the mid-1950s these were translated into languages (assemblers) that stated the meanings of the binary words (e.g. ‘LOAD R1 1011’). By 1958 the US Department of Defense had defined two new languages that were non-computer specific for mathematical formulae (FORTRAN, the formula translator) and data information (COBOL, the common business oriented language) processing purposes. The 1980s saw a proliferation of BASIC language dialects and database languages (e.g. Powerhouse (www.cognos.com), Oracle (www.oracle.com)). Programming has now become a sub-field dominated by languages created and controlled by commercial giants, such as Microsoft’s C# (msdn.microsoft.com/vcsharp), Sun’s Java (java.sun.com) and Macromedia’s Flash (macromedia.com). In specific application areas, such as video gaming, programming is now almost entirely controlled and driven by major commercial companies that include Sony (scei.co.jp), Nintendo (nintendo.com) and Electronic Arts (ea.com) through their ownership of games engines (which contain 90% of the code required to create specific kinds of video games) and development kits (which exclusively prepare programs to run on specific consoles, such as the Xbox 360 or Playstation 3.)

The changeableness of programming languages, operating systems and development and target hardware together have proved to be a major challenge for Computer Scientists. In the past new programming languages were often devised by the academy, such as the 1980's language, PS-Algol at St Andrews and Edinburgh (wikipedia.org/wiki/PS-Algol) which the author used in his research masters degree at St Andrews. Such programming environments as the OpenGL graphics kit (opengl.org) were defined by market-independent bodies, as was the control of changes to standard programming languages, by such as the American National Standards Institute (see, e.g., legacy.com/cobol). Academics once had a clear input to developments in the field, for example, defining the fundamental

¹ All trademarks used in this document are acknowledged
workings of the computer network in the late 1970s and early '80s via the Cambridge Ring project (en.wikipedia.org/wiki/Cambridge_Ring).

However, the power and money that drive CS as an application field have now moved to commerce from academia. CS academics today find themselves responding to and reflecting upon external changes to programming, hardware and applications rather than invoking them.

Although the computer has not changed in its primary architecture (CPU + storage + data + program), its uses have changed enormously over the past fifty years. Today’s typical customer-facing applications can be purchased in the High Street as the home computer, video game console, mobile phone, Internet and personal digital music player, even integrated into a single product (e.g. blackberry.com). Similarly, the computer as a hidden device is central to white goods (e.g. washing machines and microwave ovens), communications (e.g. telephones and cable TV), industry (e.g. retail and defence), black goods (e.g. DVD players and cameras) and government (e.g. tax collection and immigration control).

In the personal sphere the computer has grown as a visible lifestyle device, from the home and games computer of the 1980s, through the mobile phone in the 1990s, to the entertainment-to-go iPod of the 21st century. New software applications (programs) have also changed how people, groups and societies interact through the use of such applications as e-mail, on-line chatrooms, blogs and texting (see, for example, wosgamers.ning.com.)

It is against this backdrop of an ever-changing field that those working within the academic discipline of CS must seek to understand and teach.
3.3 The Computer Science Faculty and Students

As the field of CS has grown (Cart, 2001) along with student numbers and in accordance with UK Government funding policy, so faculty numbers have grown too. This growth was most evident in the late 1980s to late 1990s. Initiatives, such as the new blood scheme for the Scottish Central Institutions (CI’s)\(^1\), the continued open-ended funding of students on CS undergraduate degree programmes (see figure 3.1), and the until recently unlimited funding of students on taught postgraduate Information Technology and Software Engineering programmes (Student Awards Agency for Scotland, 2006), combined to produce a boom in faculty numbers that was paralleled elsewhere in the UK and in other countries.

<table>
<thead>
<tr>
<th>number of programmes</th>
<th>programme title (or group)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Computing Science</td>
</tr>
<tr>
<td>8</td>
<td>Networks</td>
</tr>
<tr>
<td>7</td>
<td>Computing</td>
</tr>
<tr>
<td>6</td>
<td>Computer Engineering, Business Information Systems</td>
</tr>
<tr>
<td>5</td>
<td>Computer Games</td>
</tr>
<tr>
<td>4</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>3</td>
<td>Multimedia Development</td>
</tr>
<tr>
<td>1</td>
<td>Computational Physics, Mechanical Engineering</td>
</tr>
</tbody>
</table>

Figure 3.1 – ‘Computing’ degree programmes available in Scotland in 2006\(^2\)

\(^1\) The author started his teaching career under a new blood appointment.

\(^2\) The data were gathered by entering ‘Computing’ onto the ‘subject’ field and selecting ‘Scotland’ from the ‘region’ field on search.ucas.co.uk/cgi-bin/hsrun/search/search.hjx;start=search.HsSearch.run?= 2006 on June 2006.
These were boom times which continued until 2001 when the year-on-year, 15%-20% growth in applications to study CS degree programmes (SFC, 2002) was reversed to a sudden year-on-year decline (Wright, 2006; BBC, 2006; Mander, 2004; SFC, 2004) and is reflected in the official statistics (figures 3.2, 3.3 and 3.4). In figure 3.4 it can be seen that the attractiveness of Computing Science as a core subject fell in one year below that of History and Social Work and would, had the trend continued, have placed CS below that of Sports Science in terms of new student popularity. This is a relative fall from 9th place in 2005 to a possible 12th place by 2007. The trend has also affected the new CS sub-field areas of design (for digital visual art), media studies (for digital entertainment applications), music (for digital audio) and electronic and electrical engineering (for computer and Internet hardware.)

![Graph showing students applying and accepted studying over academic years](image)

**Figure 3.2** The popularity of Computer-related degrees at UK universities and colleges (Higher Education Statistics Agency, 1997 – 2005; UCAS, 2000 - 2005)

---

1 The data were only available from UCAS for years 2001 onwards, and, at the time of gathering, the data were not available from HESA for 2005.
The data available (see figure 3.2) on those studying CS had not followed the trend of falling applications to study the subject. This could imply a lowering of the *going rate* entrance qualifications to study the field, which - if true - would be a further pressure on CS teachers and could have long-term effects upon CS as a renewable field if fewer, less-smart people were graduating as Computer Scientists.

![Graph showing applications to study Computer Science at some Scottish universities](image)

**Figure 3.3 Applications to study Computer Science at some Scottish universities**

It is interesting to note (see figure 3.4) that Mathematics continued to increase its attractiveness to potential new students. Historically, as mentioned above, this was the core subject of those who entered CS degrees, and so may be an indication of where the drift away from Computing has at least partly been heading.

There are indications that the downward trend in applications to study undergraduate CS may be over. In the UK (see figure 3.2) the rate of fall was far less for applications to study CS for 2005 entry. The Scottish universities (see figure 3.3) appeared to have come out stronger with a reverse of the fall into a significant rise in 2005, but this reversed sharply again in 2006, perhaps being a one-off effect relating to the introduction of fees at English universities. The latest (at the time of writing) draft figures from UCAS show a 5.7% rise in applications to

---

1 The data supplied by Heriot-Watt University for 1997 to 2004 has been adjusted back by one year as it didn't look correct; the figure for HW in 2004 is extrapolated.

2 Data gathered from responses to FOI requests to universities; see appendix C.
study CS across all UK institutions in 2008, but this remains below the average increase of 9.5% for all subjects and below the still increasing popularity of Mathematics, up 8.1% and now at 60% of the applications rate of CS. (ucas.ac.uk/website/news/media_releases/2008/2008-10-15).

![Bar chart showing applications to study at UK Universities with ranking (BBC, 2006)](image)

The numbers of students dropping out of CS degree programmes in their early years has become a serious problem (Roddan, 2002; Cohoon et al, 2003). As student numbers fall due to lower applications and high failure rates so faculty have started to lose their jobs – part-time, short-term contract and full-time contract – across UK CS departments. These trends have also been noted in the USA (Tucci, 2005; Cohoon et al, 2003) and Australia (Maslen, 2006).

A more recent development has been the threat to CS research funding. The funding models proposed by the UK Government’s Department for Education and

---

1 For example, Bolton, Abertay and Paisley universities terminated casual faculty contracts and did not renew expired short-term contracts in their CS departments for 2005-6. At the same time Derby laid off 1/3 of their full-time CS faculty.
Skills would hit the field hard, CS being the only subject to come out with less money on all four change models:

Computer Science takes the biggest hammering. The discipline faces potential funding cuts of more then £14 million ...

(Fazackerley, 2006)

The problems that CS faces are expressed in such opinions as,

The dangers for Computer Science are acute ... for the first time in its history Computer Science is not expanding; other disciplines are encroaching on its territory ... we are now seeing 18-year-old students with at least 10 years computing experience, who can write flawless code while holding a dozen simultaneous e-conversations and watching TV ... they find "hello world" uninteresting ... they are natural collaborators – wired, extrovert and hyper-social; they are digital natives whereas their teachers are digital immigrants. Will Computer Science survive the current turmoil?

Keith Mander, University of Sheffield (Mander, 2004)

I am always shocked at the low levels of intellectual rigour that seem to apply to much of the teaching of Computer Science/software engineering. We are good at teaching techniques ... but hopeless at encouraging our students to take a sceptical and informed perspective.

Mike Holcombe, University of Sheffield (Holcombe, 2001)

In my own work, and in my workaday contacts with other university, college and school CS departments, there is evidence of changing teaching practices. CS academics are beginning to make more use of teaching software, social networking, lectorials\(^1\), open-ended assessments, on-line posting of work, and teaching students to use reflective practices in their work. So, although it's hard work, and

\(^1\) A meeting of the entire class where the lecturer brings a prepared set of notes, starts it like a lecture then allows a discussion to develop, one person speaking at a time, usually with the lecturer or even between class members, witnessed by the entire class. A useful way to debate a new or developing idea on an applied field such as Computer Science at this site. The entire notes are rarely used as they are discussion-starters rather than formal content.
a large amount of CS field volatility continues to be present, it’s not all doom and
gloom in CS teaching.

Other disciplines and areas of knowledge have been, and continue to be, affected
by changes in knowledge and in applications, and in their field’s relationship to the
labour market. This study considers CS as a field that is particularly vulnerable due
to its own rapid development, where new developments are emerging at speed,
and often outside the Academy. The study presents what I hope may prove to be a
uniquely interesting case of the more general challenge to academia presented by
such rapid changes of knowledge, and how they connect to the ‘real world’
problems of teaching, learning and employment.

What evidence is there for the changing nature of CS as it affects a particular group
of academics, working in a particular department, at a specific university in
Scotland? To understand this we need to explore what Scottish universities have
been offering in their CS studies.
3.4 Changing 'Computing' in Scottish Universities

This section is based upon an analysis of data gathered from the prospectuses of five universities – Abertay, Caledonian, Dundee, Glasgow and St Andrews – and, in the case of the two 1990’s universities (Abertay and Caledonian), their precursor institutions: Dundee College of Technology, Dundee Institute of Technology, Glasgow and West of Scotland College of Domestic Science, Glasgow College of Technology, Glasgow Polytechnic and Queen’s College Glasgow.¹

CS is a young field. It is useful, for the purposes of this study, that there exist printed prospectuses held securely by university library archive sections. These documents show evidence of the first appearance of CS as an offered subject at these universities. They also present evidence of such sectoral changes as multi-institutional mergers, the removal of the binary line and institutional renaming which affect staff including Computer Scientists.

Scotland has 21 higher and further education institutions, most of which teach CS courses (universities-scotland.ac.uk). This is a varied sector that continues to change with, for example the recent merger of Paisley University with Bell College to form the University of the West of Scotland (UWS), the proposed merger of Stow and Glasgow Metropolitan Colleges and the transfer of CS from Ayr College to UWS. The chosen five institutions were considered representative as they cover a wide range of age as modern institutions (e.g. 1413 foundation St Andrews to 1994 Abertay), size (e.g. 4,000 students Abertay to 26,000 Edinburgh), elite status (e.g. Glasgow and Caledonian) and geographical spread (Dundee, Glasgow and St Andrews.)

The development of CS at Scottish universities can be viewed through their undergraduate prospectuses from the 1960’s onward. At Glasgow and Abertay the earliest degrees were in Mathematics and Computing. At Dundee this link

¹ For a fuller explanation of the choice of data source and universities, see §5.2
continued to appear in the degree title\(^1\), *Mathematics and Computer Science*, until 1995, and at St Andrews Mathematics was a compulsory part of the *Computational Science* degree until the same year. At Abertay the relatively new *Computer Games Technology* degree retains a significant core of compulsory Mathematics. In some universities, such as St Andrews and Abertay, the two subjects are still taught in the same department\(^2\).

These prospectuses show a link between Electronic Engineering and CS from the latter subject’s earliest times. In 1971 Dundee’s *Electronic Engineering and Electronics* degree had significant Computing components, as did Abertay’s similarly named *Electrical and Electronic Engineering* degree from 1974. By 1986 Glasgow had launched a degree programme in *Computer Science with Electronics*. These generally grew out of the Engineering faculties, and in 1981 at St Andrews, which lacked an Engineering faculty, the *Computer Science with Electronics* degree was created by the Computational Science and Physics departments. These new degrees appeared at the same time as the decline in popularity of Physics and Engineering as undergraduate degrees, often involving mergers around and between Computing, Engineering, Mathematics and Physics departments.

However, the Computing Scientists’ primary concern was with the aspects of the combined subject as a software discipline whereas the prospectuses’ evidence indicates that Engineers and Physicists tended to view it as being hardware based.

Computing as programming appears to have developed into two major themes: Mathematics and Engineering (via FORTRAN) and Business Information Systems (via COBOL.) The universities’ CS undergraduate offerings show growth in numbers of named awards and a sudden and recent decline. CS as a business application teaching area shows offerings that are more patchy and inconsistent

---

\(^1\) for brevity only the named part of each degree is listed; so a ‘Bachelor of Science degree with honours in Computer Science’ is listed simply as ‘Computer Science’. Also, degree titles are shown *in italics* to differentiate them in the discussions from subject areas.

\(^2\) Much has changed and continues to change with respect to the terminology used in educational institutions to describe their academic-management structures, for example some high schools now having ‘faculties’ and some universities’ ‘faculties’ being replaced by ‘schools’. For the purposes of this thesis the basic unit of the academy is referred to as the ‘department’.

95
than those for scientific CS. For example, Caledonian first launched a Data Processing award in 1972. From 1974 to 1994 at St Andrews there was an option in Information Processing for MA undergraduates and a short-lived Information Programming degree from 1983 to 1986. Abertay's first business computing degree was a taught postgraduate diploma in Information Technology launched in 1984, and the first mention of programming at Abertay is in 1978 in the Accounting degree.

Other universities show this ambiguity towards CS as a Business discipline. At the former Queen's College Glasgow, at Glasgow and at Dundee CS never emerges as a named business degree. At Caledonian and Abertay the growth of Management Studies may possibly be related to this. At Abertay there was a debate at the time of the Business Computing degree launch that was a possible predictor of change as the award was one of now many CS degrees that are awarded as Bachelor of Arts degrees; a major change in view and content from the established Bachelor of Science awards in core technical CS.

There has been a significant decline in named degrees in the overlap between Management Science and CS. At Caledonian, Information and Business Administration, Information Management Systems, Information Technology Studies, Knowledge Management, and Information Systems Development degrees were being offered to 2002 entrants, but by 2006 these had been reduced to just one, e-Business. At Abertay the Business Computing division of the School of Computing was transferred to the Dundee Business School in 2004, taking around one quarter of the Computing school's staff, often reluctantly, across this divide.

The prospectuses show a recent and increasingly strong thread in CS awards at the Scottish universities in teaching CS as the medium for developing and delivering a range of digital aesthetic products. The evidence indicates that these new degrees are probably built upon existing degree studies in the Applied Arts, including Music, Film and Television Studies. In 1999 Glasgow launched a degree in Electronics with Music, Abertay a Computer Arts degree (a combined visual art and music degree) and Caledonian created three new degrees, Music Technology with Electronics, Applied Graphics Technologies, and Multimedia Technologies. Abertay (1997, 1998 and 2000) and Caledonian (2004) launched Computer Games degrees; Abertay's
one is based on a foundation of CS, Software Engineering and Mathematics whereas Caledonian’s one is more aligned towards Art and Design.

Abertay have continued to place future bets on the strength of this new CS subfield to attract incoming students to study, with three new, and notably distinct, undergraduate degrees in *Audio Production for Digital Media, Mobile Entertainment*, and *Multimedia Development* newly offered to prospective students for entry in 2006. At Glasgow a degree in *Arts and Media Informatics* was launched in 2005. At Dundee the *Interactive Media Design* degree was co-launched by the university’s Duncan of Jordanstone College of Art with the cooperation of that university’s Department of Applied Computing. Such new degrees in digital aesthetics require Computing Scientists to interact closely with other academic disciplines in order to create these new areas of study. This can be difficult. Abertay employed a new group of staff, none of whom were Computer Scientists, to offer their *Computer Arts* degree.

This new area of development in CS – what might be called ‘digital aesthetics’ – has emerged in some universities from the Electronic Engineering with CS degrees offered to support the developing Internet and Telecommunications subject areas (increasingly the same medium) and to teach students how to work the content transferred across these media, such as movies, games and music videos. In 2002 St Andrews launched what could be seen as a radical departure for such a traditional university and its long-established CS department, an *Internet Computing* degree award. In the same year this was offered, Caledonian appears to have built upon its existing expertise in Telecommunications in the Engineering department to launch a degree in *Networking and Computer Support* and later, in 2003, a degree in *Internet Software Development*. Abertay’s Engineering department similarly launched in 2001 an *Internet and Communication Technologies* degree. 2005 saw the first intake, at 3rd year only, to Abertay’s *Web Design and Development* degree, and the launch of a highly controversial degree programme in *Ethical Hacking and Countermeasures*; controversial as it lacks the trans-disciplinarity in teaching of many of the recent new degree offerings mentioned above, and in structure, shows no scope for students to formally consider the Sociological, Philosophical, Theological or Anthropological issues around Internet crime. The two Abertay degrees – *Computer Arts* and *Ethical*
Hacking and Countermeasures – can be seen to act as illustrators of the challenges to the CS academy in attempting to create new kinds of degrees and subfield studies which require an expertise situated outside the existing boundaries of academic CS. They may also illuminate the lengths that academics both will and can go to in order to satisfy simple university funding policies as numbers applying to read more traditionally-named and structured CS degrees decline. Having been so closely and evidently linked to such fields as Engineering and Mathematics, and seen the consequences of their declines as job loss and departmental closure, so the CS academy has continued on a varied search for new students.

CS has also stretched its degree offerings into areas more traditionally associated with the Social Sciences. Degrees in Cognitive Science, Neuroinformatics and Artificial Intelligence, such as at Dundee (from 1998) and Glasgow (from 1999) bring Computer Scientists into what may be a more comfortable interface with similarly empiricist Psychologists. A particularly Social Science take on Computing was evident early at Glasgow when there was a stream offered from 1986 to 1998 in Technology and Society. Intriguingly, the earliest mention of CS at the new universities of Abertay and Caledonian (Queen’s College Glasgow) was in Educational Technology (from 1976 and 1977 respectively). In both cases the inception of these teaching offerings also coincided in the prospectuses with the first visible appointments at each establishment of Lecturers in CS, but this is probably coincidental.

The prospectuses show that CS as an undergraduate teaching subject has grown slowly at the Scottish universities. None of Dundee’s offerings - Computational Science, Computer Science and more recently Applied Computing - had more than a core offering of teaching in CS in the award. At Glasgow and at St Andrews Computing Science degrees were offered, despite a maximum of only 1/3 of the CS student’s studies in first and second year being in the field. Abertay’s Computing degree is similar to offerings at the other 1990’s Scottish universities, aimed to attract HNC and HND direct entrants from local colleges into second or third year of Computing, it was also a safety net for those who failed Applied Mathematics and, more recently, with the tough programming in their high profile Computer Games Technology degree
There is little evidence of linkage from CS to the Natural Sciences as a taught field. Abertay has offered, since 1999, two awards in *BioInformatics*. This is presented as a research strategy, bringing into interface Computer Scientists (who are interested in digital image modelling and data visualisation) with Natural Scientists (who are interested in researching natural Biological and Biochemical processes.) However, the evidence in the other universities’ prospectuses for such working together of the CS and Natural Sciences is that this coupling is rare. Others, such as Caledonian’s *Chemistry with Information Technology and Instrumentation* are perhaps evidence of a clumsy attempt to stall the decline of Natural Science studies at a new university, for which there is ample evidence elsewhere in the prospectuses.

These documents show CS being presented to potential new students. But, what perceptions do the readers of these documents bring when considering a career in CS? That is the next topic.
3.5 Computing Science as Viewed from Outwith the Academy

Tech know-how decreases with age. The younger you are, the more you inherently know in that (CS) realm. It’s true that teens easily incorporate digital gadgets in nearly every aspect of their lives.

(csmonitor.com/2005/0815/p08s01-comv.html)

College students probably spend more time than ever in front of a computer, but fewer seem to want to make computers their life work.

Tucci (2005)

This gap between the different generations’ views of computers and CS has even been noted by Larry Ellison (CEO, Oracle Corporation) - one of CS industry’s big three including Bill Gates (ex-Microsoft) and Steve Jobs (Apple). Ellison recently said that,

If I were 21 years old I probably wouldn’t go into computing. The computer industry is about to become boring. I’d go into genetic engineering. (Tucci, 2005)

There is evidence that a major – if not the - issue for young people is the perceived nature of CS itself: CS may be viewed as being dull. Students perceive a gap between their expectations of degree study in the field and in the academic teaching and learning realities once they arrive at university (Swing 2003; Draper, 2005). Draper (2004) and Chickering et al (2005) blame poor teaching in CS and lack of faculty-student engagement. In addition Draper (2005) pointed a finger at a tendency to over-pack the curriculum of undergraduate CS degrees. CS faculty can
respond by blaming others, for example high schools or corporate greed which can distort the subject (Morris, 2004). There is an atmosphere within the CS academy of having to get all of the expanding and changing field taught to students. This carries with it a danger of skimming over the field and of not allowing CS students time to digest and reflect upon their learning, which is particularly problematic with such notoriously hard parts of CS as programming (Jenkins, 2002) – hard to teach and hard to learn, yet core to the subject. CS departments are aware of this struggle,

A challenging aspect of the computer science curriculum is to present the ever growing and ever changing topics of computer science within the time constraints imposed by the program of study. (Krishnaprasad, 2000)

Computer Science ... has been shaped by a rapidly growing body of knowledge. (nmt.edu/catalog/2004/arts/cs.pdf)

Computer Science ... an ever-changing field.
(cs.unh.edu/aboutus.htm)

... a field as rapidly changing as computer science.
(cs.northwestern.edu/academics/undergrad/curricula/cis.php)

... the rapidly changing character of the field (CS).
(cs.earlham.edu)

Yet, some of these same documents which are cited above are elsewhere at pains to show that CS faculty also want to keep the teaching of CS grounded,

our students gain a solid foundation in science and mathematics
(cs.unh.edu/aboutus.htm)

Our curriculum is built on the fundamental paradigms of the discipline: theory, abstraction and design. (cs.earlham.edu)
I am a Computer Scientist with considerable experience in designing new degrees and teaching on them. I simply do not understand how they achieve this apparent balance of creating courses which are attractive to new students, rich in theory, applied in learning, passable and from which good graduates emerge. It is notable that none of those quoted above claim to do all of these. From my own experience with CS as a taught subject at many universities in Scotland, the UK and abroad, I know of none.

Despite the growth in the range of CS degrees being offered at the Scottish universities in the late 20th and early 21st centuries, CS has rapidly and recently gone into decline in popularity as a taught undergraduate academic field. Computer Scientists also struggle to keep up with the speed of the change in the applied field; for example, in my previous and current CS department there are places where academics leave obsolete books, often almost pristine, for others to take away should they wish, as the information in these recently-published books is no longer useful. CS knowledge is in flux, emerging, dissipating and re-emerging, making the knowledge-base difficult to define. In his seminal work on the nature of life in the academe, Becher (1989) discussed how the life of the academic is one of coming to terms with the nature of knowledge as such features as it being hard and soft, or, pure and applied. Both he and Agassi (1995) discussed – in what CS academics would call a 'recursive' way – how much of where this debate on the nature of new knowledge is located is itself situated in a cross-disciplinary area. CS, and the study of change in CS, are each examples of a 'little explored border zone' (Becher, 1989, p6) and a 'relatively new academic discipline' (Agassi, 1995). That is my understanding as a Computer Scientist; it is also shown in the literature investigated earlier in this chapter.

CS has also become part of a highly interdisciplinary domain, vividly illustrated by Hayes (2004) as one with, 'intellectual migrants wandering back and forth across many academic frontiers". He does offer hope that the field may settle down in due course,

At one time, the term physics had a very broad meaning, roughly synonymous with natural science ... everything from astronomy and cosmology to meteorology, chemistry, zoology and botany (but not stamp collecting).
3.6 Summary

CS academics, in the evidence considered above of the (mainly) undergraduate degrees and studies offered by these universities, can be seen to be regularly attempting to align themselves with issues relating to the nature of knowledge in a changing and diverse field. In the prospectuses considered the older universities (Glasgow and St Andrews), mid-life (Dundee) and younger ones (Abertay and Caledonian) have all shown a range of evidence of CS as a field still in development, where different approaches to delivering the subject have been and continue to be made. Some innovations did not bed down into CS, such as attempts to link the field with the Social Sciences. Other fields have shown a limited affinity with CS, such as the Natural Sciences. The linkage with and from Mathematics is all but gone in terms of teaching and degree offerings. Other areas show considerable strengths when combined with CS. The Applied Arts have linked strongly into CS, as have certain parts of Engineering, particularly Telecommunications.

In the Management Sciences there was a long-established affinity with CS, but this has been broken. It is worth noting that the breaking of the once-strong links with Management Science and Mathematics have not affected the popularity of these two areas, which continue to attract increasing numbers of new students.

The changes in CS degrees at Scottish universities has taken place against the backdrop of enormous developments in the field. Hardware, whilst remaining a constant, has moved from room-sized environments to the shirt pocket; programming from Algol to the Unreal Engine; applications from early calculators to today's virtual 3D worlds.

Some academics have embraced change as an essential part of CS. Other Computer Scientists consider CS to have a definite, sacrosanct core that must be
taught and learned. In a subject like CS that is ever emerging, developing, transforming, interfacing and even disappearing, what is the correct knowledge to deliver and teaching patterns to use when teaching a new CS student over the course of a four-year study in a named degree? How do CS academics come to terms with this constant state of change? These are the kind of questions this thesis will go on to investigate with a particular, situated group of CS academics.
4. Methodology

I have found it useful to divide what is often described as the methodology chapter into two sections: method (chapter 5) and methodology (chapter 4). This isn’t linguistic pedantry, but a distinction which I found necessary and helpful. In this chapter – Methodology – I describe my reading and understanding of the fundamental practices and understandings around qualitative research approaches particularly within the case study method and with respect to my own study. The following chapter – The Method - is my record of what was done, which, although to an extent a standalone chapter, needs to be read in the light of the current chapter where deeper explanations of the decisions made in formulating the study were grounded.
4.1 Introduction

I keep six honest serving-men
(They taught me all I knew);
Their names are What and Why and When
And How and Where and Who.

Rudyard Kipling (1903)

There are many texts on how to do qualitative research, often written by people
with years of experience in the field, some of whom are strong communicators on
the issues and complexities of being a qualitative researcher. I entered this study
with a mixed academic and professional background including Mathematics, Biology
and Computer Science. Each of these areas of study, as does Education, comes
with its own particular presentation and understanding of knowledge and its
research approaches which the tyro must learn.

One of the main warnings in research methods texts is: do not to jump in without
thinking. It is an easy, and indeed a simpleton’s, solution to decide that that which
I don’t understand is that which I need not understand, as Robson (1993, px)
stated, the research task is easy, ‘... common sense – and how difficult it is for the
same reason’. This is a case study of the perceptions of a particular group of
lecturers in an educational situation in Scotland. A significant amount of the writing
on Educational research is about comparing qualitative with quantitative data
research. Overall, this is a qualitative study of the words of these lecturers with
aspects of qualitative and quantitative analyses on the background to the study.

In the rest of this chapter I have sought to discuss, compare and contrast what
appear to be the major issues that were taken into account in order validly to
proceed to and through each phase of the study. What is deemed to be clear has
been omitted, and what required methodological decisions are discussed.
This chapter was informed by my reading of many texts, from which I created understandings of: 'how to do', 'how it was done', and, 'why to do' and 'why it was done'. Interestingly, this meta-approach was one of creating encodings (finding structures) from the qualitative data (methodology texts) from the sequences of words that are the books, chapters and papers; and so this methodology literature review was itself a small pilot for the then new-to-me approach of coding.
4.2 Case Study

It is worth stating at the outset that there is a great deal of controversy and even contradiction in the many writings on the case study approach as a research tool and over its meaning, types, methods, structures, purposes and even usefulness, perhaps particularly in the use of the case study in qualitative studies (see, e.g., Holliday, 2002, pp217-23; Robson, 1993, p147; Yin, 1994, pp11-2,14-5). Reading the work of many of the qualitative methodologists on case studies shows, to this researcher, not so much established areas of agreement to disagree as a complex set of emerging understandings. Creating dividing lines between classifications of kinds of case study - such as between intrinsic and instrumental - is, as Wellington (2000, p93) put it, ‘often difficult’, indeed even the definition of a case can according to Yin (1994, p21) be a ‘fundamental problem.’ However, I have taken the approach of using a case study because of its power and ability to, ‘catch the complexity of a single case’ which is in ‘itself of very special interest’ (Stake, 1995:xii). The case can be viewed a ‘specific instance’ (Cohen et al, 2000, p181) and also one where the site of the study provides access to ‘contextual conditions which might be highly pertinent to the phenomenon being studied’ (Yin, 1994, p13). In short, a case study approach allows a significant phenomenon to be investigated, analysed and reported.

The general approach taken to the case study is qualitative as I that seek to create some understandings through the nuances, happenings, context and the place of the individuals (Stake, 1995, pxii) in their context – my old workplace - which was ‘strong in reality’ (Cohen et al, 2000, p184). As a case study researcher I had available a wide pallette of methods for data gathering and analysis, and so was careful that decisions were not taken in an ‘arbitrary’ way (Stake, 1995, pxii) but rather through a process of ‘thoughtful and informed decisions’ (Gay, 1996, p218) that were relevant to the people in the study. As Cohen et al (2000, p181) wrote, the people, who include the researcher, are more important in making decisions on approaches than the methods or techniques that can be selected.

I approached the study with certain useful skills (writing, interviewing and organising), advantages (knowledge of the setting and participants, and, opportunity) but also with learning to do (qualitative methods). I followed the line
proposed by Stake (1995, p1) that the core of an Educational study is often its people and programmes, their combined uniqueness, their common reference points beyond the boundary of the particular case, and that the researcher can seek to understand the case by listening to the participants who are involved. As Stake continued, 'much of our gathering of data from other people will take the form of stories they tell' about their teaching practices (Grundy, 1987, pp42-3.) For myself the fascination of the case was also that the study could make visible,

'[a] group of people who regularly associate with each other [that]
develop a culture, i.e., characteristic ways of behaving and thinking ...
beliefs, attitudes, values, customs and the like' (Gay, 1996, p219),

in their bounded world which, 'is subjectively structured, possessing meaning for its inhabitants' (Cohen et al, 2000, p187) from which, after analysis of the data, could usefully be reported.

A case must be bounded to be a case (Stake, 1995, p2; Miles et al, 1984, p28; Cohen et al, 2000, p182) and interesting in order to get a specific and perhaps more general understanding of the issues it, through the study's analyses, illustrates. However, although the case in this study is specifically, 'a unique example of real people in real situations' (Cohen et al, 2000, p181) set in a particular time and viewed through my agency (Harland, 1996, p94), it also naturally sits in the context of an academic subject in an educational sector and in its age's views of the meanings of education. The primary aim of this study is to understand this particular case (Stake, 1995, p4) and so open up a particular world to the reader with its own unique characteristics, set in a wider world (Holliday, 2002, p51.)

I do not intend to draw any generalisations from the case researched, but simply to present the findings of my understandings of what I felt this case revealed about how this group of academics handled changes. The aim was to understand the particularisation of this case as it illuminated particular processes (Scott, 1996b, p150). There is a notion that particularity makes an analysis less valid or weak, but having lived the case I am with Cohen et al (2000, p183) who stated that this negative opinion is more one of 'prejudice and ideology' than indisputable fact. Nonetheless the analysis, which is the important feature of a reported case study (Silverman, 1993, p22), must be visibly logical, the findings must follow clearly from the data via strong and suitable techniques and related methods. Hopefully these presented in the next chapter, The Method, do appear so, allowing you, the
reader, to 'emerge with a well-grounded sense of the local reality in a particular setting' (Miles et al, 1984, p151) for, 'the validity of the study must be assessed and judged by the reader as much as the researcher' (Wellington, 2000, p99).

The reader should also note that in this case study the data were drawn from a relatively small database (Stake, 1995, p14), and so the interpretations and findings presented are cautious, patiently drawn and reflective (Stake, 1995, p12.) The quantity of data gathered was also carefully controlled to avoid the trap which (King, 1979) reported falling into: ending up with an unmanageable, 'half a million hours of notes made during nearly six hundred hours of observation.'

If a line should be drawn, the case is more one of intrinsic (Stake, 1995, p3) interest in and of itself rather than one which is instrumental to understanding bigger issues. I have sought to avoid 'data bending' to create a wider generalisation (Miles et al, 1984, p28). The case is also one that is not-interventive and empathetic in methodological approach; the former was achieved by careful invitations to take part, by offering the interviewees the opportunity to review their data and the findings, by making permissions verbal (but recorded), and by locating discussions in each interviewee's own office. Empathy was achieved by not acting the part of an interviewer, more of a leader of an open conversation between peers. Of course, there did remain power differences: the interviewee controlled the time and location, the interviewer controlled the duration, the interviewer opened and closed the conversations, the interviewee controlled much of the content, both shared responsibility for the direction of the conversations, and, the interviewer controlled the interpretation and publication.
4.3 The Setting

What is the meaning, structure, and essence of the lived experience of this phenomenon for this person or group of people?

Michael Quinn Patton (2001, p104)

In choosing where best to locate the study issues of context, access and culture arose. That is, the study took place in a location which itself existed in wider contexts: structurally within this university, as a particular delivery of a global academic subject and politically within education in Scotland. This case presents a research project which,

is a human construction, framed and presented within a particular set of discourses (and sometimes ideologies) and conducted in a social context with certain sorts of social arrangements, involving especially funding, cognitive authority and power. (Punch, 1998, p140)

Within the set limits of time and other resources it was necessary to satisfactorily tackle the ‘problematic’ (Good et al, 1989, pp211-232) issue of negotiating access within my own workplace; a non-trivial issue relating to gaining physical access and being trustworthy with established relationships (Measor and Woods, 1991). The setting was potentially one which was, as Holliday (2002, p74) stated, ‘sufficiently rich for data to emerge.’ However, getting the access conditions wrong would, ‘seriously affect the design, planning, sampling and carrying out of ... [the] research’ (Wellington, 2000, p63). Indeed, as Robson (1993, pp295ff) stated, poor access can compromise research, or even scupper it entirely. So, the official gatekeeper, in this case the Head of Department of CS, also had to give clearance (Whyte and Whyte, 1984, p62; Denscombe, 2003, p91).
The setting had its own micro-politics (Usher and Scott, 1996) and as such the research project was not simply ‘an abstract and neutral academic tool’ (Punch, 1998, p140) but a ‘political process’ (Usher and Scott, 1996) as was sometimes evident in the conversations. In this university’s marketplace and with its funding mechanisms, there were and are very real concerns over personal and corporate image and identity (Henkel, 2000, pp13-15) and the ‘uncertainty’ (Halsey, 1992, p271) of the potential effects, or fear of the effects, of identification as a case. This study began with a recognition that in academia in Scotland there is a context of significant recent transitions and changes in the sector and upon the lives of academics such as those who were interviewed (Henkel, 2000, p13).

Neill (1968, p19) stated that his study was based on, ‘the story of a modern school - Summerhill ... founded in 1921 ... within the town of Leiston, in Suffolk ... about one hundred miles from London.’ He was remarkably explicit in this. Coleman (1961, pp vii ff) only gave ‘a short description of the ten schools’ in his study, hiding their names behind such descriptive pseudonyms as ‘Farmdale’ but still located them in reality ‘in northern Illinois’. Woods (1979, p4) took a similar tack of a pseudonym for the particular setting and for the names of pupils and teachers, with such specific details as ‘a secondary modern ... built in 1956, with 560 boys and 30 teachers ... [which] serves a rural area in the Midlands.’ Griffin (1985a, p201) again took a similar line, ‘All of the people and places ... have been given pseudonyms’. If this case study were to state any one of the date of foundation, the geographic location in Scotland or the student roll, most academic readers would quickly locate this study to people and place. Putting the politics issue of such identification to one side, there is also, as Woods (1979) and Griffin (1985a) considered, the issue that clearly identifying the site of a study can distract from the findings.

Punch (1998, p149) stated, the elements of good qualitative research include:

- intense and prolonged contact with the field situation,
- empathetic data capture,
- identifying themes with the participants,
- understanding actions and motives,
- using a wide variety of possible instruments,
- primarily using words.
In this case these were more easily done as I had worked long in the research location, had positive contacts in the workplace, and, as Coleman (1961, p3) wrote, 'subcultures can exist right under ... [our] very noses'. This case had a culture that could be studied where were located a common set of orientations, beliefs, values, concepts, purposes, ideals, theories and notions of reality which were accepted by those within as being commonsense and natural in a site of shared knowledge and informal rapport that would be expressed easily by those within the setting (Alvesson, 2002, p54; Lankshear et al, 1997; Fairclough, 1989; Griffin, 1985a, p105).

In identifying this particular site as a location for a case study there were broad similarities to those identified by Coleman (1961, p3) in his study of a, 'separate subculture ... with a language of its own ... [and with its own] value systems'. This quality was focussed down onto academia and its subject areas by Becher and Trowler (2001, p41) who defined a separate academic culture as being one where,

- leading organisations recognise the subject in organisational structures
- a free-standing academic community has emerged;
- it has its own professional association and specialist journals,

three criteria that CS satisfies at this case's site and in the wider world as a self-reflective community (Carr et al, 1986, pp207-9.)
4.4 The Researcher’s Presence in The Research

In researching a situation that is as highly familiar as my then workplace, a place of ‘first, intimate, habitual familiarity’ (Whyte and Whyte, 1984, p282), I took the role of ‘participant observer’ (Woods, 1979, pp260ff; Bryman, 1988, p45,95ff) or ‘participant researcher’ (Robson, 1993, p446ff; Cohen et al, 2000, pp228ff) which opened up opportunities validly to be in and of the research environment. This is often a ‘key role’ (Adelman, 1985; Wellington, 2000, p45) in fieldwork allowing ease of entry to the field and enabling rapport with other participants (Bryman, 1988, p45; Wellington, 2000, p45). This rapport allows the participant observer/researcher to, ‘become engaged in the conversations, actions, and lives of the people they study’ (Judd et al, 1991, pp301-2) and by so doing the researcher finds that he needs, ‘to participate so as to observe and experience ... to know a setting’ (Measor and Woods, 1991).

There are guidelines for ‘field-workers [who] participate fully in the lives of the people they study, by becoming members of the group’ (Judd et al, 1991, pp301-2). Of course, it is significant that here I, the researcher, was already part of the research setting, and had been so for many years (Pollard, 1985, pp217-34) as a Reader in the department, ‘part and parcel of the setting, context and culture ... [I was] trying to understand and represent’ (Altheide and Johnson, 1994). My colleagues had worked with me in developing degree programmes, teaching and in active CS scholarship. As Judd et al (1991:302) stated, ‘some researchers are ipso facto participants in the lives of those they are studying’. Even so, some colleagues are always easier to interface with than others. As I was of the old guard of Software Engineers/Computer Scientists, I was already, and of long standing,
'accepted into the group' (Whyte and Whyte, 1984, p70) of the research participants.

There are pros and cons of doing a participant observer/researcher structured project. *How-to* textbooks which outline techniques and methods are useful, but the careful researcher can capture more of the pros and seek to avoid some of the cons by reading *what-I-did* accounts and reflections upon research on using particular techniques and methods, such as those of Pollard (1985), Woods (1979) and Neill (1968). From these it was possible to identify particular problems in acting as a participant researcher:

- time - many participant researcher studies are part-time. I was there all the time as part of the site (Yin, 1994, p80), but also had to take a slightly distanced eye on things, which was also itself time-consuming (Whyte and Whyte, 1984, p226);

- lack of expertise – this is recognised as being a difficult approach. As Fontana and Frey (1994) stated, the good qualitative researcher, 'must be flexible, objective, empathic, persuasive, a good listener and so on.' However, I already had a strong background in CS teaching, in interviewing and in writing on the workings of small software development studios for the CS trade press;

- lack of confidence - it is possible for the new researcher to decide that it is simply too hard (Gay, 1996, p230). However, reading previous accounts, talking with qualitative researchers and doing a pilot study inspired confidence;

- insider issues - intra-university issues of 'politics, power and values' (Blaxter *et al.*, 1996, p15) are not all visible even to a long-term employee. I decided the safest route was 'never [to] be covert in their own institution' (Good and Watts, 1989) allowing power issues to arise and be tackled naturally and so to keep the method simple;

- problems of gaining access - this is raised regularly, and by such as Good *et al* (1989), Fontana *et al* (1994), Wellington (2000, p45) and Denscombe (2003, p89). Being an employee, physical access was easy. Being a
colleague required negotiation (Holliday, 2002, p157) with those I would be involving as participants in the study;

There are also particular pros in using a participant observer/researcher approach in a study:

- **insider opportunity** - the site was already well understood (Yin, 1994, p80) by the researcher and by the other participants, who were located in close proximity to each other in the workplace (Bryman, 1988, p96). With such relatively easy access to the opinions of the researcher could view, ‘... events, actions, norms, values, etc. from the perspective of the people being studied’ (Bryman, 1988, p61). The views gathered could then be honestly portrayed in, ‘a climate for mutual disclosure’ (Miller and Glassner, 1997);

- **practitioner opportunity** - as a practitioner in the workplace in the field under investigation, ‘the researcher is inextricably part of the phenomena being studied’ (Maxwell, 1996, p67; see also Altheide and Johnson, 1994). I was acutely aware of this and so needed to face the fact that I was playing an established participant rôle in the scene being studied;

- **participant-researcher synergy** - my rôle of being-there allowed me to more clearly understand issues (Robson, 1993, p447), to act as a, ‘crucial measurement device’ (Denscombe, 2003, p234), to create positive possibilities of being ‘insightful’ (Yin, 1994, p80) through my already, ‘deep, personal involvement’ (Woods, 1979, p260) as a researcher who was genuinely interested in applying, ‘the principle of reflexivity’ (Holliday, 2002, p145) and so help to produce understandings that may then produce better practice.
complete participant

**mainly participant**

participant-as-observer

observer-as-participant

**mainly observer**

complete observer

figure 4.2: a typology of naturalistic research roles after Gold (1958) and Wellington (2000, p93).

This study was also one where I had to move my stance as researcher; I changed jobs, albeit after the interview data had been gathered, from researcher *in situ* to observer at a distance (Measor and Woods, 1991; Pollard, 1985). Figure 4.2 shows how such a change is reflected in the role of the researcher who moves away from being an 'involved observer' (Woods, 1979, p260) to being a 'stakeholder' (Shamin, 1993), yet still with an integral professional interest in the result of the study.

As the participant researcher in this study I was in a long-term relationship with those I worked with in a 'shared experience' (Burgess, 1985; Griffin, 1985b), which could make it difficult to separate out the effects of being a friend and colleague who was also, 'known to be a researcher by all those being studied' (Atkinson and Hammersley, 1998; Mac an Ghail, 1991). The research could not be allowed to slip into creating and exploiting my colleagues through the, 'inequalities, disempowerment and cultural imperialism' of the researcher over the researched (Hollliday, 2002, p15). Indeed, as Holliday (2002, p145) stated, the issue of the relationships between those involved in a participant study, 'pervades all aspects [of qualitative research]'. There are also opportunities for opening out, talking and expressing freely, by the participants. Woods (1979, p261) noted that some said to him, 'we like talking to you, it helps us feel better'. The researcher must be aware of being seen as a Freudian psychoanalyst with his participant on the interview couch. Similarly, with such a set of long-term relationships I was wary not to run, 'the risk of identifying with one or more of the participants' (Whyte and Whyte, 1984, p200) as the participant observer must be prepared both to empathise equally (Bryman, 1988, p61) and to be trustworthy with all participants (Becker and Richards, 1986, p115). I was also aware of the difference in the
experience of the study by my colleagues, the simple-participants, and myself, the researcher-participant (Herrera, 1992).

Bias is an issue that arises in all research, not just in participant-observation. It can be viewed as the effects of the investigation itself (Good and Watts, 1989), whether deliberate or accidental. As the researcher I brought to the study my own identities, values, beliefs, assumptions, knowledge, motivations and prejudices (Wellington, 2000, p43) which could not be eliminated; as Neill (1968, p89) stated, 'that I write with a bias is natural'. At worst bias can be seen as 'manipulation' (Yin, 1994, p80) when the researcher abandons objectivity. Handling bias is a delicate balance (Holliday, 2002, p178) but it can be positive and, 'seen as virtuous, subjectivity ... something to capitalise on rather than exorcise' (Glesne and Peshkin, 1992). Where necessary I have set out my biases for the reader to reflect upon, rather than unnecessarily to pretend to neutrality (Wellington, 2000, p42; Holliday, 2002, p7) but instead to embrace it as a precursor to greater insight (Wolcott, 1995). As Miller and Glassner (1997) stated, 'the subjectivity that exists in all social research will be a visible part of the project, and thus available to the reader for examination'. One of the pros of principally taking the participant-observer approach was that my own stances are made visible both in the raw data and in the analyses. There are, 'no objective observations, only observations socially situated in the worlds of the observer and the observed' (Denzin et al, 1994).
4.5 Gathering the Data

Choose your data-gathering technique based on the project's needs ...
Keep in mind that different cultures and subcultures respond differently to various techniques.

(Reagan, J., 2002).

It is often proposed that a major study should begin with a pilot (Teijlingen and Hundley, 2001), in this case as a trying out of the techniques I was considering in order to see how or if they work in situ and to modify any plans and schedules accordingly (Tizard and Hughes, 1991, p19-40; Blaxter et al, 1996, p121). This piloting was a period of exploratory research to help focus the features of the design in a more exact way (Light et al, 1990) and to generally gain more insight into the shape of the overall study where many of its practical details could be determined in advance of their being used (Jansick, 1994), not least due to my own inexperience as a qualitative researcher. With a fair degree of design uncertainty it was best not to draw the pilot's focus too tightly (Yin, 1994, p75) in order to cast light on a possible range of approaches, so I used the pilot as a way of assessing how time, location, people, duration, data gathering and general format of the enquiry might work (Teijlingen and Hundley, 2001). The data of the pilot study are not reported outside the methodology (and method) chapters as the pilot was only useful in helping understand such important factors as the availability of participants, access to the site of the study, use of the researcher's personal network and to test the human aspects of language, culture and rapport (Wellington, 2000, p59; Fontana and Frey, 1994).

The site of the study, in this case an academic workplace, is always significant. Griffin (1985a, p4) openly located her study in the schools, workplaces, pubs and homes of her participants, in their historical context (Gay, 1996, p224) and social worlds (Miller and Glassner, 1997) in order to illuminate the participants’
background. When there is access to such a wide network of participants there are opportunities to use such sampling forms as deliberate, opportunistic, convenience, purposive, or a combination of these (Punch, 1998, p193; Wellington, 2000, pp63-4). All sampling procedures have downsides (Patton, 1990) and must be justified (Weiss, 1994) in providing access to participants and aimed at being approaches which will yield data from the case site that can be analysed to provide the clearest understanding of the phenomena under study in the focus of enquiry (Maykut and Morehouse, 1994). Sampling may be justifiable with respect to the overall population (Gay, 1996, p112), but there will ever be a problem in any case study with provable representativeness (Bechhofer and Paterson, 2000, p42) which is also true when applied to an academic department and subject area, as in this case.

The aim was to find a rich source of data, preferably one with more quality than mere quantity. In this case the participants were chosen for ease of access to be interviewed, length of experience, range of experience, easy willingness to talk, and being a clearly bounded group. The use of a small sample emphasised my aversion to a study that could be seen as a basis for generalisation, but instead I was seeking an opportunity for in-depth understanding (Gay, 1996, p213; Colorado State University, 2006) from a well chosen group (Bechhofer and Paterson, 2000, p52). A small group of people can have a broad experience (Silverman, 1993, p9; Patton, 2001, p227) quite out of proportion to the sample size (Holliday, 2002, p91).

The sample included those who had been involved in, from the study’s viewpoint, significant events, who were perceptive and reflective about these events and who may well themselves feel some benefit from being involved in the study (Whyte and Whyte, 1984, p105, 109), a combination which would encourage active cooperation (Griffin, 1985b) and help ensure ongoing trust (Becker and Richards, 1986, p115).

The interview is the most common technique used for data gathering in case studies (Yin, 1994, p84) as a way to acquire data not obtainable in any other way (Gay, 1996, p223) and is itself, in analysis, a form of data making (Baker, 1997). The interview can be a way to talk about such sensitive topics as the meaning of work (Holland and Ramazanlu, 1994), to focus in on the interviewee’s thoughts (Denscombe, 2003, p167), perceptions, attitudes and motivations (Judd et al,
1991, pp260-1), his anxieties, ambivalences and uncertainties (Kleinman et al., 1994), a way to discuss his interpretations of the workaday world in which he lived, and to allow him to express how he regarded a situation from his own point of view (Cohen et al., 2000, p267). Further, and of particular interest in this study, the data produced could be seen as a perspective on human knowledge (Cohen et al., 2000) and culture (Silverman, 1993, p9).

There are many types of interview (Fontana and Frey, 1994; Tellis, 1997a; Punch, 1998, p175; Wellington, 2000, p75ff) which could be classified under the headings: formal interview, less formal and informal interview (Robson, 1993, pp230-1; Cohen et al., 2000, pp268-9; Denscombe, 2003, pp166-7).

The informal interview (Holland and Ramazanoglu, 1994) can be conversational (Clandinin and Connelly, 1994), as used by such as by Miller and Glassner (1997), Woods (1979), Griffin (1985a) and Tizard and Hughes (1991). Such an interactive style often best suits data gathering from such experts (Patton, 2001, p402) as the university lecturers in this study. I was aware that the interviews could become rambling chats (Bryman, 1988, p46) rather than a 'conversation with a purpose' (Kahn and Cannell, 1957) that was aimed at bring new material into the discussion from the interviewee (Whyte and Whyte, 1984, p97), and one interview-conversation did go well off-topic despite gentle attempts to steer the interviewee back to CS. Yet, an unstructured interview has strong points: the questions can be salient, timely, reflexive, spontaneous and individual and the answers given were often vivid. The generic weaknesses of the approach include a lack of systematic structure, and creating complex and individualised data sets that can take a great deal of time to analyse (Sayer, 1984, p220; Judd et al., 1991, p299; Bechofer and Paterson, 2000, p55; Cohen et al., 2000, p271; Patton, 2001, pp227, 343).

In deciding on an interview strategy and form I took Patton's (2001, pp271,279) advice to use the general 'unstructured' interview but bookend it with a prepared opening and closing statement or question.

Silverman (1993, p208) stated that interviews do not produce pure data, as the data gathered are mediated by the reasoning of the speakers and the listeners, in this case seasoned Computer Scientists. This study analyses these subjective data at a social interface (Sayer, 1984, p202), from events that were not merely data gathering but person-to-person, colleague-to-colleague, even friend-to-friend conversations (Clandinin and Connelly, 1994). These data contained many
elements of depth and breadth (Cohen et al, 2000, p279) and flowed in part from the context (Sayer, 1984, p51) and the time of their gathering (Sayer, 1984, p213; Mercer, 1991): that room with these interruptions on that date.

The interviews exposed issues of personal trust (Clandinin and Connelly, 1994) where the interviewee and interviewer both actively and reflexively presented and withheld data and created meanings individually and together (Cohen et al, 2000, p267). These raw data also came in a human and technical language (Holstein and Gubrium, 1997) that was presented as a raw, unstructured experience (Atkinson and Hammersley, 1998; Patton, 2001, p343) that could be difficult to organise beyond the bounds of CS knowledge structures (Silverman, 1993, p197). There were also an admixture of sentiments and attitudes (Cohen et al, 2000, p271), recollections, points of view and interpretations (Whyte and Whyte, 1984, p120), beliefs, knowledge and perspectives (Altheide and Johnson, 1994) which were then subsequently transformed by myself in being transcribed (Baker, 1997; Punch, 1998, p60) and reduced, before and during analysis and reporting (Tizard and Hughes, 1991).
4.6 Analysing the Data

You don't need to get to the bottom turtle to have a valid conclusion. You only have to get to a turtle you can stand on securely.

(Joseph Maxwell, 1996, p87)

In reading the literature on analysing qualitative data it quickly becomes clear that there are no exact ‘how to’ guides for general or specific approaches to case studies or interviews. In the regular warnings to the tyro it is posited that very few part-time researchers would be capable, in terms of the space available for data analysis in work and life, to gain such a full understanding of all the methodological issues to allow even a personal, definitive hold on qualitative data analysis to the extent that, given a scenario, I, or anyone else, would be able to say, ‘that’s the way to do it’.

There are general texts (e.g. Marshall and Rossman, 1989, pp52-4; Robson, 1993, pp370-407; Silverman, 1993, pp144-70; Yin, 1994, pp102-6; Blaxter et al, 1996, pp173-206; Maxwell, 1996, p87; Wellington, 2000, pp133-50; Holliday, 2002; Denscombe, 2003, pp267-83), and exemplars of data analysis (e.g. Lynd and Lynd, 1929; Pressman and Wildavsky, 1973). Most do not prescribe any particular approach, end even when they do so, only as a help, never as an explicit how-to instruction set. I agree with Janesick (1994) that,

there is no best system for analysis. The researcher may follow rigorous guidelines described in the literature ... but the ultimate decisions ... reside with the researcher.

123
So I faced my data with Peräkylä’s (1997) advice that there is no single, coherent set of ‘qualitative methods applicable in all analysis of texts, talk and interaction’.

It was not enough for me simply to restate what the respondents had said. As Whyte et al (1984, p228) critiqued, ‘you have given me a description [when] ... I asked for an analysis’. The raw data came in digital sound recordings, accompanied by notes, all in the form of words, as unstructured conversations one-to-one between participants, and not just words but also as silences, sighs, outbursts, avoidances, expressed emotions, etc. all of which had meaning.

When tackling this, in analytic terms, chaos of words and meanings the researcher must find a place - a turtle - on which to stand. I had to find a general strategy and approach (Whyte and Whyte, 1984, p228), one where the evidence was tested fairly, where the conclusions were compelling and where alternatives had been also considered, and rejected. Yin (1994, p102) argued for two possible general approaches to case study data analysis: theoretical or illuminating. I opted for the latter, a descriptive approach simply illuminating the case as a case without drawing wider conclusions.

Only once such a general strategy is in place does Yin (1994, p104) recommend the researcher then begins to closely attend to the data. These data had to be thought about, immersed within, read, and re-read, listened to and listened to again. As the individual interviews became ever more familiar, so the need to be intuitive, aware, insightful and inspired allowed me to let the data speak, not in an unstructured way (Yin, 1994, p106) but with care (Denscombe, 2003, p271).

Although there are many different ways things can proceed towards an analysis, I opted to follow a commonly recommended one: that of coding the data in order to make its analysis possible and clearly meaningful to the intended audience. The aim was to draw conclusions on the meanings hidden in the data (Blaxter et al, 1996, p176) whilst taking Punch’s advice to, ‘distinguish the incidental from the essential; neither divide the indivisible nor lump together the divisible or heterogenous’ (Punch, 1998, pp203-4) through the use of meaningful encodings in an iterative and refining process of data classification and inter-data relationships.

Interestingly, the literature gives little hint at when an analysis is complete. On reflection I would not put this down so much to glaring gaps in these authors’
knowledge, but probably more to an understanding that the end is itself a point as undefinable in advance of analysis as is any specific approach itself for any specific project. However, the final analysis should be valid and reliable (Sayer, 1984, p82; Altheide and Johnson, 1994; Seale, 1999; Wellington, 2000, p29ff) and I believe I do present an analysis that has avoided drawing conclusions too quickly, nor treating the data as a definite statement of the situation. Instead it is an analysis that reflects the ambiguity and contradictions within the data, and, does not over-emphasise that which is weakly present (Silverman, 1993, p200).

I have made it clear that I am present in the data in this study (see §4.4). This was taken into consideration as the analysis proceeded in order to, for example, attempt a separation between objective analysis and subjective presence by treating, whenever possible, my words as the same as those of the other conversers (Kanpol, 1998; Patton, 2001, p495; Denscombe, 2003, pp273, 281) and also by my analyzing the data after leaving the case-site. Again, there are no hard and fast rules, so I have taken care to avoid the pitfalls and traps indicated by such writers as Robson (1993, p242) in terms of rigour, detail, trustworthiness, reliability and validity. However, some questions posed on this issue regarding the safety of the analysis are difficult to answer absolutely. If someone else analysed the data I gathered would they get the same categories or conclusions? Possibly or even probably not. However, in the fullness of this dissertation I also present evidence of the analysis process, including such as notes on early encodings (see appendix E), ideas discarded, negative instances, outing of biases, field decisions (see appendix B) and a transcript\(^1\) (see appendix D) as an audit trail which should make the analysis process more visible (Altheide and Johnson, 1994; Denscombe, 2003, p274).

Finally, there is the issue of how to go about the analysis. There is a choice of approaches which include using paper, word-processors, and specialist software analysis tools (Holliday, 2002, pp47ff; Denscombe, 2003, pp275-6). However, as Patton (2001, pp442-7) put it, ‘the real work takes place in your head’. Having had long experience as a Computer Scientist with a wide range of software tools which have attempted to assist semantic analysis from raw data, I opted to use paper, scissors, glue-sticks, highlighters, the floor and pens, and so to perform the analysis as much as possible in my head.

\(^1\) Note, however, that the case's place and person names are not recorded in this dissertation
4.7 Ethics

We are discussing no trivial subject,
but how a man should live.

Aristotle, 'The Republic'

The research project could have been covert (taking place without the knowledge of significant people involved in or affected by the study), open democratic (allowing a to-ing and fro-ing between the researcher and the other participants) or autocratic (allowing little or no feedback loops from the other parties in the research process) (Hudson, 1986). The approach chosen depends on the individual research project and even then each stage of the research can have its individual ethical challenges (Scott, 1996). No 'pure' approach can be defined as social research is never politically innocent in situ nor humanly neutral to those involved; indeed it may hurt or heal, but even these terms are not so simply mapped onto 'bad' and 'good' (Cohen et al, 2000, p49). The principle of, 'don't harm people and keep true to the research aims' is admirable, but does not easily become real research in the field (Patton, 2001, p406).

The participants in the study were Computer Scientists, not qualitative educational/social researchers (Cohen et al, 2000, p49) so there was an issue of communication - that is from the researcher to the participants and from researcher to the readers of this study. The participants were unlikely to understand such a sentence as Levinson's (1988), 'I offer the following critical questions as a prologomenon to any reflexive and socially committed research project', nor may many readers of this report understand the principles of object orientation or stack-heap buffer overflow. There was an onus on myself as the researcher to
communicate ideas and principles across the epistemological boundaries of the participants, the researcher and the report's audience (Levinson, 1998).

There are significant official and academic sources of advice on following a research process ethically. For example, from the American Psychological Association (2002), the British Psychological Society (2008), British Sociological Association (2002), American Educational Research Association (Strike et al, 2002) and from others (see, e.g., Whyte and Whyte, 1984, pp193-224; Robson, 1993, pp29-34, 51; Blaxter et al, 1996, pp145-8; Punch, 1998, pp281-4; Patton, 2001, pp405-6). These made clear my responsibilities to the participants involved, also that all should be informed of the purpose of study as part of the Doctorate in Education programme at The University of Edinburgh, and the nature of the study as a CS research project; but these not insofar as to compromise the study (Judd et al, 1991, p485).

All access was negotiated with the interviewees and with senior departmental staff (Frankfort-Nachmais and Nachmais, 1992), including return visits (Walford, 1991a). I could not assume that I could turn up and demand time with anyone. Those involved were actively giving permission – which, after analysis of the pilot (see §5.3) did not mean a long form to sign (Maxwell, 1996, p66) - and they were led to feel that they were dealing with an honest colleague/researcher who would treat them, their words and their workplace with respect.

The participants were allowed to take some power during the research: to say yea or nay, to choose times, to read or not read transcripts, to comment on the progressing study, to know their words were (within bounds of reporting) sacrosanct to our joint confessional, to feel free from pressure, to feel respected, and, most importantly, to be allowed to speak what they thought rather than simply to be 'led' by myself as the researcher.

There were such other issues as the risk of potential defamation or negative effects upon participants' employment (Cohen et al, 2000, p71; Patton, 2001, p409). Our employer, at department level, was kept aware and gave permissions for the study and, wherever possible, anonymity was used to ensure minimal negative feedback to participants, the researcher, the case study site or the University of Edinburgh (Brooks, 1989). After the research was concluded and I had changed job I made myself available on a further two occasions at the site in 2007 (Cartwright and Seale, 1990).
I have sought to balance data confidentiality with data analysis; not to say enough to allow the reader who is interested solely in the analysis to accidentally locate the study; but, yes, there is enough information presented in this dissertation to allow the reader who kens the small world of Scottish academia well to uncover unspoken specifics of the site (Burgess, 1985; Clandinin and Connelly, 1994; Maxwell, 1996, p67; Patton, 2001, p405; Holliday, 2002, pp155-6; Denscombe, 2003 pp134-41; Neill, 1968). With the passage of time since the data gathering and the continued rapid rate of change of Scottish academia, any adverse effects will be further moderated.

There are silences where I have not given the whole story of my being in situ (Becker and Richards, 1986, p108ff), my exiting the site, and how I went about exposing to my then-colleagues obscure (to a Computer Scientist) data analysis techniques. I was careful not to create a situation where any participants could veto any part of the work (Whyte and Whyte, 1984, p219; Wellington, 2000, p55) in recognition of the responsibilities I have to myself and the reader (see, e.g., Reynolds, 1979; Judds et al, 1991). And finally, the researcher is and the reader should be aware of the limitations and purposes of this as of any study: not everything could be known, the data gathered was limited, this reporting is subject to self-editing and I performed censorship (in the limited sense of removing references to places and persons) of what was said by participants and myself in polishing the final report (Walford, 1991b; Whyte and Whyte, 1984, p201; Walford, 1991a; Wellington, 2000, p55).
4.8 Drawing Conclusions and Reporting Results

Even though the information collected on a single classroom group ... is extremely rich, the basic fact remains that for a single-classroom study, N = 1 ... [it] may help to generate hypotheses ... it does not allow for generalizations and broad conclusions

(Feinberg, 1977)

This research report is a partial account, limited by external factors. It is an edited and filtered version of the truth written with the benefit of hindsight (Denscombe, 2003, p285), researcher and researched influenced, a result of the practical and ethical limitations of the people involved, of its time, and individual (Wellington, 2000, p180). A lot of effort has been expended in creating this study and its final report, so, what is its use? (Wellington, 2000, p166ff) It is presented to you as part of an ongoing debate, increasing the choice of options to those who find it relevant, and hopefully, offering a new perspective on what I found to be a relevant issue: academics handling change (Silverman, 1993, p186).

In terms of the conclusions that can be drawn, there can only be some general ones (Gomm et al, 2000). However, the potential audience for such a study into a field at university level is large and could include CS colleagues, government policymakers, other universities, curriculum designers, the public and the press (Howie, 2006; Nell, 1968; Mortimore, 1991; Punch, 1998, p266). Can or should such a single report as this be readable by all of them? The answer is 'probably not' (Yin, 1994, pp129, 152; Blaxter et al, 1996, p221; Denscombe, 2003, p286) or even 'no' (Clandinin and Connelly, 1994; Wellington, 2000, p154). However, the reactions of other accidental stakeholders should be considered as this report does
contain sometimes sensitive, controversial and critical statements (Mortimore, 1991).

With these further thoughts in mind it is clear that the writing-up of academic research is a complex task and that really good research and its reporting are hard to pin down to a set of easy rules (Wellington, 2000, p151) of what to reveal and what to conceal (Donmoyer, 2000). As a case study this is,

generically a story; it presents the concrete narrative of the actual, or at least realistic events, it has a plot, exposition, characters, and sometimes even dialogue (Boehrer, 1990).

This report attempts to tell clearly all that has been done, how and why, and to let the participants be allowed to speak in a way similar to those in the studies of Woods (1979) and Griffin (1985a, 1985b) which inspired me.

This report should work on several levels. At the macro level (Hollliday, 2002) I hope that it shows the power structures behind the study, the voices of those interviewed, the micro-politics of the setting, and how the study was a product of negotiation. At the middle level, the people concerned - the participants - are presented as acting within the study’s authority structures and as the visible means of marshalling evidence. At the micro-level of wording (Webb and Glesne, 1992; Usher, 1996; Holliday, 2002; Denscombe, 2003, p289) I confess to finding a mix ‘n’ match appropriate, sometimes hiding, sometimes revealing the author. The other participants are shown as persons and where their words are interpreted I have tried to make this clear (Denzin and Lincoln, 1994.) The aim is a report that is accessible, seen through the author’s eyes, but not defensively (Usher and Scott, 1996) where the others who participated in the study also talk clearly (Donmoyer, 2000).

Yet, this is not simply a positivist, nomological, generalisable study which produces clear results that are unrestricted as to time and space (Stake, 1998; Lincoln and Guba, 2000) and which can be used to predict and control educational behaviour free from time and context. No. This case study is a production of problematic knowledge (Kaplan, 1964; Bryman, 1988, p88) of a quite different kind: ‘experiential knowledge’ (Marshall and Rossmen, 1989, p147), ‘personal knowledge’ (Donmoyer, 2000) and my own ‘general contribution to knowledge’ (Polanyi, 1958).
The findings and data presented here are mineable for possible later uses that are not yet defined (Donmoyer, 2000), and which may or may not be of a very similar kind to the case I investigated. After reading the many works on case study, and performing the study, I still believe experientially that it is doubtful that any single case study can be generalised directly to another, even to a similar, situation. Instead, if this is a good qualitative case study, as Stake (1980) stated, epistemologically in harmony with the reader's experience ... [it] thus [can] be to that person a natural basis for generalization, through its verisimilitude (Stenhouse, 1985) and resonance with the reader, giving him a feeling of being in and with the author in perhaps a similar way as does a good novel, video game or movie.

While I accept that the wider relevance of any case is questionable (Punch, 1998, p153) and remains a matter of judgement (Bechofer and Paterson, 2000, pp47-9), such a single case study as this one may extend the repertoire of what is available to practitioners (Stenhouse, 1985) that are posing and seeking to answer new questions. Donmoyer (2000) talked of his personal experiences as being individual yet also cumulative as what he learned from each job (or 'case') he took on to the next one in a reflective and reflexive practice (see also, Schön, 1987b and Lave & Wenger, 1991). In a case study one man's written reflective experiences can become a knowledge short-cut for others. As an oldish Computer Scientist of 1970's mainframe vintage I am reminded of an aphorism once attributed to IBM: whoever is first gets the glory and whoever is second gets the profits.
5. The Method

Two important statements have to be made at the outset:

1. what were the research questions?

2. what is this a case of?

The research questions were:

- What is happening to knowledge production in CS?
- What kind of knowledge field is CS?
- What underpinning beliefs about CS and academia do academics in this site hold?
- What processes of teaching and learning follow from these beliefs?
- How far have things changed over the years?
- Where is change driven from?

This is a report of a research project that was an investigation of how a situated group of professional people handled change in their field of work. It is an investigation of a single bounded case and as such it is set in a particular time and place.

The setting was a Computer Science (CS) department in a 1990's university in Scotland. The site was therefore primarily one of a teaching department which
aimed at staying close to commercial relevance in its teaching in order to increase graduate employment.

The time was not so much a date, as a window of opportunity with a specific group of academics. These academics, six in all, had, when combined, over 200 years of experience in the field and represented the core of the established department. They had worked together for decades but would soon disperse due to retirement, redeployment or resignation.

The time of data gathering was therefore a significant opportunity to gather, from these academics in this setting, their reflections on handling change in their field, and to investigate their understandings, reflections and coping strategies on being a CS academic.

This report is an analysis of the words of these people set against a background of historic CS change, recent educational sector change and the nature of knowledge.
5.1 Introduction

"We were in an environment that was almost like a bubble within the university" - research study participant

- as reported by Schön (1987b, p341)

Having read and digested a range of texts on methodological issues (see chapter 4), I set myself the task of designing a study that would work in such a way as to provide answers to these research questions. I was particularly inspired by the accounts of educational case studies given by such authors as Coleman (1961), Neill (1968), Woods (1979) and Griffin (1985a) (see §4.3 above.) As I read these accounts it became clear that there were strong possible parallels between their research aims and my own: a seeking to answer particular, real-world located, educational research questions. Significant parts of the approaches they each took also appeared to be generally suitable: to understanding the perceptions as expressed in the words of those located within a single, interesting teaching site and to report back an analysis.

These inspirational studies all took a similar general methodological approach of using a case study and particular rich-data gathering techniques in a bounded setting, that together created an individual study from which significant data were gathered for complex analysis and clear reporting. A parallel case for the purposes of answering the above research questions would be an investigation of a bounded case of a particular group of Computer Science (CS) academics, working together at a single site, with significant experiences of change, who expressed themselves well enough for rich data to be gathered for analysis and reporting (e.g. as reported by Schön (1987b, pp327-43).)

These, to me, inspirational educational case studies also placed an emphasis upon setting the case against a relevant background in order for the analyses of data gathered to be set within wider contexts. It became clear that the site where I
then worked was a clearly bounded case that was methodologically suitable for a Coleman, Neill, Woods or Griffin type of single case study approach. Further, as this was within a 1990’s university CS department the relevant background to the participants’ words would be found in issues of the time, sector and academic field as these relate to this particular site.

CS is a relatively recent field (see chapter 3), dating back, effectively, only to the mid 20th century. Such long-established CS academics as those involved in this study have seen a significant time-slice of the development of this subject in their often 30 or more years of working in the field. So this study of how CS academics handle change had to be considered along with the simple fact that CS grew from a seed, to a shoot, to a tree and into a forest in their work lifetime. As certain effects of change will be unique to academic CS the study is set against the background of the general growth of CS as a field and the life of CS as an academic subject.

Further, these were academics in a particular setting: Scottish academia, with its own distinctive history, degrees and legal regulation. At the time of the study there were clear lines in terms of funding, expectations, teaching workload and sectoral control of the entitled ‘universities’ when compared to other Scottish educational establishments such as the further education ‘colleges’. I decided the study should be related to what was known about the particularities of CS in the Scottish university educational context.

Finally, during data gathering and in early analysis it became clear that CS as a taught subject was dichotomously grounded on substantial theory and in applied practices. This created a significant tension - not unique to CS - that reflected meta-knowledge issues that affected the working lives, beliefs and practices of these teaching CS academics. And so at this stage of the study it was decided that further background on the nature of knowledge in CS was required in order to discuss these tensions as they arose in the data.

In summary the general methodological approach chosen was one of,

• A single, bounded case study situated in my then workplace;

• Based on one-to-one interviews with a close group of highly-experienced CS staff;

• Analysed by qualitative analysis via coding of the interviews;

135
• Supported by reference to secondary data on the historic and current situation of CS as a taught subject in Scottish universities;

• Set within the context of ongoing sectoral change;

• Against a background of the recent growth and development of CS as a field;

• With reference to discussions on the changing meta-knowledge approaches in teaching of relevance to CS.
5.2 Data Sources

The following are the sources of data used to inform this study:

Secondary data –

CS in the archived prospectuses of five Scottish universities (one ancient, two 1960's and two 1990's founded) in order to find a consistent, high complete, available and close to comprehensive information on some undergraduate degrees and other studies in Computing (see §3.4);

Recent UCAS, HESA and Scottish Executive data on applications, enquiries and registrations to study undergraduate Computing degrees at Scottish universities (see §3.3);

Freedom of Information requests to all the Scottish universities to provide information on admissions to study undergraduate Computing degrees at their university (see §3.3).

Primary data –

One-to-one, semi-structured/unstructured interviews of around one hour involving six long-established academic Computer Scientists at a single, 1990's-entitled Scottish university.

The data were analysed and reported in a mix of qualitative and simple quantitative ways, with the interview data relating to the university concerned and its academics being anonymised.
5.3 Secondary Data - Undergraduate Studies in CS at Scottish Universities

Visits were made, over a period of weeks in autumn 2006, to the archives at the libraries of a range of Scottish universities. Having studied CS and/or worked as a Computer Scientist at five Scottish universities, and interacted professionally with the CS departments of five more, and with mergers and demergers amongst universities and colleges during the period when CS began to appear at the Scottish universities, I saw no prima facie case to select out particular universities or kinds of university (such as the one that was the site of primary data gathering for the case study). Accordingly, I choose to visit a range of universities, covering both age (ancient, 60’s and 90’s) and geography (West, East and North), (listed alphabetically): The University of Abertay Dundee, The University of Dundee, Glasgow Caledonian University, The University of Glasgow and The University of St Andrews.

I found there all the extant copies (either complete or close to complete sets) of individual prospectuses of the universities, and their precursors (Queen’s College Glasgow, Glasgow Polytechnic, Glasgow College of Technology (but not The Royal Technical College, a precursor of Glasgow Caledonian University and Strathclyde University, as these archives are held by the latter), Dundee Technical College, Dundee College of Technology and Dundee Institute of Technology.) After discussions with the archivists, I decided that looking directly at the old prospectuses would provide the most comprehensive and consistent overview of CS as an undergraduate taught subject. I acknowledge that this means there is some lack of depth in this part of the study, but I believe this is more than compensated for by the breadth of data in the nigh 100% coverage provided by the old prospectuses, particularly as the older prospectuses often gave more detail on non-graduating short courses which showed early evidence of the rise of CS. Studies in taught postgraduate (e.g. postgraduate certificate, postgraduate diploma and taught master of science) were not considered, however below-degree awards (e.g. individual modules, higher national certificates, higher national diplomas and advanced national diplomas) were considered.

The prospectuses were read and notes written in a plain paged Moleskine bound notebook (see Appendix B). The author used his own knowledge of CS to identify
all occurrences of relevant teaching delivery in the prospectuses and so to show the
growth, diversification and/or decline of the subject as a named delivery.

Reference was also made to on-line available HM Government agencies’ data on
applications and enquiries to study CS and using UK or Scotland-wide figures, as
available and as appropriate, for the subject. Requests were also made to all the
Scottish universities for data on applications to study undergraduate CS (see
Appendix C). The UCAS figures are most difficult to interpret as their categorisation
of degrees relating to CS has been regularly reviewed during the past few years,
with subjects entering and leaving the overall ‘Computing’ classification; also,
degree subjects with very similar contents are given quite different UCAS numbers
and some quite different degree subjects are sometimes clumped together. There
was also some missing and incorrect data from the replies from the universities.
The data from Herriot-Watt University did not look correct when its trend was
compared to that of each of the other universities which replied, and when
compared to the national trends so I moved their data forward by one year (this is
also noted in the analysis.)
5.4 The Pilot Study

The pilot study was run, formally and informally, over a period of around six months from late 2005 to early 2006. It was used to test the potential usefulness to the study of the group interview, individual interview, semi-structured interview and unstructured interview. It was also used to test who were the most useful people to talk with, how and where.

A semi-structured interview was set up for December 2005 with a small group of students, invited and then self-selected, from a general call to all students reading a range of CS degrees, at the same university at which the academics were later interviewed. The invitation was made formally in a short email to the students. Eight responded and six turned up for the group interview. The interview took place in a bright, comfortable, recently-built presentations and staff/research meeting room around a large table. A colleague with recent experience in interviewing and a PhD in a Social Science helped. The students were provided with food and drinks. They were also given a one-page document for both parties to sign which explicitly laid out my duty to the individual student-participant and his or hers to the study. All signed, but some extra explanation was required at the time. The actual group interview was based around a set of open questions which the students were able to discuss. The process took around 1.5 hours of talking recorded on analogue audio tape and by digital recorder, both of which were visible on the table.

Despite trying to make everyone feel comfortable, there was a feeling of confrontation in the interview. The students never really relaxed into simply conversing, especially the two 1st year students who said very little. The conversation was dominated by two students, one of whom already knew me very well. Both ‘talkers’ were unusual, one being a Greek mature female student, the other being a French student; but they did bounce ideas and questions between each other as they talked. No-one consumed any food or drinks. The data gathered were remarkably rich, but were felt to be not entirely relevant to the aim of the study, being the concerns of tyros in the field of CS rather than ones related clearly to issues of change in the field. The two recordings also proved to be
differently useful: the digital recording was far easier to transcribe whilst the analogue recording was clearer to listen to. The transcription process proved very worthwhile as I did it myself and heard the data at first hand, but was also rapidly overwhelming as the amount, and richness, of the data showed that a large number of hours of such conversation could prove nigh impossible to process within the structure and limitations of the proposed study.

The second stage of the pilot was more informal. I went around chatting to colleagues in the workplace about how they were handling change. A critical moment in the study design was when one notably said,

'I don't!'

It became clear that those colleagues with the longest experience in CS were very willing to talk, comfortable in conversation with the author, confident in their opinions, reflective in their accounts and reflexive in their professional practice, even admitting to such as their failure to adapt. I also found that I was drawn into the conversations, submitting my own reflections and experiences, and so was definitely and perhaps usefully present in the conversational data. We had all worked together for decades, so these were relaxed conversations. It became clear that here was a rich source of data in these conversations, by their structure (i.e. relaxed) and by their nature (i.e. conversational.)
5.5 Some Principles of the Design that arose from the Pilot Study

The study would be performed in digitally recorded conversation, semi-/unstructured interview, with colleagues, of long experience in the field of CS and, in circumstances that relaxed them, probably their own offices. These should be shortish, probably around one hour, one-to-one conversations with a small number of carefully selected individuals. These would all be in the same university and department as a neatly bounded case. Agreement to proceed would be dealt with verbally and would be reached without the use of printed forms, but would be present on the recordings.

It was also tentatively proposed to do one or more group sessions as semi-structured interviews. In the event there was ample rich data within the one-to-one interviews to apply to the research questions so group interviews never took place. It was also an intention to allow the participants to review and comment on the data gathered and the ongoing analysis. However, I left the site to work for a competitor university and so this was not done as return visits to the site became awkward\(^1\). I did pass the transcript of one interview to the Computer Scientist involved but he didn't read it, saying instead much later, 'I haven't yet finished your coursework, John!' It was also clear in practice that the other participants did not want to read the transcripts to check them.

I decided to perform all transcriptions in order to immerse myself fully in the data. In discussion with my then supervisor it was suggested that I also use listening to the recordings as a means of reflection and analysis, in addition to transcription. She also recommended keeping my data gathering to those within the CS staff at this one university in order to retain the integrity of the study as a single case study.

---

\(^1\) On a return visit shortly after I had changed job one senior academic, not from the interviewed group, said to me, with a note of seriousness, 'How did you get past the cannons at the front door?'
5.6 The Data Gathering

Five individual CS lecturers were invited to take part in the study in addition to the participant-researcher. Those involved were a Reader, Senior Lecturers and Lecturers. They were aged between 48 and 59. All had been employed as CS academics at this one university since the 1970’s or 1980’s. Only one had had experience as an academic elsewhere (Senior Lecturer in a further education college fifteen years previous to the study.) All were male. None had a first degree in CS, these being variously from the Mathematical, Natural and Engineering Sciences. Three had worked for a time as professional CS developers before entering academia. Three were qualified to PhD level, two to Masters and one with highest award at Bachelor’s level. A selection of the UK universities where they had studied include Glasgow, St Andrews, Edinburgh, Cambridge and Durham.

The invitation to take part was by personal, verbal invitation for four of the participants, and by email to the fifth. They were each told of the purpose of the study (in short, to try to understand how they handle change in CS), its structure, their rôle, anonymity, opportunities for feedback, etc. All agreed to take part.

Each interview took the same format. It took place in the participant’s office at a pre-agreed time. It took no more than one hour. The digital recorder was set between myself and the participant who simply sat and faced towards each other, usually with no desk or table between us. With the recorder running, I explained the rules and responsibilities again and asked the participant to agree.

There was one advance question and one closing question. The advance question was always the same, and grew out of the first interview, “How did you get into Computing?” The closing question was specific to the participant and grew out of my own reflections on the individual and on the conversation. Apart from that I used my existing skills as an interviewer to allow the participant to pick apart his understandings of change in CS. No attempt was made to drive the conversation by the interviewer, except in order to allow this personal unpicking to continue.

The degree of personal chat varied. None of the conversations stalled. All began easily, went comfortably, and ended easily too.
The conversations were recorded using an Olympus DS-600 digital voice recorder. The data were downloaded onto an Apple Mac mini and an Apple MacBook for storage, listened to (using AKG-44 headphones) and transcribed using Apple Pages.

When Denscombe stated that, 'in an interview neutrality is a chimera' (quoted in Cohen et al, 2000, p121) I take it that he was implying that interview-based studies gather together a variety of different data gathering problems. Participant error was reduced by using a range of similar people interviewed at different times and in different places, by the researcher having a similar knowledge of the research setting as the other participants, and the participants having a clear and similarly-founded understanding of the areas being discussed. Participant bias to the interviewer was reduced by the use of a comfortable interview approach by a colleague in the participant's own office.

Observer error was reduced by using an interview approach that encouraged the participants to speak freely and conversationally in the same duration of time each time. Construct validity was assisted by the participants' immersion in the interview topic and by reference to background materials on CS in general and in academia. External validity is shown in my being entirely open in this dissertation about the group, the setting, the site and field history and about the research processes, to allow the reader to see what the raw data was like, how it was encoded and how quotes were selected (see also appendices B, C, D and E).
5.7 Data Analysis

Advice on coding and analysis of the data was taken principally from Punch (1998) which I found to be a lucid text on the overarching approaches and principles of qualitative data analysis, Patton (2002) which was useful at explaining the thinking behind different approaches to the analysis, and, Miles & Huberman (1984) for their clear and specific details of how to perform the analysis. Each interview was listened to at least twice and read several times during coding. Coding was performed in three phases (see appendix E) to get a feel for the meaning in the data. The data were then gathered into codings using the word processing package, printed out, cut into small slices (often duplicated into different sections), then pasted into an A2 bound notebook with all similarly encoded text fragments brought together onto the same page(s). This notebook was then read and reread several times to identify significant issues indicated by individual text fragments, then the fragments were filtered to identify those to be selected and used as exemplars in the data analysis for this dissertation. By the end of this process I felt I knew every word in the interviews and could find individual quotes fairly quickly in the notebook.

The data analysis was performed with due respect to the contents of the interviews as real conversations between trusted colleagues. These were easy conversations, not unlike those that would take place regularly in a common room atmosphere. The importance difference is that I was aware of being the interviewer/researcher and the interviewee was made aware (I cannot tell to what extent he was always so aware during the conversations) of being a participant. However, I had tried not to lead, nor get in the way, nor to stop myself talking when I felt I had a useful part to play in what was always meant to be a dialogue, not a monologue. As such, I was not a simple sounding board, this was a social interaction aimed at gaining substantive content and my words were as much part of the conversation and understandings as those of the other participants as, with respect to our place in the setting, we were equals.

As I was present five times and each other participant was present only once, does this mean the data are skewed towards myself? I believe not so for several
obvious (on reading the transcripts) reasons. Firstly, I tended to repeat myself in what were, to me, repetitive conversations so much of my input was naturally elided on analysis. Secondly, this gave the other participants, especially after the first interview, an increasing slice of the conversation and, in one, I found myself a minor participant. In this I was using previously acquired skills as a press reporter for newspaper, radio and television: probing and asking in order to bring out from the other participant what he felt to be the underlying issues and how he reasoned his way round the issues.

Yin stated that,

For some topics, there may be no other way of collecting evidence than through participant-observation. Another distinctive opportunity is the ability to perceive reality from the viewpoint of someone “inside” the case study rather than external to it. Many have argued that such a perspective is invaluable in producing an “accurate” portrayal of a case study phenomenon. (Yin, 1994, p88)
5.8 Reporting

Being a case study, it was decided not to produce any conclusions on the wider meanings of the data as possible generalisations, but instead to report the primary data analysis as the findings with respect to the research questions.

The university and its academic staff in this case are real. The ethics of the approach taken were embodied rather than simply followed from advice available. However, it is worth noting a few significant aspects of the ethics followed. It was decided to follow the British Educational Research Association (2004a, 2004b) guidelines on general ethics and reporting. Principle 21 in the general ethics advice on ‘detriment arising from participation in research’ was followed as all the interview participants knew I was leaving, either directly from me or from the general scuttlebut of an academic department. All were aware that the research was a case study and that, as academics at that university they were participants in the case under consideration. All names were anonymised in accordance with principle 23, ‘privacy’.

The financial sponsors (articles 30 to 41) are myself, the university in which the study took place, and the university I am currently employed. In reporting I followed article 33,

Researchers must avoid agreeing to any sponsor’s conditions that could lead to serious contravention of any aspect of these guidelines or that undermine the integrity of the research by imposing unjustifiable conditions on the methods to be used or the reporting of the outcomes,

as a strong principle, together with article 6,

The Association considers that all educational research should be conducted within an ethic of respect for the person, knowledge, democratic values, the quality of educational research and academic freedom.
As such, neither sponsor university were given a veto or the right to change any part of this dissertation nor did they need to as I also followed BERA’s guideline on reporting research, article 5.3, which states,

The research ethic of respect for truth, or academic integrity, requires researchers to be scrupulous in avoiding distortion of evidence and weakly supported assertions in the reporting of findings.

To slightly alter a Computing Science mnemonic, WYSIWIG: what you see is what I got.
6. Findings

This chapter of the report is based principally on the analysis of the one-to-one interviews involving six, long-established academics in the CS department at a 1990's 'new' university in Scotland. Full details of the reasoning and the process can be found in chapters 4 (Methodology) and 5 (The Method). An example, full transcript can be found in Appendix D ('Dr Blue'). Exemplar extracts from the coding process used to aid selection of the quotes used in this chapter can be found in Appendix E.

The findings from the interviews are supported, as required, by reference to chapters 2 (Literature Review) and 3 (Computer Science Studies at Scottish Universities.)
6.1 Computer Science as a Knowledge Field

In this section the nature of CS as a knowledge field is examined with respect to the conversations with the CS academics. Several areas are discussed, including:

**technical terms** - what did the use of technical terminology, and the terminology itself say about the academics’ understandings expressed through their use of jargon?

**abstraction** - how, when and why did the academics use metaphors and similes to express their understandings of the complex and abstract concepts that are core to the field?

**currency** - on further examination, how did the technical terminology used by the academics illustrate the changing nature of the field?

**theory and application** - how did the academics express their understandings of the divisions between theoretical and applied Computer Science, where do they see the divisions in CS, and how did they assess how this affects the stability of CS?

**rate of change** - what evidence did the academics supply for the rate of change of the knowledge-base in CS and how has this affected those involved in the field

**other fields and CS** - which other fields did the academics perceive as being crucial to the underpinning of a valid subfield of study in CS?
6.1.1 The use of Technical Terminology

Even a quick glance at the interviews (see appendix D) shows long swathes of conversation peppered with technical terminology that is apparently central to being a CS academic. These technical texts were used naturally and as part of the normal conversation between CS academics. Each CS academic interviewed spoke in this way, for example (technical words underlined),

It's a bit object oriented, a bit procedural, there's no protection from memory management or hacking or anything else like that. (Dr Blue)

... this is the way to do loops, decisions and declarations and statements, and this is the way to [do] objects - object oriented approach with inheritance and polymorphism. (Dr Scarlet)

A lot of things don’t matter. Things like savings 2 bytes here and there. Used to be critical ... make sure the words are left-aligned ... don’t use a double use a float ... don’t use four bits, use 2 bits, but if it’s a four-bit word extend it to 4 bits. (Dr White)

In the five hours of interviews, over 700 identifiably different technical terms and over 1,000 different usages of technical terms were spoken by these academics. This indicates a significant meta-knowledge beyond but connected to the technical terminology. For example, Dr White (see paragraph above) discussed 'things that don't matter' and which 'used to be critical' via the use of 'bytes', 'bits' and 'words'. He was talking through these technical texts about the lack of a need to teach aspects of data storage and of handling efficiency in fast, high-memory computers; it was in essence a discussion about the loss of the need to teach some aspects of CS theory. Large chunks of conversations would be, I propose, nigh
incomprehensible to the reader who does not have a CS background, or one that is sufficiently similar to that of the older CS academics interviewed.

The breadth of terminology and range of concepts in the interviews illustrated and indicated that CS, a relative newcomer to academia, is nonetheless a field that has significant depth and breadth (see §3.2 'The Developing Discipline'). It is noteworthy that at no time were any of those interviewed posturing or showing off their knowledge. The extensive usages of complex, technically-populated conversations on a range of high-level concepts shows that a broad vocabulary and extensive field knowledge were seen by these CS academics as a necessary part of explaining experiential, pedagogic or epistemological concepts in CS. For example, Dr Blue explained and illustrated one of, to him, the most significant differences in experiencing the field by academics and practitioners as follows (again, technical terms underlined),

I'd been working on a big project that had kinda come to an end anyway. So, what was I going to do. So, they had this kinda one-off program that they were needing written in PL/1. You can knock that off in a week or two, you know. So, sure enough, it was just a week long program, it wasn't a big one at all. And I decided to do it going back to my university knowledge, going back, doing it properly, so I wrote it in PL/1, parameters, procedure calls, there wasn't one goto statement in the whole program, while loops, beautifully structured. And I thought, an academic couldn't have faulted it any way, between the looping, the parameter passing, all the rest of it. Compiled it, ran it, it worked perfectly. ... About a month later my old boss telephoned me and said, we're needing your program and we've got this same situation again, but your program is almost what we want, except, it's that badly written that none of us have a clue how it works. Now, that was his words: 'badly written'. And yet I would still maintain it was beautifully written. But they were used to flowcharts, GOTO's, they didn't understand parameter passing, cos they'd never met it, you know. I sometimes wonder, like, that's one of the reasons I've been able to move. I think, the science, if you like, the underlying principles that I got in the 70's, I don't think have changed, until today. Except for object orientation, which we didn't, although, even that you did actually get a sniff of, because there was a language called Simula-67, which was object.
oriented and had virtual functions and everything in it. So we had a
sniff of it there. But, everything else. I'd hate to see some of these
programmers who were programming in the real world in assembler and
in PL/1 with GOTO's going all over the place, how they coped with going
into a language like even C, but certainly going up into the C++'s and
the Java's, where you were forced to be structured. That must have
been a huge conceptual jump for them.

This highly technical extract also illustrates the gaps between the theoretical and
practical in CS knowledge, (Gibbons et al, 1994). Dr Blue took a theoretical stance,
close to Gilmour (2003) whilst he reported that the practitioners took a stance
6.1.2 Abstraction

Much of the CS knowledge evident in the conversations is of an abstract kind (Polanyi, 1967, 1969). This is illustrated in the regularity of the academics talking either in metaphors or similes to explain an abstract CS concept in conversation or to show how they explained certain of these concepts to students. For example, in discussing the strength of the C programming language, Dr Scarlet compared it first to a motorbike, then to the English language,

Well, I don't know, it's a bit like motorbiking. It's a bit like a Triumph Bonneville. Something old and kind of vigorous. ... It's tremendously expressive. It's like English. English allows you to say things in lots of different ways.

This use of real-world parallels to illustrate abstract concepts was also made by the other academics. Dr White discussed how he taught the concepts of object oriented analysis and design using board games in order to help students visualise and understand the abstract nature of the concepts being discussed,

I show them how to do object orientation using a game of Monopoly. Because everyone know Monopoly. All have played Monopoly. And they see that there are slight variations on the theme ... you get your Park Lane's and your Mayfair's, but you also get, I mean, my daughter has the American one, the New York one.

Dr Green used a library as an example to explain to students the complex abstract concepts involved in defining the overall structure of a database and how it is accessed by a computer program,

I like the library. Because you can handle multi-dimensional arrays. You can extend it, so, you've got a bookshelf, its got lots of books in, but the books are different. You know you can get to the bookshelf,
because you can say, I need to get the fourth book from the right in this shelf which has got all of these books on the same subject.

Dr Blue also used the example of a natural language. However, whereas Dr Scarlet used this to explain a computer language concept, Dr Blue used human language to help explain the need for verstehen in getting to grips with the fundamental meanings of object orientation. In this extract he explained how he understood the relationships between the three different but related concepts of object orientation, procedural programming and system design,

Oh yeah, I got it. I get it. But, I think it is a similar kind of question to asking if you are a fluent French speaker: are you thinking in French or English? You know. And I think my answer to that would be, I get it, I get it in both perfectly, and I can think in both.

These illustrate the working of the site’s Department as a discourse community (Fraser, 2006) illustrated by these combination of highly technical words and real-world metaphors which pepper the CS educators’ conversations.
6.1.3 The Currency of Terminology

The technical terms used by the interviewees can be divided into established, obsolete and emergent. The division of these can be seen in figure 6.1.

Figure 6.1 - the currency of the 714 different technical terms used by the CS academics

On analysis, only half of those terms used in the conversations were of established use in The Department's teaching, one third were educationally obsolete at the site, and a further one sixth had very recently emerged into use by these academics. Yet, each word was used with a precise meaning as an indicator of wider and deeper concepts.

A term can said to be established (see table 6.1) once it had become of normal use in The Department, and was a generally-accepted CS term (e.g. 'programming') or it could be one that is specific to The Department but unlikely to be found in use in the wider CS academy (e.g. 'games guru').
<table>
<thead>
<tr>
<th>general CS term</th>
<th>not a CS term</th>
</tr>
</thead>
<tbody>
<tr>
<td>algorithms</td>
<td>bioinformatics</td>
</tr>
<tr>
<td>cache</td>
<td>MCV</td>
</tr>
<tr>
<td>hello world</td>
<td>Edge</td>
</tr>
<tr>
<td>object</td>
<td>hydro-electric</td>
</tr>
<tr>
<td>UML</td>
<td>art</td>
</tr>
</tbody>
</table>

Table 6.1 - a selection of established terms used by the CS academics

A term in the conversations can be deemed **obsolete** either because it has generally fallen out of use in CS (e.g. ‘Data General’ is the name of a once significant computer manufacturer) (see table 6.2). Or the term can still be current in wider CS but no longer in use in The Department (e.g. ‘COBOL’ is a programming language no longer taught but is still used by the Finance industry).

<table>
<thead>
<tr>
<th>totally obsolete in CS</th>
<th>still used in CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 hole tape</td>
<td>bitwise AND</td>
</tr>
<tr>
<td>Autocode</td>
<td>mantis exponent</td>
</tr>
<tr>
<td>Rekursiv</td>
<td>program counter</td>
</tr>
<tr>
<td>Teletype</td>
<td>trees</td>
</tr>
<tr>
<td>VAX cluster</td>
<td>Z80</td>
</tr>
</tbody>
</table>

Table 6.2 - a selection of obsolete terms used by the CS academics

Or a technical term can be deemed **emergent** (see table 6.3) either because it has very recently entered The Department’s vocabulary but is a widely accepted CS term (e.g. ‘computer organisation’) or it may be generally emergent in CS (e.g. ‘WIFI’).
<table>
<thead>
<tr>
<th>new to The Department</th>
<th>new to CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>back-end scripting</td>
<td>3G</td>
</tr>
<tr>
<td>C Sharp</td>
<td>ethical hacking</td>
</tr>
<tr>
<td>server-side scripting</td>
<td>smart systems</td>
</tr>
<tr>
<td>timebomb</td>
<td>virtual bonking</td>
</tr>
<tr>
<td>WiFi</td>
<td>Games guru</td>
</tr>
</tbody>
</table>

Table 6.3 - a selection of emergent terms used by the CS academics

This classification is one that depends on the group of CS academics interviewed at the site. The very fact that they are so experienced, having been in the field for decades each, and that the field is itself only decades old, allows these individuals an insight without parallel in more established fields. They could see almost the entire history of the development of the field of CS, from some of its earliest beginnings in the 1960’s to the time of the interviews. It is their level of historical comprehension that allows this use of terms to take place, for newer, younger members of the academy would be unlikely to refer to the terms in the left column of table 6.2 and, if they did, it would be without the experiential knowledge of the meaning of each term as a knowing actor (Askling et al, 2001).

Drs Scarlett and White talked about a young CS academic (let’s call him Dr Pink) being fooled by a demonstration of a new technology because the concepts involved were no longer or not recently taught in CS degrees,

Dr Scarlett: Dr Pink went to a demonstration ... he saw live video on a 3G phone. It hasn’t come yet. Why not? Well, there was a mast in the corner of the room and he had the only phone there with the network to make it work. [he laughs]

Dr White: But we all know its to do with [whether] its broadcast or narrowcast, and its bit-rates and its error-checking. We know all these things. Its nothing new.
6.1.4 Theory versus Application

There is a clear divide evident in the interviews between the theoretical and applied nature of CS. This was not presented simply as a divide between academia and industry, it also existed in another form within academia between the old and new universities. There was another division evident between what these CS academics perceived as being knowledge and that knowledge acquired and owned by end-users in the real world. This is an academic field that was seen to divide into a fragmentory ‘us’ (new university CS academics) and ‘ thems’ (old university CS academics, CS practitioners, and CS end-users) deeply felt by these CS academics, evidenced in such strong language as: ‘ignorance’, ‘rubbish’, ‘impenetrable mess’, ‘hand-waving’, ‘crappo’, ‘ugly’, ‘awful’, ‘detest’, ‘miles from reality’ and others reminiscent of the emotions recorded by Winter et al (2000).

This is worrying as it goes against the theory that universities work together with other partners in order to maximize their national economic benefit, as proposed by such authors as Leydesdorff (2006) and Gunasekera (2006). This division between types and uses of CS knowledge, whether theoretical versus practical (Gibbons et al, 1994) or explicit versus tacit (Polanyi, 1958) or production versus utilization (Short, 1973), is a theme that echoes throughout the interviews.

It has already been mentioned above how Dr Blue found that applying his old-university theoretical CS knowledge on how-to-program to a real-world programming site produced a dichotomy of meaning over the term, ‘badly written’. Dr White found a similar problem when he was teaching theoretically good CS practice to professional software engineers,

I picked up Dr Mauve’s module one year when he was on sabbatical, on quality. And, standing chatting, we were talking about the horrors of using global variables, whatsoever, and they [the students] were all really restless and really upset, and I remember stopping dead and saying, well, what is it? And they said, that’s just rubbish. That’s
rubbish. If you want the fastest, most efficient program, just use global variables and repeat the code as often as you want. They worked in a different world. There was a complete paradigm clash, if you like, between what I was saying to them and what they were saying to me.

Dr Grey explained the dilemma of teaching both CS teaching and CS practice,

I still think back and think about how you did things and how you used to produce projects and realise how the work environment is different to this environment. Everything here works in a one-year cycle and to some extent you’re not totally committed to control everything in one. You’re not directly responsible for people coming along and saying: things don’t work. When you have to maintain things and there are commercial pressures to do. A story I still tell students is that I learned COBOL on [a] Monday morning when I was the only programmer at work and the project planning run didn’t run over the weekend - batch things - and the printout came out at one line per page and a ream of paper that says ‘error in program’ ... And the [computer] operator doesn’t think. What will we do now? And you’ve got a load of guys standing round doing nothing.

The issue of the different nature of CS in the older and newer universities was often visible in the conversations, but lacked a simple we-are-good/they-are-bad stance. It was evident that the academics who had read CS as part of their undergraduate studies - who did so in old universities - expressed a strong affinity to the core CS theory they had learnt. This dichotomous stance contrasted the importance of the fundamentals of CS as learned in their old university studies and compared with the applied nature of CS in their past work experience and current new university teaching. It was almost as if Stiles’ (2004) multi-faceted view of inter-field relationships existed as a matter of intra-field fact within the workplace and the daily grind of these CS academics. These tensions were illustrated by the following two extracts from Dr Blue,

I’m not interested in Computer Science except in so far as I know it’s there and I can trust it. What I’m interested in is applying it.

In one respect we move backwards because we were lucky, as you said, we had Algol-W. We were in academic institutions in the 70’s ahead of
our time. So we learned Algol-W then you went backwards. You went
backwards in the real world into Assembler, perhaps then forwards into
COBOL, FORTRAN and then into C. And that was still way below Algol-W
in terms of, you know, programming language theory. And then you
moved into what I would call the real good sound theoretical languages
which were Pascal and Java. Now, in the interim, we now had C++
appearing. Which I absolutely detest. As I say to students, C++ is a
bastard language.

The two extracts are peppered with contradictions of pure CS being good ('we were
lucky, we had Algol-W'), bad ('I'm not interested in Computer Science') and applied
CS being good ('I'm only interested in applying it (CS)') and bad ('C++ ... I
absolutely detest'). Dr Blue is seen to be struggling with the very nature of the
field he had learned, applied and taught in.
6.1.5 The Rate of Change in CS

Each of the academics talked of the difficulties in keeping up with changes in CS. For example,

Just trying to keep up with, well not all the changes, trying to keep up with the changes relevant to what I’m doing at the moment … Och - impossible! It’s impossible. (Dr Black)

I’m well aware that things have steamed past me and I just haven’t had the time to get into them enough to really understand them. Which is worrying. (Dr Green)

There is evidence in the academics responses of pragmatic rather than considered, reflexive decision-making to handle this rate of change. For example, .NET ("dot net") is a major applied CS-wide strategy from the world’s leading computer company, Microsoft. This is an example of the constant innovation, and it is fatiguing,

I know nothing about dot net. I’d never heard of it. Because I don’t need to. I mean, there’s something out there called dot net. Like I care. (Dr White)

In addition to these major developments in the wider CS field there is also evidence of major subfield changes continuing to occur in CS. The academics talked of developments in such novel areas as video games production, digital arts, smart systems, ethical hacking, telecommunications and Internet website creation. Krishnaprasad (2000) described this growth and change as ‘challenging’ and this reflects universities’ admissions that CS is, as a knowledge field, rapidly growing (Becher et al, 2001). None of these new subfields were in existence as a major teaching area ten years ago, but all of them now employed significant numbers of academics in The Department teaching on named degrees. Further, these subfield
novelties were the areas of specialisation and growth in The Department, not core CS (see also figure 3.1).

The academics noted that the continuing development of subfields had made it difficult to partition the subject area into meaningful and manageable chunks for administrative purposes, especially as these new areas appeared rapidly, and often with each change of Head of School (which event had taken place at the site in recent times every few years.) The academics found it hard to locate their areas of teaching to their location in the University structure, for example in the following conversation,

Dr Black: No, I meant, which Division [are you in]?

Dr White: my teaching is in Dr Fawn’s Division, but I’m in Dr Purple’s Division

B: That’s sensible [laughs]

W: I’m completely confused. I just teach the modules.

B: I’m much the same

W. Which Division are you in then?

B: I’m in Dr Fawn’s

W: But your modules are?

B: One from Dr Fawn’s and two from Dr Purple’s [laughs]

The administrative structures of the university were seen as, quite literally, laughable. There was another dichotomy between these administrative structures and the rapidly changing subfields being taught in the department. The need to keep bums-on-seats, especially after recent job losses, drove the selection of teaching subfields delivered in the Department. This general problem is apparent elsewhere in academia in other fields. Bolton (2001) called for greater autonomy in academic structuring, which might also be useful for CS.
6.1.6 CS in Relation to Other Academic Fields

There appeared to be two principal underlying fields that these academics considered to be fundamental to CS: Mathematics and Engineering. This may well relate to the academic backgrounds of these teachers which were (as expressed in the interviews): Mathematics, CS, Computational Astronomy, Electronic Engineering and Computational Chemistry which fields are close to the roots of CS as a discipline (Smith, 2004; Copeland, 2004).

Mathematics was seen as being of primary theoretical importance to CS in general and also to some of the newly-emerging subfields being taught. For example, one of the fastest growing new subfields in CS was video games programming,

The games course, its just Software Engineering and Maths. (Dr White)

The newly emergent areas of teaching also showed evidence of a similar Mathematical content or the need to consider such content,

Dr White: What are the entry qualifications for Smart Systems?
Dr Grey: AAA, AAB. [Scottish Highers]
W: You don’t need Maths or whatever?
G: Systems thinking would help.

Further, the lack of Mathematical knowledge in current teaching was seen to be damaging to students’ understanding and problem-solving skills as future CS professionals,

I think there are a lot of dabblers who are pretty ignorant. I’m not using this in a derogatory sense, but, they are ignorant of the area and flounder around and get something working despite their ignorance. I was looking at a games [student work] ... its quite woeful the ignorance. I’m not blaming them for it, but them to get, it was, what steps should I be having for these physical simulations? I mean, to ask that question
needs a course on numerical analysis, essentially. They're just asking
this question naively and don't realise what's behind it. That's just an
isolated example. There's lots and lots of such as those. (Dr Black)

Engineering, a field that has declined in the Scottish new universities, was on the
other hand seen to be increasingly important to the applied side of CS as CS
continued to change. Consider the following extract,

Dr White: So, the department is drifting in that kinda direction of usable
technologies and things?

Dr Grey: Yeh. I can't sell BSc Computing because its a boring subject.
But everyone in school has a computer, why would you want to do it at
university? There's no BSc Telephones!

W: BSc Cars? You're right, yeh, yeh.

G: You still need people to create systems. Successful departments are
those which have a speciality. You know that if you're going to this
university you're going to do real computing degrees like B.Sc. Games
Technology. So hacking is another attempt to do the same thing: this is
where our focus is in: ethical hacking. Professor Orange's focus, what
he's interested in, is trying to get it all to hang together: smart systems.
We were talking to someone yesterday, communication, in essence,
what you're trying to do is put together a communications system where
you have your data at the right place, and you can use your WiFi device
and whether that's your PSP or your phone, or a laptop to gather
information, and you can put a game on top of that which is not just a
sitting-in-a-room game but a game in an environment. Obviously a
laptop is not very useful in that case, but a phone or a PSP would work.

This conversation extract, on The Department's current and future take on CS,
showed several instances of Engineering concepts. Firstly, there is the Electronic
Engineering conversation around various pieces of computer hardware (laptop,
desktop, PSP and phone) which is in marked contrast to the other general talk in
the conversations focused on Software Engineering programming. There was also
talk on the usability of the Human-Computer Interfaces of the hardware/software
combination \textit{in situ} (the locational preference for a PSP or phone over a laptop). Finally, there was talk about Telecommunications Engineering (covering such issues as WiFi, hacking and the need for an agreed approach to telecommunications systems). These brought together ideas that combined Computing, Mathematics (including Statistics) and Engineering (Lidskog \textit{et al}, 2005).

On the evidence of the conversations of these CS academics the field, despite its constant changes, had been and continued to be and perhaps would always be a naturally cross-disciplinary field fundamentally based upon Mathematics and Engineering.

In contrast, emerging subfields that lacked the rigours of Mathematics or Engineering were disparaged by these academics. For example, Internet web page design and creation was considered to be on the very edges of academic respectability, despite this being a popular degree offering in The Department and elsewhere in academic CS,

Dr White: Web design and development.

Dr Green: Is that a big enough chunk of a field ... to be self-standing? ... I don't know the contents of that, but I would say just about. If it was done as I might envision it was, it could be done. Yes, I would probably say there is just enough. Then, I'm putting in lots of programming there.

This is even more strongly expressed by Dr Grey,

Right, right, my prejudice here! There is not a subject called web design and development and we should not running a degree course in web design and development.
CS, as expressed by these academics in conversation, is a wide, changing and, quite possibly, growing field with a large vocabulary of technical terms that are necessary to know in order to communicate meaningfully within the field’s complex and varied knowledge base and with others working in CS. However, this vocabulary is volatile and quite probably incomprehensible in large areas to a lay audience - the users - of CS technologies. This is further complicated by the abstract nature of much of the knowledge in CS. These CS academics were adept at finding parallels in life to help explain abstract concepts to each other and to their students.

The knowledge-base of CS continues to change. Some terms become obsolete through world-wide extinction, others go into abeyance, to perhaps return for re-use later. From the evidence of these conversations it is likely that at any one time perhaps only half of possible CS terminology is in current use. It is a feature of these older CS academics that they could draw upon decades of experience of terminology-cum-knowledge to populate their understanding of the field.

There are significant and, on the evidence analysed, sometimes heated divisions over CS knowledge ownership. These are expressed in such a way as to divide the field’s ownership into four different areas centered upon (i) the older universities (theoretically orientated), (ii) newer universities (application orientated), (iii) industry (solution orientated) and end/lay users (pragmatic orientation). The CS academics showed evidence of being embedded in taking up their position and being willing to defend what was their shared understanding of the ‘true’ nature of CS.

The evidence shows a rate of change of knowledge and its scope in CS which had gone beyond the ability of these CS academics to keep up with. Attempts to make these changes concrete in managerial structures had created confusion and brought
these structures into open ridicule. Government funding structures had also affected the taught field, creating instability and producing emergent subfields of dubious academic worth which continued to extend the CS teachers knowledge-base into many new areas that would not have been seen as part of CS expertise a decade ago.

And yet, what stands out is that these CS academics were remarkably agreed that the field remains one with clear theoretical and applied foundations that would not have seemed odd decades ago. CS, as theory, was still seen as being an extension of Mathematics and, in application, as being an Engineering discipline. All other teaching, even when held under the administrative umbrella of CS, was seen as being potentially invalid.
6.2 New Knowledge Production in Computing Science

This section is a consideration of two major aspects of knowledge creation in CS. Firstly, where the academics indicate that the issue is one of the changing nature of CS knowledge itself. Secondly where they believe these changes originate from. The second section is structured to allow consideration of whether and where the new knowledge is derived from: within the CS academy, from CS industry, by the CS students or entirely outside from the field of CS applications, what is normally referred to from within the CS community as the ‘users’.
6.2.1 Knowledge Change

Each of the academics interviewed had significant concerns about the rate and amount of change of the knowledge base in CS, reflecting what Becher et al (2001) had found more generally in academic subjects. This change was often seen as being not just being natural, but indeed healthy both for the field and for those working as academics in the field. However, handling both the amount of change, and constant rate of change was wearing and a challenge to the professionalism of the CS academic.

Dr White: Subjects change. They change over time. But we’ve had this concatenated timeframe where, in a period of 30-40 years an unbelievable amount has changed. Do you reckon this rate of change can continue?

Dr Black: There’s no indications it’s ever going to slow down.

Each of the academics concerned came to the blackboard with years of experience of this changing field already behind them. There is something slightly depressing about the above question and answer clip. There is a feeling that this paradox of constant change has worn down the academics themselves, made even more evident when the audio recording was listened to. Elsewhere in the interview for example, Dr Scarlet, talked of just hanging on in there until retirement was possible.

However, the academics were also upbeat about their greater understandings of the meta-changes and the forces behind them, even when applied to novel technologies that were, on the face of them, entirely new products and experiences.

Dr White: What do you think are the biggest challenges we’re facing today as we’re teaching?
Dr Grey: It's the management of change ... with mobile phones, for instance, it's such a rapidly changing market. Phones are obsolete in nine months - seven months.

In less than the time it takes to teach one year of a full-time undergraduate degree programme a technology may well completely change. At the meta-level Dr Grey pointed out that the CS academy needed to recognise this and to have a coping strategy to handle this, as Ahier (2000) indicated. This strategy must allow the academy to see these changes, to have a structure to analyse the changes, to compare what is known and taught currently with this new knowledge, and to plan to implement any required changes to the taught curriculum. What he proposed is no mean feat when performed mid-year and teaching at the cutting edge of a huge (in terms of market size, technology change, national differences and product range) subfield as mobile telephony or video games (see figure 6.2).

```
subfield   --------> current subfield knowledge
           |                     |
           v                     \ teaching curriculum <------------------------/
           |                     |
          new subfield       |                     |
           --------> analysis of new developments
           |                     |
           v                     |
           |                     |
           comparison to existing subfield knowledge
           |                     |
           v                     |
           |                     |
           reappraisal of curriculum
           |                     |
           v                     |
           implement curriculum changes---------------------/
```

Figure 6.2 - The necessary mid-academic year reassessment of the CS subfield curriculum due to rapid subfield developments.
Questions must exist about the quality of this mapping from real-world knowledge to the classroom when it is sometimes to be done so rapidly,

You just do the best you can and hope to lash things up and change ship. Professor Taupe [a previous Head of School] used to say that everything has a five-year lifecycle. As soon as it gets going, the next one is coming along. (Dr Scarlet)

Holding together a knowledge field as wide and volatile as CS - video games, Telecommunications, Knowledge Systems, Information Systems, etc. - is more done by dint of the activity of doing it, than by applying a set of meta-understandings to the meaning of the nature of the field. Dr Scarlet's contribution (above) usefully provided a metaphor of imagery of a sailing ship battling a storm, using practical rather than purely theoretical curriculum review structures. Perhaps the CS department that succeeds lives to hope for quieter days and even a personal port of ease; the one that does not succeed, sinks.

Another difficulty is attempting to understand, or accepting not to understand, the meta-reasons for changes in the field,

Dr Black: For various reasons we decided to change [the programming language we teach]. So, for me, keeping up with these kinds of things is very time consuming ... the area is huge. The expertise is huge and all over the place.

Dr Grey: I think Java's dying a bit now.

As will be seen below (§6.3.1), nothing is more important to CS academics than programming. Yet, this core subject is subject to constant change. The reasons for making these changes are not simple, nor are they driven by the academic himself. To those outside the CS field's decision-making group the reasons for such changes can appear opaque, and even, as Dr Black stated above, the changes can be difficult to understand from within the field too. Indeed, any reasoning is inexact where, as Dr Black again indicates, the defining variables in change include the requirement for time (which is in short supply), understanding a field so large, having limited expertise in applying such a large field (as the senior lecturer, Dr
Scarlet honestly put it, 'What do I do? I just teach Java now'), and the disparate nature of a field that stretches across so many horizons.

But, if there is one area of recent difficulty that regularly raised its head with these CS academics when discussing their attempts to get an understanding of changes in the field, it is that of Object Orientation. The change from procedural (actions and verbs) systems thinking to object orientated (nouns and things) approaches is a change that originally came from an independent research laboratory, but has slowly grown to become the primary programming paradigm. Those who promote this approach have a triumphalist, even evangelical zeal for this complex concept,

Object-oriented programming came into existence because human consciousness, understanding and logic are highly object-oriented. By way of "objectifying" software modules, it is intended to promote greater flexibility and maintainability in programming, and is widely popular in large-scale software engineering. By virtue of its strong emphasis on modularity, object oriented code is intended to be simpler to develop and easier to understand later on, lending itself to more direct analysis, coding, and understanding of complex situations and procedures than less modular programming methods. (wikipedia.org)

These quotes also illustrate one of the major problems in CS acadaemia and its knowledge field: such ideas enter the field and become embedded despite (as is discussed below in the section §6.3) the sometimes extreme difficulties in understanding these concepts by practitioners, academics and students. The first quoted paragraph flags up issues of nomenclature in the field. Just what are 'inheritance, modularity, polymorphism and encapsulation' and why must they be understood in order to become a programmer? The second paragraph is less than honest as it implies a change from instructional to object orientated approaches, whereas, in reality object orientated programming also involves knowing and using the instructional approach. There is a pseudo-philosophical paragraph on the nature of human sentience, quite probably not open to challenge by a CS academic. Words such as 'greater flexibility', 'popular', 'simpler' and 'more direct' promote this 'great' new idea; yet there is a slight dissonance in the passive text: 'it is intended to promote'. There is an disturbing echo of the Wilsdon et al (2004) Environmental Sciences report discussed in §2.5.4.
Contrast these positive world-view quotes with what was said by the CS academics themselves, for example,

It really blew my mind ... it’s a head thing, not a technical thing. (Dr Blue)

I struggle ... [it’s] incredibly complex and ugly. (Dr White)

Can you think of a brick wall you came up against? (Dr White)
Object orientation. (Dr Scarlett)

Yeh. I mean, if that’s the way it is, then that’s the way you have to do it. So, I’m not sure they [students] understand the ins and outs of it. (Dr Black)

With something like Java [the leading object orientated programming language], the whole point is that you need to take, to use a class and you know what the classes are and the methods are so that you can tackle each problem. So, here’s this particular problem and I want, which class do I go to, and there’s this vast array of classes. Which one do I go to? Oh bugger I’ll just do it myself. And so, you’ve got this extra load that you’ve got to get inside your brain. (Dr Green)

These quotes from the CS academics show that not just the knowledge within the field, but the emergence and acceptance of new knowledge, is contestable. The academics are acting within, and often responding to, changes in knowledge which are created out-with their control and even out-with their understandings of where changes fit within the current field.
6.2.2 The Academy

There is remarkably little evidence in these conversations that any part of CS academia has any relevance to new knowledge generation in the field. This may be due to the academics taking up a stance, considering CS as a pure, theoretical field such as when Dr Blue (who was the only interviewee with a first degree in CS) noted that,

Dr Blue: I don’t see there have been many advances in Computer Science in the last 30 years.

Which was in stark contrast to what he had said about change and the difficulties of dealing with it. At least to Dr Blue, these changes were nothing to do with core CS.

This group of CS academics acted quite differently to academics in many other more traditional (and newer) fields in where they located their knowledge. This knowledge was not on their bookshelves. When asked, none of these academics could point out more than a tiny number of books, never more than five or six, that might be of long-term use to them, and sometimes these texts were merely of what might be considered sentimental value, such as Daniel D. McCracken’s 1961 tome, ‘A Guide to FORTRAN Programming’, or obscure CS foundation theory texts, such as, John von Neumann’s 1963 series of ‘Collected Works’. None of these books would be of any relevance to teaching or curriculum development in applied CS. Yet perhaps their possession allowed these academics to hold on to a feeling of belonging to an academy.

Some of those interviewed, such as Drs Grey and Brown, could not name any medium-to-long term useful CS textbooks that they possessed. The conversation with Dr Grey took place as he surveyed his bookshelves, about 3m of books in all:

Dr White: Is there anything there you reckon is the wisdom of the years and will last another ten years?”

174
Dr Grey: Yeh. [long silence] There's a Software Engineering [book].

Dr W: Have you got Somerville?

Dr G: Yeh

Dr W: You'll always have Somerville?

Dr G: Yeh

Dr W: Ince?

Dr G: Yeh. And what looks like a nice book on the shelf on Java mobile programming. Out of date now.

Dr W: 'Developing Games' is [also] out of date.

Dr G: That one there on developing J2ME games is alright but its two years old. It'll be out of date in a year's time, I guess.

What does not show in this extract are the long pauses in the conversations as the two academics hopefully surveyed the bookshelves. They were looking for an academic textbook in CS which might be of long-term use, but could only find two as examples of the need for a general text on Software Engineering. In addition to the sentimental value texts, the other books were all short shelf life 'how to' books relating to current, or past, versions of commercial CS software products.

A usual way of generating new field knowledge in academia is to do research or scholarship. None of the CS academics interviewed were active in research. This may seem extraordinary in a field where new knowledge is so quickly created, but it also indicates a lack of time and resources as all the academics had been active in research or scholarship at one time,

I loved research ... its fairly difficult to maintain any momentum in research. (Dr Black)

Dr White: Scholarship or research - do you get any time to do them?

Dr Green: No. No.

In this particular CS Department in the CS academy there was no indication in the interview answers given of academic scholarship from anywhere producing any useful forms of new knowledge into the field. This was noticeable to these
academics and could leave them feeling frustrated and inadequate as their knowledge was reactionary rather than generative, reflective or reflexive,

For some reason, [on] that machine there, I couldn't open PDF files without it hanging. Until I got a new computer. And the new computer doesn't hang. Why was it hanging on PDFs? I have absolutely no idea. And you feel like a really stupid end-user. (Dr White)
6.2.3 Commerce

The major source of new knowledge into CS academia was from commercial CS companies. The plethora of terminology and developments was an almost entirely commercially driven force, with little evidence of input from the CS academy. If programming (see §6.3.1) was seen as being the core skill of CS, then it is a commercially-driven field only. The academics talked of redundant and new programming languages, but just how many of these languages they mentioned in the interviews were ever academically generated or more generally used? (see figure 6.3 and table 6.4).

![chart]

figure 6.3: the distribution of sources of high-level computer languages

Table 6.4 shows just four of the discussed languages as being academic-sourced (Algol, BASIC, LISP, ML), three were never used outside academic teaching and research (Algol, Smalltalk, ML), the first of these (Algol) is long redundant, while of the other three only BASIC still exists today and then only in a form of complete
commercial rewrites outwith the academy such as the DarkBASIC games
programming language (darkbasic.thegamecreators.com) and Microsoft's Visual
BASIC (msdn.microsoft.com/en-us/vbasic) event-driven visual programming
language.

<table>
<thead>
<tr>
<th>language</th>
<th>Year</th>
<th>source</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algol</td>
<td>1958</td>
<td>ETH Zurich</td>
<td>academic</td>
</tr>
<tr>
<td>FORTRAN</td>
<td>1954</td>
<td>IBM</td>
<td>commercial</td>
</tr>
<tr>
<td>COBOL</td>
<td>1959</td>
<td>US Dept of Defense</td>
<td>government</td>
</tr>
<tr>
<td>LISP</td>
<td>1962</td>
<td>John McCarthy</td>
<td>academic</td>
</tr>
<tr>
<td>BASIC</td>
<td>1963</td>
<td>Dartmouth College</td>
<td>academic</td>
</tr>
<tr>
<td>Simula</td>
<td>1963</td>
<td>UNIVAC</td>
<td>commercial</td>
</tr>
<tr>
<td>PL/1</td>
<td>Early 1960's</td>
<td>IBM</td>
<td>commercial</td>
</tr>
<tr>
<td>RPG</td>
<td>1960's</td>
<td>IBM</td>
<td>commercial</td>
</tr>
<tr>
<td>Pascal</td>
<td>1970</td>
<td>CDC</td>
<td>commercial</td>
</tr>
<tr>
<td>C</td>
<td>1972</td>
<td>Bell</td>
<td>commercial</td>
</tr>
<tr>
<td>ML</td>
<td>1973</td>
<td>John Milner</td>
<td>academic</td>
</tr>
<tr>
<td>Smalltalk</td>
<td>1980</td>
<td>Rank Xerox</td>
<td>commercial</td>
</tr>
<tr>
<td>SQL</td>
<td>1981</td>
<td>IBM</td>
<td>commercial</td>
</tr>
<tr>
<td>C++</td>
<td>1983</td>
<td>Bell</td>
<td>commercial</td>
</tr>
<tr>
<td>PERL</td>
<td>1987</td>
<td>Unisys</td>
<td>commercial</td>
</tr>
<tr>
<td>Visual BA-</td>
<td>1991</td>
<td>Alan Cooper</td>
<td>commercial</td>
</tr>
<tr>
<td>Java</td>
<td>1995</td>
<td>Sun Microsystems</td>
<td>commercial</td>
</tr>
<tr>
<td>C Sharp</td>
<td>1990</td>
<td>Microsoft, Hewlett Packard, Intel</td>
<td>commercial</td>
</tr>
</tbody>
</table>

table 6.4: generic programming languages mentioned by the CS
academics, with origination and date (wikipedia.org).
The programming tools and languages that came out of commercial CS were not always welcomed by these members of the CS academy who took part in this study. The video games industry primarily coded its programs in the non-academically-designed object-based language C++. This was thought of as a bad thing,

I think it [a video game being programmed in C++] is purely through historical and through ignorance by games producers. (Dr Blue)

and the need to teach programming skills in academically inadequate languages caused no small degree of frustration in these CS academics who said that they felt they had to teach students techniques and approaches which were poor, falling well short of the academic training and background the academics would like to be able to give their students, in an ideal world.

There was also a difficulty in accessing explicit knowledge on these languages. In the past languages came with definitive texts on the syntax and application of the language, called manuals: 'I miss manuals' (Dr White). Programming teaching and learning were then supported by short-life textbooks aimed at professional developers, changed regularly - sometimes annually - as the versions of the language change, and by programmers' experiences posted in on-line developer discussion websites. The academic had become one person in the learning field, with students, developers and lay-users occupying other places, each with their own knowledge and understandings.

The knowledge of one person in CS - whether academic, student or practitioner - as was noted above (§6.2.1) in the discussions on object orientation and below in those on the use of a wide terminology in CS (see §6.1.1, §6.1.3), could be incomprehensible to the outsider, even if the outsider was another CS academic working in a different subfield. Take, for example, this explanation on programming by Dr Blue on Microsoft Corporation's approach to systems,

The machine that you need to know about isn't even a real machine. For example, if you're programming against the .NET environment what you're needing to look against is the CLR, the .NET intermediate language. The .NET assembly language doesn't exist, because its run against a virtual machine.

179
Fifteen years before the concept of a ‘virtual machine’ was not the same thing as was being discussed by Dr Blue here; the label remained but the underlying concept had slipped sideways. In a similar way, five years before there was no meaning for the commercial label ‘.NET’ nor was there any concept or term for an ‘intermediate language.’ Despite my 30 years in the field, I was unaware during the conversation of what ‘the CLR’ was or how it related conceptually to the ‘.NET intermediate language’. These terminologies, their meanings and the rate and source of change again raised issues of academic feelings of competence and confidence, which also had implications for student teaching and learning.

It was often this - the changing nature of CS knowledge, its speed and rate of alteration - that drove the academics forward. None of the CS academics concerned would have wished to leave academia. Indeed, some expressed a desire to carry on in the field post-retirement,

To keep abreast of changes in the computing industry is something more interesting because you know that there are interesting challenges in them. (Dr Brown)

There was a notable lift in all the interviews when those interviewed expressed their enjoyment in tackling the issues of how to understand and explain the way applied CS is today, and finding new ways to make sense of these changes. In this milieu of externally-driven change, teaching CS remained overall something these academics found to be highly enjoyable.

But, how did the academics then keep up with changes well enough to feel confident in their knowledge when so much of it is generated by non-academics? Apart from one passing comment, none of the academics mentioned a professional body, such as the Association of Computer Machinery (acm.org) or Institute of Electronic and Electrical Engineers (ieee.org) both of which have a series of academic CS journals or lighter publications dating back to the 1950’s, nor was anything mentioned of the British Computer Society (bcs.org.uk); these are the industry-academia lead bodies in the UK and the USA. Nor did any interviewee mention academic conferences even though these would sometimes be held within easy
travelling distance of the study site. So, how did these CS academics stay on
top of their field, as they clearly felt they did overall,

Well, you always have to do something to keep abreast, and I tend
to read Dr Dobbs. I’ve had a subscription to Dr Dobbs since 1985.
And that’s a good way to keep abreast of things. In a way that
allows you to pick and choose quite easily whether you want to get
into some subject closer. (Dr Green)

I read MCV which is telling me what goes on in the games industry ...
MCV, this is what is really going on. (Dr White)

They read commercial programming and software development journals and access
similar material on the Internet, via such sites as gamasutra.com or ddj.com. In
effect these applied CS academics sourced knowledge of the field as it then stood,
and as it changed, primarily from short-life (i.e. of a few weeks, months or years
usefulness rather than decades) commercial chat sources. This is not to say that
these were lightweight sources as they were the same sources used by serious
commercial developers.
6.2.4 Students

There was a disappointment expressed by the academics with the students who were then studying applied CS at the site. This was seen as being due to many possible reasons that included: the sudden growth of the university population (Ennals, 2004), the difficult position of a 1990’s university being truly perceived as being a university, the lower numbers of students wishing to study CS (see figures 3.2, 3.3 and 3.4) and the consequent dropping of standards to let them in,

weak students (Dr Brown)

The quality of student who went to university [once was] much higher than the quality we take now (Dr Black)

This was a paradox. The academics perceived that, before the institution gained university status, it was an institution that attracted good students but now that was entitled a university the quality of students had lowered. The numbers had certainly risen, despite the recent drop in applications to study CS,

Their [the students] knowledge can be quite limited. They can do certain things, but start probing it and you find they don’t know as much as you first thought. (Dr Black)

This may also have been evidence for the students having how-to knowledge of applying CS but lacking the underpinning theoretical understandings of the CS academics. The perceived drop in student quality cannot simply be put down either to there being a lower quality intake of students in terms on school-leaving qualifications as, when the institution gained university status most of the students were reading for HND awards and now all students read only for BSc (Honours) awards (although some would exit with a DipCE, DipHE or BSc ordinary award.) However, there remains a possibility of seeing the past through rose-tinted spectacles as the days of the ‘college’ had ended 15 years ago, a long time in the life of any academic.
A major change had been in what the students come in knowing about Computer programming,

Dr White: To what extent do they have programming knowledge?

Dr Green: Its poor ... they come in with no computing, no programming, whatever.

This is critical when it is considered how programming was seen as being core to the subject by these CS academics, indeed Dr Green may even have been equating CS and programming in this instance. The interviewees also talked of how students used to come in with programming knowledge when their home games machines were computers and were easily programmed in a dialect of BASIC on such machines as the BBC, Atari and Amiga computers. It was widely accepted by the group of academics that programming had become both tougher and students came in with zero knowledge of the subfield. This was a recipe for difficulties and may also have been an explanation for the perceived weakness of the students.

As the academics found it challenging to handle the changing nature of CS knowledge, they also saw this as being a problem for the students too. The students gained new CS knowledge, but were perceived as having conceptual difficulties in bringing this together into a way that made sense,

They have this applied knowledge of using computing devices. (Dr Black)

Dr White: Do you think they [students] handle the subject as well now as when it [CS] was a more discrete subject?"

Dr B: No.

Students did not come into university as tabulae rasaee, as perhaps was more the case in the less-technological world in the past, but as people who had long interfaced with this complex and increasingly pervasive industry. However, the expanding, changing and evolving nature of the breadths, depths and heights of CS knowledge may have made it more difficult for students to create meaningful knowledge structures, at the same moment as the CS academics struggled to teach what they knew and understood into what their students already knew and understood.
Mike Holcombe (2001) pointed out that teaching quality can be poor in CS departments. These CS academics in the study were all dedicated and hard-working teachers, but they also admitted to having a poor grasp of some parts of the core subject. It is not difficult to see that the two areas of knowledge (that held by the academic and that held by the student) and teaching (by academics and engaged in by students) might be related to this problem.
There was an acknowledgment by the academics that the fast-changing nature of CS was a wider problem than just for academics, students and practitioners. The academics talked of how their friends and relatives struggled to stay in touch with changing CS, and of how they, the academics, had to occasionally claim ignorance,

I don't really know how ordinary people understand [how to connect to the Internet and use email] (Dr Black)

This could also be read as an acknowledgment that knowledge change and an associated lack of understanding may be two sides of the CS coin. You have one because you have the other. Change causes lack of understanding, rapid change causes considerably more lack of understanding, and a lack of understanding causes a search for new knowledge at a pace related to the felt importance of the knowledge that is lacking. A partial and incomplete understanding may be a useful descriptor of what CS is, meta-epistemologically.

However, the user community that these academics interfaced to was far from ignorant, and indeed was seen by them as a possibly rich source of understandings of innovations in CS. This is perhaps not too dissimilar from such other academic fields as Management Science and Environmental Science (see §2.5.3, §2.5.4). Wensley (2002) was concerned that Management Science academics and practitioners were both entrenched and deeply divided. On the evidence presented by the CS academics in this study, this was not entirely the case in CS: divided, yes, but these CS academics were trying hard to stay in touch with other holders of CS knowledge.

Dr Black talked both of how an end-user gains knowledge and how he, the academic, kept a track of such end-users with their useful knowledge,
[I get] stuff off the Internet that other people put together, like games engines. There is a community of such developers out there. (Dr Black)

The Internet makes it easier to do in some sense. It also makes it harder as I maintain links to web pages ... it takes quite a lot of time to maintain that. (Dr Black)

This end-user community is itself computer mediated, and so is subject to the same inherent instability and change that it’s parent field, CS, is. There is a commonality of applied understanding by academics and lay users as all potentially could be using the same toolsets. Jamison (2003) discussed how new knowledge is sometimes located outside academia and these CS academics take a pick’n’mix approach to find knowledge that is fit for their purposes.

There is a source of help and assistance for CS academics - either through direct interface or via surfing or even lurking - in cyberspace itself (in such media as web sites and community chat sites) and through into the real world (via these sites). This community is not academic and is creating knowledge of variable quality and uncertain provenance. Indeed, much of what is posted or made available was recognised by the academics as being wrong, whether it was commercially sourced (as on the first quote, below) or from a more general end-user source (as in the second quote, below),

We were told that WAP was the Internet on your phone. It’s never going to work. The screen is black and white. I’m never going to get colour on the damned thing. (Dr Grey)

There are a lot of dabblers out there who are pretty ignorant ... and get something working despite their ignorance ... they are asking questions naively and don’t realise what’s behind it. (Dr Blue)

Again, this can be seen as a feature of the nature of applied CS knowledge: it is a mixed bag of statements of various qualities whose truthfulnesses are not easy to grasp. In the first quote above quote from Dr Grey, he was referring to the information from a major Telecommunications company which ran a multi-million pound advertising campaign in the 1990’s promising something that was known to him as being impossible with the then available technologies.
Creating understandings of how people in the lay community understand and make decisions based on their existing CS knowledge and CS needs can be nigh impossible. The CS academic could find himself stretching out to find new ways to create understandings of the decisions of others, even when the others were CS professionals,

Dr White: Why has something as complex as Java caught on?

Dr Green: Its a hard question to answer. Whether society has in general moved into a mode whereby people don’t bother to evaluate things. They just say, here’s a new thing, it must be better.
6.2.6 Conclusions

CS is a field that is ever and always in a state of turbulent change. Its knowledge is to be found in many places: the academy, commerce, students and lay people. This knowledge is of variable quality and can sometimes be difficult or impossible to grasp. The churn in CS knowledge, where it is created and where it is found, is a challenge to a teaching CS academic, especially when the change can occur during a teaching semester, and almost certainly will occur during a teaching programme's many-year duration. The required changes to the curriculum need to be controlled, involving an understanding of the current CS subfield situation, knowing where to find details of change and of changes, and knowing when and how to apply these changes to the curriculum and to students in classes.

Full knowledge of what applied CS is is unobtainable, and increasingly so. The corpus of knowledge that exists is too wide to grasp, hard to locate, lacks any established structure, is in constant flux, ever growing, and is of variable quality and provenance. Indeed, provability depends (in a way that is perhaps strange for such a mathematico-logical subject) to an extent on the position and opinion of individual people involved in knowledge creation, storage and location, and in the validation of the knowledge by third parties.

There was widespread agreement among these academics that the core subject of CS remains that of programming. This area was itself affected by the highly complex and mentally difficult concepts involved in the current favourite approach: object oriented programming and the flux in language development.

There was no evidence in the interviews of much, if any, new CS knowledge available from within the CS academy itself based in traditional university laboratories. This is a startling thing. Books were seen to be of little value, and of short-term use. The books that were used were commercial and student texts, and rarely, if ever, ones that were involved with meta-knowledge in CS. Instead, the
primary source of CS knowledge was from the commercial and on-line user communities. Non-academic practitioner magazines were also seen as important sources of knowledge, and websites, email lists and chat sites were used as knowledge exchange sites.

As this knowledge is not academically mediated, so it lacks a feeling of being a structured epistemology. Indeed, much of it is probably simply not true, even when produced by reputable sources such as major companies. However, it is this lack of structure, the speed of change, the unpredictability of the future and the variability of subject content that appears to be exciting for the CS academics.

These academics’ opinion of students was that they were less able to learn than in the past. This despite them now all being degree-level study students and studying at a university, compared to those who studied at the same institution for diplomas when it was a college. There is ever the possibility that this may be due to a rose-tinted rear-view mirror. It is also true that new CS students no longer enter the university being able to program, and, as the field continues to expand and transform, it becomes ever harder to gain an understanding of the changing field of CS. This may further be affected by the different field knowledge CS students now bring to their studies and the wide range of degrees now being offered.
6.3 Teaching and Learning in Computer Science

For a subject as young as CS, the amount of change in its knowledge content has been significant over its few years. This has been seen already (see figure 3.1, §3.4) to have been reflected in teaching offerings. The group of CS academics interviewed were there, that is they were practicing or teaching during much of this time of transitions and saw students learning change, or not, as these changes have occurred.

An analysis of their responses during the interviews identified five major areas of concern to be discussed here. The teaching of the art and craft of Computer Programming remained the central concern of these CS academics who considered this to be the core skill in CS. There also arose from the data concerns about staffing the subject into the future. Rampant managerialism, or its perception, were often a deeply felt hurt. The wide range of degrees then being offered, in particular recent developments in non-programming degrees were another concern. Finally there was some discussion on the issues relating to students studying CS.
6.3.1 Programming

It doesn’t matter what you want to do, you have to communicate with the machine ... you need programming to actually do things, to build systems. (Dr Grey)

To this group of academics, programming was the essence and core of CS. It was a subject that has within it many of the difficulties of the wider field as a taught, learned and practised skill. Some of the academics programmed in what Dr Blue called ‘the real world’ while others had only practised as academics. Dr Blue identified himself entirely with this core task, programming: ‘If somebody asks me what I do: I’m just a simple programmer.’ Dr Green echoed the opening quote from Dr Grey, stating that ‘Computing is programming right from the start’.

Each of the academics in the study had been involved in teaching programming and all, with the exception of Dr White, still taught programming and saw this as their core task. As Dr Blue stated, ‘I enjoy teaching programming’, and, when asked if he would see out his career as a programming teacher, Dr Green replied, ‘probably’ in a most positive sense. To Dr Blue it could also be ‘frustrating’ teaching students how to code even though he did ‘tend to teach a lot of programming’; he even saw it as being ‘good for the soul’ for students to learn to do programming, a statement that is perhaps only slightly masking how difficult this core task can be to learn.

The academics talked of the changing labels applied to subjects and degrees within Computing - games, networking, telecommunications, hacking, smart systems, etc. - and noted how, nonetheless, they remain at core programmed systems. They talked wistfully of the elegance of many older languages - Algol, Pascal - or their ease of programming of others - BASIC, RPG, FORTRAN, COBOL - compared to what Dr Green called the ‘tough computer programming’ of many modern programming languages - PERL, Java, C++. This toughness was on top of the
naturally abstract nature of programming. As Dr Green said, programming cannot be handled or felt, in much the same way as for the essential truths of his first-degree field, Chemistry.

The move of power away from academia to commerce in terms of programming language design and development was keenly expressed and felt. These academics all entered CS when academia or academically-sponsored bodies were major driving forces in the field. One academic noted the past influence of the para-academic American body, The Association for Computer Machinery, which still set the definitions of academic programmes in the United States of America. They also talked of past academic-driven innovations, such as Algol-60, and non-commercial innovations, such as COBOL and BASIC, and their central role in defining the early field. As things had changed Dr Brown, essentially a telecommunications programming lecturer, complained that Microsoft had driven his field down the entirely commercially-defined route of ASP since 1998. Dr Green similarly queried the reasons for the rapid uptake and ubiquity of Java; when asked as to why something as complex as Java caught on, he replied,

Society in general has moved into a mode whereby people don’t bother to evaluate things, they just say: here’s a new thing, it must be better. The hype that came in with Java said: this is much better, and people just said: OK.

For each language there is a small coterie of companies driving these changes, and that sell associated products: Microsoft with their control over the languages and programming tools for C++, C# and J#; Sun Microsystems with their effective control over Java; Apple with Objective C; Macromedia with Flash (see §3.2). These few companies are in intense competition with each other for dominance in a world-wide arena, and in this commercial battle there is little space for the applied CS academics to be heard with their concerns for the quality, direction and stability of these languages

The academics in this study were also concerned about the lack of time to teach programming theory. Dr Blue, the sole undergraduate CS graduate in the group, felt this strongly,

What he learned in the 1970's had been of regular use to him during a career in this changing field. Students, he said, did not have the field foundational knowledge he had been given and so, as graduates, may not be as able to handle CS change in the future,

I think that maybe we are doing a disservice to our students by making them ready for market.

There were other issues which the academics said made teaching programming a tough charge. The tools that come from the above-mentioned small coterie of tool developers were, as Dr Scarlet stated, 'fiendish and frightening' and not what 'you’d actually want for first year students'. Such tools provided a difficult step from no-knowledge of programming to where the tool expects you to start. The academics also talked of solving such student learning problems as this by sourcing programming tools off the Internet as teaching tools to help teach such complexities as object oriented programming. The Internet was often their first port of call when looking for help in teaching.

Academics at different universities regularly shared experiences in trying to teach programming. As Dr Scarlet stated,

It’s quite reassuring in a sense to find guys from Stanford standing up [at conferences] and saying how much trouble they are having with their programming classes. Some of their students are brilliant and take to it like ducks to water, some of them struggle a bit, and some of them are really turned off by it.

Dr Blue stated with stark openness, '30 years later and we still don't know how to program.' And if you don’t as a programmer know how you program, how do you teach new people to program; this is a perennial weed in CS academia.

Programming, the core skill of the subject, remained the hardest thing for students in CS to learn. Dr Scarlet noted that, 'I don’t think people came, by and large, thinking: programming's the thing I’m keen to do'. It is a peculiar truth that CS is
one of the few fields where students enter without needing any pre-knowledge of
skills in the field. This had changed for the worse over the years, the academics
noted that almost nobody arrived any more being able to program, for which they
again blamed another group of commercial companies: those who developed the
popular computer hardware that young people used. Whereas, in the recent past
popular computers came with easy and popular programming languages (for
example BASIC on Amiga, Atari or BBC microcomputers) their modern equivalents,
(PC, Apple iTouch or Nintendo Wii) did not come out of the box with any, despite
easy, programming tools. Students now generally arrived with no programming
experience unlike their predecessors 10 or more years before who arrived able to
code - albeit, badly. This contrasts with Mander’s (2004) statement that students
arrive at 18 with ten years coding experience; if this is so, it was not true in this
case.

For even the brightest students, such as those on the high entry video games
programming degree, Dr White noted, ‘students struggle appallingly ... with the
Maths and the programming’”. Dr Scarlet noted more generally in learning
programming, ‘[for] those [students] who don’t get it it’s hard.’ Dr Green added
when asked about one particular group of students reading postgraduate taught
masters, and whether they knew much about programming knowledge, ‘It’s poor ...
they come in with no Computing, programming whatever’, and yet they have to
reach a level of mastery within 18 months.

Dr Blue, who was known in the group as the most committed to being an educator,
complained that some new Computing-labelled degree programmes had retreated
from the hardness of teaching programming into increasingly teaching non-
programming skills - ‘edubabble ... handwavingness ... employability ...
transferrable skills’ - which both restricted the time available to teach basic
programming skills to the weaker students and weakened the potential skills of the
stronger students by not exposing them to real languages.

One of the major difficulties discussed is the change from procedural (viewing a
system as being a set of actions) to object-oriented (viewing a system as being a
set of actors) programming. Dr Grey noted,

In the educational computing community there is a divergence of
opinion between [teaching] object-first and basically procedural first.

194
The problem is that although students need to know how to write object oriented programs, they also must know how to program the objects procedurally too. This is harder than simply teaching them to program procedurally; both skills have to be learned at the same time.

In short, programming was still seen as being central, students were seen as being less able, some degrees were considered to be dumbed down by removal of programming, and languages and their paradigms had become increasingly complex.
6.3.2 Staffing

The academics concerned were a diverse group. Two (Drs Blue and White) studied CS as undergraduate students, four (Drs Blue, White, Green and Grey) studied CS as a postgraduate discipline, two had PhDs but no formal qualifications in CS (Drs Scarlet and Black), four had worked as commercial computer programmers (Drs Blue, White, Black and Brown), and one had entered academia from a background in commercial Telecommunications then via College lecturing in CS (Dr Brown.)

Despite the many years, undergoing many field (mainframe, minicomputer, microcomputer, personal computing), sectoral (central institutions, new universities, single Scottish FE/HE sector), institutional (faculties and departments to schools and divisions), and managerial (five heads of school in 9 years) changes, these academics remained upbeat about their personal commitment to the field and to teaching. None made a major issue of, such as, students getting dumber or the field becoming worse, nor did they complain about their teaching colleagues in other parts of the Department’s wide and varied view of CS. As teachers, these were reasonably content people.

However, they tended to see themselves as the last in a line of traditional CS lecturers, and one that would die out with them when they retired or left. They also talked about those who had come before them, who had defined the newly-developing field in the 1960’s, such as Professor A ‘Jack’ Cole who established the Department of Computational Science at The University of St Andrews and who had taught four of the CS academics in the school and two of those involved in the study. The 1960’s generation of CS academics had entered the field with no CS knowledge base, except perhaps Mathematical Sciences, to build upon. The current maturing crop, as evidenced by the CS academics in this study, are their protégés. But, one wonders, where will the next generation of CS academics who will replace them come from?
The field, as reflected in this Department was changing, and this was seen variously as interesting, confusing and bad,

Dr White: Computer Arts. What they lecture is something different.

Dr Scarlet: Construction of Narrative.

To Dr Grey the changes in staffing were an essential part of the current drive to create a niche in the changing world of CS degrees, ‘successful departments are those which have a speciality.’

One of the biggest problems perceived by the academics was the lack of renewal of the traditional CS academy. Dr Black noted that,

We’ve lost a lot of our young software engineering lecturers ... I don’t know what’s going to happen when us old farts go in the next few years. I mean, they’re having difficulty recruiting anyway.

Dr Green talked of how the university had acted to save money as CS student recruitment fell (see figures 3.2, 3.3), ‘We lost the teaching assistants, fellows, and people like Dr Pink, who you’d look upon as a hotshot programmer’. The university had simply found it easier and cheaper to lose the younger, less-established staff than the older, long-established academics. When asked about the issue of whether there were new programming-teaching staff coming up behind the current group, Dr Grey said, ‘we have a big gap.’ This problem had worsened with other university restructuring and departmental mergers; Dr Scarlet blamed these changes for precipitating the loss of younger CS academic staff,

We just got all the engineers. We’re not getting new people. We just got a job lot of engineers who actually forced out new people. Teaching fellows [and] teaching assistants have been made redundant.

Models of how government and universities can work together don’t reflect the realities experienced by these academics. They saw the effects of short-term money-saving schemes over long-term planning, reflecting Ennals’ (2004) model of universities as a cost to be cut in thin times, creating circumstances in which, as Krucken (2003) said, the survival of academic fields is endangered.
One of the external issues affecting staffing which the CS academics identified was that of financial remuneration. Two complained that the lack of pay had directly affected them. Dr Scarlet was generally disparaging about the inability of acadæmia to offer a decent living, despite the huge effort required to enter it,

Why waste your time getting a PhD and waste time getting behind in the housing market and getting pension contributions. ... I'd have been far better off - I mean vastly better off - that is, had I gone off and got a job and doing something for real.

Dr Brown was also critical of poor pay and career prospects in acadæmia. He too felt he was poorly advised to enter acadæmia when he would have had better financial prospects in the wider world. Dr Black, with Drs Brown and Scarlet, felt academic advancement was defined by factors not associated with ability in the field. Drs White and Green had other significant sources of financial support outwith their academic salaries. Only Dr Grey, a senior lecturer married with no children, did not see salary and prospects as a major issue in the post.

There was a reluctance in these older CS staff to keep learning new skills, for example Dr White's reluctance to teach programming in newer languages. It has also already been noted above that Dr Green expressed a hope to spend out his last years only teaching programming. There was also a resistance to being pushed around by new teaching initiatives from out-with CS. Dr Scarlet said about a current proposal to jointly teach a masters degree with another, overseas establishment, 'I don't want to volunteer to go to another country and to teach crap in a university college.'

Dr Black looked forward to retiring in a few months,

I'll probably dabble in artificial intelligence, neural nets; probably; possibly; it's something I've been really interested in for a long time but never had enough time to do it.

This statement is odd – isn't this what an academic traditionally used to be able to do, spending time understanding and researching his field? It was to Dr Black a retirement objective and one that he was bitter not to have had the opportunity to do in his long career. To him this was due to mis-management of the Department and the research structures within the university. Sporn (1996) noted ten years
before that internal management structures were inadequate for the new challenges universities faced, and this was reflected in the experiences of these academics. To them the challenges facing CS were numerous, and none were supporters of the increasing managerialism in the department and university.

Dr Black often talked of the issue of lack of time for teaching preparation. To him this was mainly caused by the ubiquity and incessancy of changes in the field,

Keeping up ... is time consuming. ... and the same thing, again, I try to keep abreast of it. Its area is huge. The expertise is huge and all over the place.

What swamps me is having to constantly update my lecture notes ... it takes a lot of time, not necessarily learning the new stuff, just having to update to the new version of the old stuff ... you have to learn new stuff as well.

Dr Green, like Dr Black, talked of his frustration about not having any time to do any scholarship activities in his academic time. He noted that he missed the 'enjoyable activity' of doing scholarship or research which 'I can't find time to do.' Dr Grey talked of his need to spend time thinking imaginatively about how to teach the ever-emergent new areas in CS. Dr White compared this with the Sisyphean task of needing five years to get together a good set of lectures notes, in a field of annual or more than annual changes.

These constant changes in the field were seen by the academics as a good thing,

To keep abreast of changes in the Computing industry is something interesting because you know that there are interesting challenges in them, and you want to do that. (Dr Brown)

It's wonderful to do a job that interests you. Because you choose - to a large degree its what you do. I teach things and I do things that I find interesting. Its an indulgence. (Dr Scarlet)

Dr Grey saw the aspects of teaching and working in a subject in such constant flux as part of its joy-producing nature,
Dr Grey: It’s always the case I think, certainly at the moment [as we develop the new degrees] to come along and I say, I think I can do that. And it’s a new thing. And, you think, here’s a chance for me to do something new.

Dr White: To get some enthusiasm?

Dr G: Yeh, that’s right.

Drs White and Green saw the hardest part of teaching and doing programming, assembly language, as being the happiest part of being in the field,

Dr Grey: It always struck me as great fun doing an assembly language. I always loved playing around at that level.

Are these people content or discontent with their lot as Computer Scientists? The complexity of their situation was summed up by Dr Brown,

Dr White: It’s a strange thing [academia]. Why do we stay?

Dr Brown: That’s a big question. The answer is normally you perceive job satisfaction. I don’t want to say you have job satisfaction. You perceive it because you think that you have managed to produce something at the end of the day which is knowledgeable students, or graduates. And when they are successful, you are quite happy. You see this as job satisfaction.
6.3.3 Management and Control

There were three aspects of management that bothered the academics: general university management, direction of teaching within their field, and wider policy towards teaching and assessment within the sector.

The academics were outspoken and clear in their criticisms. Drs Brown, Blue and Scarlet were critical of what academic life had become after years of externally-sourced tinkering. To Dr Scarlet it had simply not lived up to what he expected it to have been,

You take the brightest people with the most up to date technical skills and treat them like crap. I'd call it a really appalling career opportunity ... If you're boss is going to be good I'd have been far better off - I mean vastly better off, had there not been glass ceilings ... No, probably the abuse in this job ... shop assistants. We're the guys [who are the] shop assistants.

It took an effort in the interview to steer him away from his criticisms of management, which were heartfelt. Dr Brown was similar; he opened up with an attack upon managerialism and had to be steered off this onto the wider CS topic. He identified ex-academics who had moved into managerial positions as the cause of many of the problems in his job,

Generally speaking I have confidence in the academic world. Outside that it becomes a little bit difficult. And also I've got confidence in the majority of the academic world, except those that move up into management and change hue ... a metamorphosis takes place.

There was a regularly expressed concern about the effect the growth of management had had upon the university in general, as an overhead cost for the institution, which had reduced the status of the academic. Dr White thought it was significant that the senior management shared their physical space with administrators rather than with the academics, partly echoing Dr Scarlet's 'shop
assistant’ jibe by his use of the word ‘janitor’ to describe the new status of the academic in the university.

Dr Scarlet also noted the contrast in recent recruitment and redundancies between administrators and academics,

The same thing has happened here, hasn’t it, if you consider since the current University Principal has arrived, the number of non-teaching posts has gone up dramatically while the number of teaching posts has even gone down, or at least has not gone up demonstrably.

These criticisms of managerialism resonate with Smyth’s (1989) use of ‘AIDS’ to describe its debilitating effects, Dunkin’s (2002) criticism of it as a group of defunct practices, Kogan’s (1999) warnings of the pathological damage it is doing, and Delany’s comparison of modern-day Taylorist educational management structures and mid-twentieth century Fordist approaches to mindless production lines.

Dr Brown considered this problem from a management perspective as being caused by a lack of proper systems in over-managed institutions,

[One of] the problems of staying in academia, you’re always tackling bureaucracy, you’re always challenging rôles from outside where you can’t see logic behind them.

To most of the academics in this group, university management was more a power of bad than of good. This attitude towards management affected the Department in one significant and critical way: by a self-denial of opportunities for leadership in the department. Dr Scarlet - the senior lecturer of longest standing - and Dr White - the Reader in CS - had a telling, short dialogue,

Dr White: Did it cross your mind to apply to be the head of school?
Dr Scarlet: <laughs>
W: I mean, how many nanoseconds did you consider it for?
S: <laughs> No, I didn’t.
W: Me neither.
S: I didn’t entertain the ...
W: For even an instant!

202
This in a Department that had seen five heads of school in less than ten years. Dr Scarlet summed up his position as, ‘Unless things go awry I think I’ve protected my position by having enough to do.’

University management decisions were also perceived as being harmful. A reorganisation of a related department precipitated the loss of academic staff, which in turn widely affected staff morale,

Well there was quite a lot of people who left. It was a big, unpleasant stushie that didn’t actually complete the job. They didn’t end up with a slimmed-down [department] that was viable. They kind of lanced the boil but didn’t kind of solve the problem. It just carried on wasting and festering. (Dr Scarlet)

To Drs Blue and Brown there was also a wider issue of the educational direction of the teaching within the department as it had been affected by recent extr-institutional imposed innovations. Jacoby (1997) noted this tendency to place external norms upon academics, for example currently popular educational jargon which was dismissed by Dr Blue as ‘edubabble’ and ‘handwavingness’. He was also critical of the tendency to have to teach students transferrable skills and group working which he saw as being the responsibility of an employer, not the CS educator. He summed up the effects of this political interference on his teaching,

We teach them silly things at silly times. Group working, you know, that’s one of the favourite in-things at the moment. Team-based problem solving. Group hugs, you know. And what’s even worse, it’s becoming non-technical. I mean, we used to have to work in technical groups, but now its in non-technical groups. I’m sorry, but, you know, why does someone have to know how to manage a project?

Dr Brown continued the criticism of initiatives applied uniformly across academic fields, describing them as ways of creating opportunities for staff to ‘punish’ students unnecessarily. In a similar vein Dr White described how having to make presentations before staff and other students could reduce otherwise competent CS students to tears. Dr Brown had a major issue with the formalisation of assessment regimes, modularisation and university-wide quality assessment regulations,
[We are] punishing them, and, saying, yes, because this time you failed, the next time you do it you only get a pass grade rather than reflecting your original or your actual work. So, um, people, this is something that I don’t like at all. We punish them two or three times.

These academics found it difficult to understand the popularity of external initiatives, perceived as being politically motivated and intended to improve matters, both softened the usefulness of the subject by removing necessary technical skills, and hardened it by making it harder for students to pass their studies.

These CS academics saw themselves as being teachers, yet with little control over what they were timetabled to teach:

Dr White: I was told: you’ll be teaching this thing on ethics and blah-blah-blah to the fourth year students. How would you feel if you were told you’d be teaching that?

Dr Green: I’d have tried to resist ... [it’s] a bit harder if there’s nothing on your timetable ... sometimes you get modules you absolutely hate, and that’s happened in the past.

Dr Brown was also critical of the tendency to use simple quantitative measurement of performance that had invaded aspects of academic life, from the measurement of academic work through quantifying every minute of every day of work, to the linking of funding to student counts by government. To him, there was little scope for resistance as the changes were all ‘politically inspired’ from outwith academia and upon academia leaving little scope, if any, for individual universities to resist these tendencies.

This was a tale that echoed much that is written elsewhere on modern academia, particularly on pervasive managerialism by such authors as Dunkin (2002) and Gumport (2005). There was no echo in the interviews of Rhoades’ (2005) hope that the new professional university managers would be highly and suitably skilled.

Interestingly, the more rules that management applied to the academics, the more gaps that would appear in the system,
Dr Brown: So, why do I penalise them and tell them, yes, Friday afternoon, and, since I don't look at [it before Monday, why] I just simply allow them the extra time?

Dr White: Nobody knows. Nobody knows.

B: Well, if they submit it through email there is a time-stamp, but I don't think that matters. Nobody else looks at it.

W: Nobody else does.
6.3.4 Degrees

The CS department where the interviewed academics were based was one which had gradually moved away from teaching a single CS named degree. Dr Scarlet noted that ten years ago locally, and probably nationally in the sector, there were only BSc Computing degrees. This had grown over time to reach the point where the Department was now noted for its peculiar offerings. Dr Grey, the one most involved in departmental management in the group, saw a pattern,

Successful departments are those which have a speciality. You know that you’re going to this university you’re going to be doing real Computing degrees like BSc Games Technology. So, ‘Hacking’ is another attempt to do the same thing; this is where our [newest] focus is: Ethical Hacking.

To Dr Brown it was significant that these developments had taken place, 'in a young institution and not an institution steeped in theoretical computing.' The examination of recent prospectuses (see §3.4) also showed a tendency for such new or innovative degree development to take place more readily in Scotland in new than older universities.

Each of the CS academics talked about the problem of creating stability in curricula. They saw a stable central structure to the programming orientated new CS degrees, a general CS degree structure that could easily be transported to new subfields, such as games, hacking and networking. However, the need to teach new areas tended to squeeze out CS theory. Dr Grey, who had experience as an external examiner, noted how the older universities were less prone to this while the newer ones were more so,

There are a number [of universities] who teach Java because it is more theoretically based. There are not so many hacks around [it] ... I think this could be the reason that a lot of places never taught C++.
There was a real danger of selling snake oil: new degree offerings being created not by academic need but by the funding structure of the universities. Dr White talked of some new degree developments as being merely ‘kite flying’ and ‘absolute nonsense’ and Dr Scarlet was concerned that these degrees were aimed at capturing a poorly informed and immature audience resulting in an influx of unprepared students he saw as being destined not to succeed in their studies.

Dr Grey discussed how degrees could fail not only because of irrelevance or incorrectness, but merely because ‘I can’t sell BSc Computing because it’s a boring subject [to sell]’. He compared this with newer areas, such as the developments in Smart Systems that have a ‘wow factor’ to excite potential new students. However, there is a dose of realism also in that the teaching funding system stops unnecessary developments of areas that merely excite academic staff, as Dr White stated, ‘We lost “virtual environments” that, uh, was close to my heart. The Virtual Reality stuff. Still, there’s no jobs there anyway’.

The continual minor (relatively to completely new degree titles) developments in existing CS degrees deeply challenged the academic staff. Two academics talked of how they got stuck trying to understand object orientation, and had to get help from the one person in the department who ‘got it’; a third academic in the group gave up. This can be a highly emotional challenge to an academic who, in other circumstances, might be seen as a subject expert,

I was never a database person. And we [Dr Blue’s undergraduate CS studies class] maybe had to look at a bit of databases, and latterly, OK, now a program has to speak to a database. But, you look up the curriculum, look up the textbooks, and all you see is normalisation: 1st order form, 2nd order form, you know, all this stuff. And I was looking through and thinking, this was some kind of big deal. [Then] I thought: this is useless, this is irrelevant, this is silly, this is stupid, I don’t understand this. (Dr Blue)

These academics have sought areas of specialism to concentrate their work upon, limiting the Department’s options for developing and redeveloping degree programmes,

I can’t say that I’ve shied away from something. But of course you can’t do everything so you have to concentrate upon one thing or another.
And since the early 90's with the World Wide Web coming, I've chosen that line. (Dr Brown)

Similarly, Dr White saw himself as being a video games person, Dr Scarlet as a C++ teacher, Dr Black an Artificial Intelligence lecturer, and Dr Grey was a mobile/personal technologies Java programming teacher, each after many years of personal field realignment and each expressed the hope that this decision would see him out for the last years of his employment.

There was discomfort when discussing degrees within the Department which were not based around computer programming. The Computer Arts degree, which produced digital-medium artists, and its staffing, were considered by Dr White as 'totally wrong'. The softer Web Design & Development degree which taught students to create Internet 1 websites was considered to be inadequate as, for Dr Grey, it needed,

the deeply complex coding of the database and security handling of the website. Programming it to do that is hard ... I have big books on my bookshelf about it.

It becomes clear that the constant that these established academics held on to was that CS degrees must be centred around programming. What this programming is may be volatile, but it must be there. Management and government-initiated innovations were measured against their effects on the degrees' possibilities to be conduits for the creation of new programmers. When new degrees are added, they are seen as being best centered around programming too.

If this analysis paints these academics as narrow minded, this would be quite wrong. To Dr Blue there is a fundamental need to produce graduates who can think, and who have enough of a grasp of fundamental theories of learning and the field to be buffered against a career and life of changes. In embracing the changing nature of CS degrees for work. These were academics at home in the subject, applied CS, who measured all innovation and change against the one scale of whether or not teaching for the creation of strong applied CS graduates was enhanced or damaged.
6.3.5 Students

This short consideration of the place of CS students within teaching & learning is taken entirely from the CS lecturers' viewpoints. The validity of the analysis is in this being an analysis of these secondary perceptions, the considered opinions of CS teachers each with decades of experience of this changing field. What follows is therefore a mix of primary (directly from the academics) and secondary (their suppositions on students) data. The data was expressed by the teachers of the students and so is the result of their reflective and reflexive work as professional academics and teachers.

Students were apparently less bookish and had shorter attention spans than previous generations. As Dr Black said,

"Students will not read books. They will not read papers. They couldn’t even find them if you asked them to. But, if you give them a link to a website they’ll find it. And, in fact, they’ll dig."

This perceived change in the mindset and knowledge-finding skills of students was also expressed by Dr Grey,

"I’m now convinced that students look at things, and they switch between TV channels, and they watch ten minutes of that, fifteen minutes of that, and when the adverts are on they watch them, and I think they learn the same way, with a bit on their iPod and a bit on the radio."

There was also a concern about the leap between pre-university knowledge and what was being taught to the students. Dr Grey was concerned over the gap between not being a CS student to being one, with no previous understanding of the field. The problems this caused disengaged students, ‘you’re teaching in ways that require quite a sophisticated systems approach to everything, and they [the students] may get turned off.’ Dr Scarlet combined this with a concern that
students opted to come on attractively-name CS degree courses without the critical understanding of the central role of programming – which to almost all new students was an unknown skill - in the degree study,

People who take Computing courses quite often have in mind not doing programming. I don’t think people applying for a Computing course think, ah, I want to program. They are increasingly unfamiliar with what Computing is.

The problems of getting students to understand systematic ways of thinking via complex industrial programming toolsets, could be heartbreaking for both teacher and student. It was this form of thinking that was required by students wishing to become a computer programmer that was of greatest concern to their teachers. Dr Green talked of, 'constantly searching for different ... similes to use to illustrate what you’re trying to tell them about ... [how to] organise their thinking.' Dr Scarlet said there were large numbers of students who could not make the necessary mental leap into CS thinking. For, failing students,

It's demoralising, because they don't get it. They write the wrong things all the time so they get errors all the time. This gives them robust and negative feedback all the time. Everything gives them trouble, everything gives them grief, everything they try fails. (Dr Scarlet)

This is inherent in the nature of computer programming. It is an exact science where errors are dumbly interpreted at face value by the computer which provides, in many cases, a dumb answer to explains the problem the computer has with the program code, not the error’s cause as made by the programmer. Dr Grey warned against thinking simplistically and blaming students for their 'not getting it'. Academics, he continued, trying to help struggling newbie programming students, should always remember that, 'we [have] had years of hacking around with FORTRAN and Pascal and things'. (Dr Grey)

Dr Black criticised the current paradigms of programming over the older, careful, systematic methods,

If you go back to when I was a student ... it took two to three days to get a turnaround [the time for feedback of errors in a computer program’s syntax or logic]. You extracted as much as you could out of
your program, so, you didn’t just correct one error, you looked through the entire program looking for other possible errors which you might have made ... I don’t want to go harking back, “in my good old days this and this and this”, but they’ve lost that discipline ... and just go <keyboard-clicks> wrong <keyboard-clicks> try this <keyboard-clicks> and just ramble around rather than thinking logically.

He missed the hard-copy printouts used for error and logic-checking and contrasted this with students using the computer screen instead, not simply as this could cause simplistic thinking on what could be an enormous computer program of up to 500,000 lines, but also because there was a perception and learning problem inherent to visually-based computer systems, ‘[if] something appears on the screen, it’s bound to be right. Students never check to see if it is correct or not’.

It was noted that students worked differently from the traditional university’s model where the institution sought to hold and control all its major learning resources. Students all had their own computers at home, often better than those at university, with access to better software development tools and the entire Internet and so they tended to work at home more than in university labs.

The kinds of students entering CS were also variable. Dr Brown, who told how he was educated in a traditional way where he was born abroad, thought that the cause was a poor Scottish state education system. Dr White perceived the intake to certain degrees as being ‘pretty good’ but he also noted that with the removal of the binary line between the established universities and the Scottish Central Institutions (CI’s) better students of the kind that used to come to the CI’s pre-binary line removal had almost all gravitated towards the established universities.

Students were perceived as continuing to have a high degree of trust in academics and in academia. None of the academics talked about being out of their depth in relation to students’ increasing applied field knowledge; all felt positively about their students. The student trust in academia was a source of concern for the CS academics. Students came on degrees because they were, as Dr Grey stated, sold to them through, as Dr Green said, the marketing of vocationally/excitingly entitled degree courses. Dr Blue had concerns that there was a slippage towards student-friendliness over real skills in degree studies which would affect graduate careers,
but Dr Scarlet reckoned most of the core knowledge required was still taught, albeit not as explicitly as it once was.

There was also a concern that the inherited CNAA structure of taught degrees as used by the new universities did not allow students the flexibility of choice in studies which had been available to the academics when they were young undergraduates at the established universities. Also, students failing these highly-structured post-CNAA degrees in programming-orientated study programmes were left with few options for alternative study when they hit the wall of ‘I can’t program’.
6.3.6 Conclusions

CS is a young field and the academics interviewed could be viewed as being the second generation of teachers in the field. To these teachers CS was programming and it was around the problems related to teaching the abstract art of programming that concerns focussed on the pressures, joys and pains of teaching in the field. Programming had changed considerably since these CS academics’ youth when it was strongly academically-driven, to now being a craft that was almost entirely defined by commercial forces outwith the academy. These companies provided commercial programming tools that were simply too difficult for new programmers to learn with. In addition, the academics had so little time to reflectively devise teaching programmes that they had to scrounge around the Internet to find ways and means to teach the ever-changing core skill of programming.

The field of CS, at this site, was fractured into programming and non-programming degrees. Some of these new degrees were deemed to be not worthwhile offering, others were beyond the understanding of these academics, still others remain at core traditionally CS despite skillful market-driven naming, and the old-fashioned CS-named degrees were expected to be lost within a few years.

The political pressures on CS were deplored. Educational trendiness was seen to have made learning more difficult for weaker students, creating more areas where they could achieve failure. Managerialism within acadæmia was looked upon as a force for bad that had created structures that so repelled these established academics that filling senior academic positions was difficult and the most senior departmental positions had been long been occupied by short-term appointees.

The interviewees were ageing protégés of the first generation CS academics who had defined the taught field. Below the second-generation academics in this study there was now a generational gap, caused by poor government funding regimes and poor managerial decisions. This lack of medium or long term planning affects the future teaching of the core skill of CS: programming.
Despite significant pressures and many areas of personal disappointment, the academics enjoyed being in their chosen field, none regretted having entered CS (although two did regret having become an academic), and some even looked forward to continuing to add to the knowledge within the field after retirement in such a way as they had not been able to do as teachers.

There were worries about students expressed in ways that showed a personal concern and attachment by the academics. There were worries that degrees were being crudely marketed to a generation with no understanding that the core skill of CS is programming. There was little flexibility in management systems to help students find their feet again when they stumbled in the rigid and failure-orientated assessment systems. However, finally, and within the quiet pages of this document, it should be noted that as management create rules that can harm students, these highly qualified and experienced professional educators have ways to find their way around the rules, to the betterment of students and CS teaching.
7. Conclusions

7.1 Introduction

This is a study that was provoked by the author's need for greater understanding of why he has had difficulties coming to terms with understanding and practicing in a field - applied academic Computer Science - which he entered in 1975, and which has been subject to a range of forces causing a range of changes.

In order to do this a case study was performed, at a 1990's entitled Scottish university, involving a small group of long-experienced CS academics, who had long worked together. These academics were interviewed in a mainly-informal way and asked to recount their tales of the changes they had seen, the causes of the changes, their successes, failures, understandings, past regrets, present disappointments, future plans, hopes and fears of being an applied CS academic.

I was not disappointed by what they said. The conversations were illuminating, sometimes in concord with the literature, at other times in disagreement. The academics spoke openly, freely, emotionally, clearly and cogently. They also spoke at times in a deep technical jargon that I have attempted to extract deeper meanings from, as necessary.

In concluding this report I would return to a basic truth of this case study. It is a bounded, specific instance in time, place and in the lives of those involved. It is not intended as a examplar from which generalities can be found. That would be unverifiable. However, it has helped me understand the questions I posed and, I hope, may help others to better understand situations of change.

The rest of this short conclusion will go through the research questions (see 5. The Method above) and tie these together with the findings.
What is happening to knowledge production in CS?

The findings are one of changing knowledge. Change and the control of change is a primary concern of these academics. CS knowledge, as used by these academics is not of an academic kind. These academics do not produce knowledge in CS, nor do they gather any knowledge from wider CS. Instead knowledge is to be sourced and found from other parties outwith academia.

The main producers of applied CS knowledge are the commercial computing companies. Programming languages are created, disseminated and updated by major commercial computing companies. As these academics seek to teach students to become commercial programming graduates, so they must use commercial programming tools. The knowledge on these tools is found in short-life textbooks and on the Internet.

CS knowledge is also produced by lay users of computers. There is a significant lay community who create software and software development tools. Although these lay communities might contain academics, the latter label has no particular meaning. These lay communities are a further source of help in teaching programming to students.

There is also knowledge that new students bring in with them. This is experiential knowledge on the use of computers. There is little evidence in the conversations that new students bring in any useful knowledge on the fundamental truths of CS, or even programming skills. In this, new students can be grouped in the same knowledge category as end-users.

And so knowledge production in applied CS is from three principal sources:

- Commercial computing companies – concerned with software creation and selling
- Lay-users – concerned with software creation and use
- End-users – concerned with software use
7.2 What kind of knowledge field is CS?

Those involved in the study were a group of CS academics in a new university. This is significant as it is a site concerned with the teaching of useful graduate skills and foundational knowledge. There is a balance in a four-year degree programme between skills and theory and, at this university, the balance is tilted towards skills over theory.

Students coming to this university can expect to be taught CS in a practical way. These practical skills are ever in flux. Languages come and go, albeit rather slowly, and versions of languages change rapidly. A new version of a commercial computer language, such as Java, may appear within a student’s course of study. New operating systems appear almost biennially (on, e.g. Apple Mac computers) or every few years (on MS Computers) or sparodically (on Linux computers.)

An amount of this CS knowledge is hidden, either deliberately, or by its nature. Commercial programming tools for some platforms (e.g. Nintendo Wii) are not available for academic use. Information on how to use products can be deemed commercially sensitive and not be issued to outsiders (e.g. certain internal workings of commercial operating systems). Knowledge on other commercial products may only be available if a need can be proved (e.g. programming some mobile phones.)

The core skill of CS, programming, is a hidden skill. Few would claim to understand how this works in the mind or how to teach it. It is bound up with being an abstract skill, performed in an exactly-defined grammar, expressed in a contradictory combination of object and procedural thinking. It is something, to quote one of the academics in the study, you get.

The knowledge base of CS is enormous and is expanding, contracting, emerging and re-emerging. New CS products – hardware, software, applications – appear every year in a market that covers consumer electronics, telecommunications, defense, the arts, and many others. The CS academic must both be a specialist in

217
his own ever-changing subfield, and have a well-established and broad understanding of the changing nature of knowledge in the field.

Much of the knowledge in applied CS is flakey. This is not a field that is driven by careful, peer-reviewed journals. Knowledge is found in difficult to provenance corporate press releases, bulletin boards, terminological-laden discussions, books, etc. Much of this knowledge is also time-stamped, reducing its direct effectiveness, a problem created by commercial CS ability to store large amounts of online data.
7.3 What underpinning beliefs about CS do academics at this site hold?

Strong ones.

At core they believe that CS is programming and that anything that is not programming at core is not CS. Anything that affects their teaching of programming, whether positively or negatively, is important and they are willing and able to express considered views upon.

Each degree programme in the department is judged by its teaching programming content. A degree is sound and right if it teaches programming. The academics do not express any concern about what kind of programming that is – HTML, C++ or an assembler – as long as it is programming. This programming is then applied to a specific area of practice, for example, video games creation, computer security or mobile telephony.

Any degree without programming is not a degree if the degree pretends to be CS. However, a degree with no programming could be a degree if it has a different field of application, for example digital arts, but the CS academics are not sure exactly what to make of such new degrees.

External effects upon the teaching of programming are deplored. New teaching fads – group working, presentation skills, etc. – are a waste of time as they distract students from programming and the thought processes required to become a good programmer. Government funding regimes have both benefitted CS by bringing in new students and allowing the development of new programming-centred degrees, and hurt the field by causing the loss of staff and encouraging the development of non-programming degrees.

Internal university management structures are an anathema. They have broken down trust between the academics and management and have created a perception that senior posts for academics beyond simply teaching are for Quislings. Management innovations through the creation and maintenance of structural rules
that affect degree programmes are considered to be factors that seriously damage the delivery of good quality teaching, and are there to be nodded at, ignored or followed as the academics perceived are in the best interests of the students.

The academics hold beliefs in CS as a theoretical field too, but this does not clearly map onto their teaching. Some believe that all knowledge of fundamental theoretical CS is no longer taught to students, while others think it is still quietly there under the bonnet.

CS is a field of programming, and one to be defended strongly. Outsiders are outsiders. This particular group of insiders paint themselves as the last few defending Rome against the Barbarians.
7.4 What processes of teaching and learning follow from these beliefs?

This is a deeply dedicated group of CS teachers, strongly committed to teaching relevant skills, in relevant way to students who they are preparing for a career as a software developer in a range of selected fields. Their efforts are focused on this target. It would be incorrect to read from the interviews that this is necessarily a group who spend a long time griping about the perceived horrors of management and change. This was not so: they enjoy their teaching and they enjoy the efforts that must be made to keep their teaching relevant and correct.

If they do, as some said, believe that theoretical CS is very important, then it does not drive them to make any particular changes to their teaching. Instead, much of their effort is made in seeking, validating, learning and utilizing sources of knowledge in teaching.

They acknowledge that administrative structures are inflexible and unresponsive to change, so they teach within these structures, but they teach relevantly. It is interesting that although the degrees have very different names and target careers, the academics still saw themselves as a coherent group teaching computer software development. If the group were divided into teachers of games development, artificial intelligence, mobile telephony, telecommunications, etc. then they would each be in a group of one.

There is evidence that the academics are close to their students. They spoke extensively and knowledgeably about the kinds of students, the problems they have with the field, and students' responses to teaching approaches. This is a group that teaches their students in close proximity in order to ensure that the students are learning to program.

The group was dismissive of applying external educational ideas to their teaching, even if these would be found in some textbooks. For example, there is a corpus of literature on group work in CS, but at least one member considered this to be a
waste of time in undergraduate teaching. As so many of the academics came in with commercial programming experience behind them they have drawn partitions between what is to be taught at university and what is to be learned on the job. As a university calendar and laboratory structure does not easily map onto the real-world of programming, the academic has to teach what he can, where, how and when he can. Therefore some aspects of applied CS knowledge cannot be taught.

Lecture notes are of short life as they reflect the rapidly changing state of knowledge in CS. Teaching is ever incomplete and never definitive as there is neither enough time to grasp this knowledge, if it can be grasped, nor the time available to create the kind of reflections on knowledge that might be more generally useful in other academic fields.
7.5 How far have things changed over the years?

Not much and a lot.

If, as this group does, CS is considered to be programming, then the subject has changed very little during my career from the mid 1970’s until now. This, of course, is a result of the bounding created by these academics, delimiting what is in and what is out. But it would be wrong to consider this to be an artificial or invalid boundary as it is clearly expressed in significant degree studies and reflects the academics’ experiences both before and during their careers in academic CS.

Programming has not stood still. It has moved from the simple languages, often academically-defined, of the Algol family, through commercial languages defined by such government bodies as ANSI, including COBOL and Fortran, and into a range of languages that are now almost entirely commercially defined and controlled. The academic seeking to teach these languages must go to people who know about them, people who are unlikely to be in a university.

Tools to teach programming are now more varied in quality, sources and useable lifetime. The paradigms of programming have also changed towards object orientation and away from proceduralism. This is a difficult co-concept to learn and to teach and one that students often fail to understand. In this the teaching of programming has become more difficult.

The degrees being taught are in flux. New degrees, at this site, appeared every few years. The academics must be willing to reskill and reorientate their teaching, often at very short notice and with little prior knowledge or skill-base. On the other hand each academic in the group still considered himself to be a particular subfield specialist.

A lot of the changes in structure and management of the department and the university seemed to have a limited effect upon the academics. There is some evidence that they have distanced themselves from such innovations or may even

223
actively oppose them if territory that is considered significant is encroached upon. Some aspects of management were even seen as being laughable. This could be seen as a healthy scepticism in that there was little evidence of psychological stress in the conversations.

The major change would be the increasing age of the academics. Some had arrived in the 1970’s, others in the 1980’s, but none in the 1990’s or beyond. There is a question mark over the renewal of the CS academy at the site, as a group who believe in the centrality of programming as the main skill to be taught and learned. There are few young academics of this church in the department. That it is easier to create new, attractive degree programmes than employ a new programming-orientated lecturer, may define the future direction of this department.
7.6 Where is change driven from?

Within the field of applied CS, change in knowledge is driven from the sellers and users of computers. This is a commercial field, purchased by people and business, but one with a communalist arm. The teacher of applied CS, seeking to create graduates who can create systems, must gain his knowledge of the field from these sources.

Computers are a means to an end, enabled by the software that runs on them. This software is created using tools developed by commercial companies and non-profit developers. Knowledge on the location of these programming tools and how to use them must be gained by the teaching academic in order to teach the skills in using these tools to his students.

This knowledge is ever in flux as companies and individuals seek to create better tools to create better applications. And so, they change the tools, often regularly. These changes must be reflected, eventually (for no academic can ever be entirely up to date in his CS teaching) in teaching programmes.

End users use the software developed for them, provide feedback, and the software may then be updated and changed. The academic must find ways to gather this ever-changing, experiential, word-and-image mediated knowledge. This knowledge is accumulative, reflective and reflexive and comes usefully packaged in the students who attend classes. So every new intake of students is an opportunity to plug into a source of new CS knowledge.

It would be wrong to think that these academics are 100% reflective and reflexive, essential though these skills are to the practising CS academic; or, on the other hand, that these professionals have no effect upon the discipline of CS. They are at the cutting edge of degree development and as teachers could be considered to be setting a standard for other applied CS departments to follow. It is true that in CS terms they do not create knowledge, but they do create packages of understanding in their degree programmes that also alter the field as graduates are produced with
skills that will be useful to expanding areas of applied CS and so will skill-up these areas to create better products.

Ultimately, the changing nature of CS could be considered to be one that is oiled by CS academics such as those in this study. As they looked back upon the past greats who made CS what it was for them, for example, Jack Cole, Ian Somerville and Tony Hoare, so it may well be that they too will leave a lasting impression in the changes they have created in CS through their carefully directed teaching practices.
**Glossary of Terms**

Computing Science terms unlikely to be obvious to the non-specialist

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3G</td>
<td>mobile phone that can receive, e.g., television</td>
</tr>
<tr>
<td>5 hole tape</td>
<td>obsolete method for storing data and programs</td>
</tr>
<tr>
<td>algorithms</td>
<td>pieces of logic in a computer program</td>
</tr>
<tr>
<td>autocode</td>
<td>obsolete programming language group from the 1950’s and 1960’s</td>
</tr>
<tr>
<td>backend scripting</td>
<td>program code that runs on an Internet server, e.g., the Google search engine</td>
</tr>
<tr>
<td>(server-side scripting)</td>
<td></td>
</tr>
<tr>
<td>Bioinformatics</td>
<td>the use of computers to model biological systems</td>
</tr>
<tr>
<td>bitwise AND</td>
<td>very low-level data bit handling operation</td>
</tr>
<tr>
<td>C Sharp</td>
<td>programming language devised by Microsoft as an alternative to the standard language, Java</td>
</tr>
<tr>
<td>(C#)</td>
<td></td>
</tr>
<tr>
<td>cache</td>
<td>temporary storage area for data you may need to look at again quickly</td>
</tr>
<tr>
<td>digital art</td>
<td>art produced on computers</td>
</tr>
<tr>
<td>dot net</td>
<td>an all-embracing term for developing software for Microsoft Windows systems</td>
</tr>
<tr>
<td>(.NET)</td>
<td></td>
</tr>
</tbody>
</table>

227
<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge</td>
<td>monthly magazine discussing events in the video games developer community</td>
</tr>
<tr>
<td>ethical hacking</td>
<td>legal attempts to breach computer security in order to test it and understand illegal breaches</td>
</tr>
<tr>
<td>Games guru</td>
<td>leading thinker in the video games industry, e.g. Peter Molyneux</td>
</tr>
<tr>
<td>hello world</td>
<td>the first program a programmer will write in a language new-to-him is one that outputs the text, 'hello world'. It creates a recognised initial understanding of the language.</td>
</tr>
<tr>
<td>Internet 1</td>
<td>Internet pages with predefined text, images and sound coded in HTML. Internet 1 pages look much like static magazine pages. Contrast with Internet 2 which has view-time moving content such as television, radio and games.</td>
</tr>
<tr>
<td>mantis exponent</td>
<td>a method of storing real numbers (e.g. .12.034) in binary format (e.g. 10110110).</td>
</tr>
<tr>
<td>MCV</td>
<td>the weekly games industry market magazine, only available by paid subscription and never in newsagents.</td>
</tr>
<tr>
<td>object orientation</td>
<td>designing and programming computer systems by considering their components as parts that function, communicate and remember.</td>
</tr>
<tr>
<td>(objects, object oriented, object orientated, UML)</td>
<td></td>
</tr>
<tr>
<td>program counter</td>
<td>the computer's way of remembering which line of the program code it is currently executing</td>
</tr>
<tr>
<td>PSP</td>
<td>Sony's handheld games console</td>
</tr>
<tr>
<td>Rekursiv</td>
<td>Obsolete computer circuit board for DEC VAX computers</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>smart systems</td>
<td>computer systems that respond intelligently</td>
</tr>
<tr>
<td>teletype</td>
<td>obsolete electronic typing and storage device</td>
</tr>
<tr>
<td>trees</td>
<td>a way of logically and sometimes physically structuring data</td>
</tr>
<tr>
<td>VAX Cluster</td>
<td>a method for linking together the obsolete computers, DEC VAX.</td>
</tr>
<tr>
<td>virtual bonking</td>
<td>simulated sex using a computer, e.g. the infamous hot coffee mod in the Grand Theft Auto San Andreas video game.</td>
</tr>
<tr>
<td>Z80</td>
<td>really old, but still popular computer processor</td>
</tr>
</tbody>
</table>
Bibliography


232


Dundee College Of Technology (1968) *Dundee College Of Technology Full-Time Prospectus 69 70*, Dundee: Dundee College Of Technology.


Dundee College Of Technology (1973) *Dundee College Of Technology Prospectus 1974-75*, Dundee: Dundee College Of Technology.


Dundee College Of Technology (1979) Dundee College Of Technology Prospectus 1980-81, Dundee: Dundee College Of Technology.


Ecclesiastes, Chapter 12, Verse 1 [New International Version].


Glasgow And West Of Scotland College Of Domestic Science (1963), The *Glasgow And West Of Scotland College Of Domestic Science (incorporated) Prospectus Session 1964-65*, Glasgow:Glasgow And West Of Scotland College Of Domestic Science.

Session 1965-66, Glasgow:Glasgow And West Of Scotland College Of Domestic Science.


244


Hebrews chapter five [New International Version].


246


Leydesdorff, L. (2005), While A Storm Is Raging On The Open Sea: Regional Development In A Knowledge-based Economy, Journal Of Technology Transfer, 31, 189-203


251


254


Queen's College Glasgow (1974), The Queen's College, Glasgow (formerly College Of Domestic Science) Prospectus Session 1975-76 Centenary Year, Glasgow: Queen's College, Glasgow.

Queen's College Glasgow (1976), The Queen's College, Glasgow, Glasgow: Queen's College, Glasgow.

Queen's College Glasgow (1977), Prospectus The Queen's College, Glasgow, Glasgow: Queen's College, Glasgow.

Queen's College Glasgow (1978), Prospectus The Queen's College, Glasgow, Glasgow: Queen's College, Glasgow.

Queen's College Glasgow (1979), Prospectus The Queen's College, Glasgow, Glasgow: Queen's College, Glasgow.

Queen's College Glasgow (1980), Prospectus The Queen's College, Glasgow, Glasgow: Queen's College, Glasgow.

Queen's College Glasgow (1981), Prospectus The Queen's College, Glasgow, Glasgow: Queen's College, Glasgow.

Queen's College Glasgow (1982), Prospectus 1983-84 The Queen's College, Glasgow, Glasgow: Queen's College, Glasgow.

Queen's College Glasgow (1982), Prospectus 1983-84 The Queen's College, Glasgow, Glasgow: Queen's College, Glasgow.

Queen's College Glasgow (1983), Prospectus 1984-85 The Queen's College, Glasgow, Glasgow: Queen's College, Glasgow.

Queen's College Glasgow (1984; Undated; Probable Date), Prospectus The Queen's College, Glasgow, Glasgow: Queen's College, Glasgow.

Queen's College Glasgow (1985; Undated; Probable Date), Prospectus The Queen's College, Glasgow, Glasgow: Queen's College, Glasgow.

Queen's College Glasgow (1986; Undated; Probable Date), Prospectus The Queen's College, Glasgow, Glasgow: Queen's College, Glasgow.

Queen's College Glasgow (1987; Undated; Probable Date), Prospectus The Queen's College, Glasgow, Glasgow: Queen's College, Glasgow.
Queen's College Glasgow And Glasgow Polytechnic (1992). 1993 Entry Full-Time Prospectis Glasgow Polytechnic The Queen's College Glasgow Soon To Be Glasgow's New University, Glasgow: Queen's College, Glasgow & Glasgow Polytechnic.


257


260


University Of Dundee (1967), The University Of Dundee Prospectus 1968-69, Dundee: University Of Dundee.

University Of Dundee (1968), The University Of Dundee Prospectus 1969-70, Dundee: University Of Dundee.


University Of Dundee (1970), The University Of Dundee Prospectus 1971-72, Dundee: University Of Dundee.

University Of Dundee (1971), The University Of Dundee Prospectus 1972-73, Dundee: University Of Dundee.

University Of Dundee (1972), The University Of Dundee Prospectus 1973-74, Dundee: University Of Dundee.

University Of Dundee (1973), The University Of Dundee Prospectus 1974-75, Dundee: University Of Dundee.


University Of Dundee (1976), *The University Of Dundee Prospectus 1977-78*, Dundee: University Of Dundee.


University Of Dundee (1979), *The University Of Dundee Prospectus 1980-81*, Dundee: University Of Dundee.


University Of Dundee (1992), *The University Of Dundee Prospectus 93*, Dundee: University Of Dundee.

University Of Dundee (1993), *The University Of Dundee Prospectus 94*, Dundee: University Of Dundee.

University Of Dundee (1995), The University Of Dundee Prospectus '96, Dundee: University Of Dundee.


University Of Dundee (1999), The University Of Dundee Prospectus Entry 2000, Dundee: University Of Dundee.


University Of Dundee (2001), The University Of Dundee Prospectus Entry 2002, Dundee: University Of Dundee.

University Of Dundee (2002), The University Of Dundee Prospectus 2003, Dundee: University Of Dundee.

University Of Dundee (2003), The University Of Dundee Prospectus 2004, Dundee: University Of Dundee.

University Of Dundee (2004), The University Of Dundee Prospectus 2005, Dundee: University Of Dundee.

University Of Glasgow (1979), University Of Glasgow Undergraduate Prospectus For Entry In 1980~81, Glasgow: University Of Glasgow.

University Of Glasgow (1980), University Of Glasgow Undergraduate Prospectus For Entry In 1981~82, Glasgow: University Of Glasgow.

University Of Glasgow (1981), University Of Glasgow Undergraduate Prospectus For Entry In 1982~83, Glasgow: University Of Glasgow.

University Of Glasgow (1982), University Of Glasgow Undergraduate Prospectus For Entry In 1983~84, Glasgow: University Of Glasgow.

University Of Glasgow (1985), University Of Glasgow Undergraduate Prospectus For Entry In 1986, Glasgow: University Of Glasgow.

University Of Glasgow (1986), University Of Glasgow Undergraduate Prospectus For Entry In 1987, Glasgow: University Of Glasgow.

University Of Glasgow (1987), University Of Glasgow Undergraduate Prospectus For Entry In 1988, Glasgow: University Of Glasgow.

University Of Glasgow (1988), University Of Glasgow Undergraduate Prospectus For Entry In 1989, Glasgow: University Of Glasgow.

University Of Glasgow (1989), Glasgow University Undergraduate Prospectus 1990, Glasgow: University Of Glasgow.

University Of Glasgow (1998), University Of Glasgow Undergraduate Prospectus For Entry In 1999, Glasgow: University Of Glasgow.


266


Wright, IM (2006), General Council Agenda, St Andrews: University Of St Andrews.


Appendix A

The Accepted Research Proposal

The names of real places and people have been removed.

Abstract

Computing Science (CS) is facing several challenges at this stage of its short history. Students are reported as being dissatisfied with their learning experiences, potential students are opting not to study the subject, and current students are leaving soon after starting their studies in significant numbers and proportions. This is a phenomenon recorded and debated widely in the UK and the USA. There appears to be a gap between what is expected by students and what is delivered by faculty\(^1\) in classes. In addition there is widespread acceptance of the volatile, bloated and transdisciplinary nature of the knowledge domain by those within CS, which is an ongoing expansion effect. Further, and perhaps unsurprisingly, CS faculty are not agreed as to the curricular requirements of a CS degree today and into the future.

This thesis explores the hypothesis that these effects are related to the volatile nature of the knowledge-base of CS, and that this may contribute to an epistemological gap between what CS knowledge is to CS students and what it is to the faculty.

It is acknowledged that other factors may be significant but the principal focus here is on the changing nature of CS knowledge as it is understood, delivered and received by faculty and students.

The topic will be investigated through a case study of CS faculty and students within the Department of Computing at the University of *******. The selection of the site is,

\(^1\) For the purposes of this report the term faculty will refer to teaching staff.
to a degree, opportunistic. However the department is typical of CS faculties that are active in promoting innovative approaches to knowledge in CS. Thus the case should enable generalisable findings about the nature and processes of knowledge change in CS higher education programmes.

The investigation is shaped by the literature on changing knowledge in the Academy and in CS, and draws on a number of data gathering techniques, including focus groups, interviews and observation to arrive at an analysis of the views of CS knowledge held by faculty and students.

And so the proposal is for a designed study in several parts, based upon informed investigations of the perceptions of the issues around CS degree programmes by ***** University's faculty and students, and founded upon an understanding on the literature relating to knowledge domains. It is intended to locate and identify the significant factors that are perceived as being the knowledge that is CS by the two groups (faculty and students) and to comment upon differences of perception of the CS knowledge domain.

The study will be one that questions faculty and students, in focus groups and open questionnaires on their CS knowledge, knowledge delivery, study behaviour and attitudes to the learning environment[^1]. This will be underpinned by studies on knowledge in general and in CS, and in the ways realities are constructed around this knowledge.

---

[^1]: that is, to include such as how the study spaces and facilities relate to their perceptions of what CS is.
Background

Computing Science (CS) as a discipline dates back to three phases of development in the early and mid 20th century. Its pre-technological roots are in philosophy (logic structures and abstraction)\(^1\) and mathematics (problem definition and solving), as defined by Von Neumann, Turing, Church, et al.\(^2\). The first computers were developed in World War II, principally by the Americans and British, for defence applications using developments in analogue electronic engineering (radar) and digital mechanical engineering (data encryption and codebreaking). As a digital electronic engineering discipline it began in the 1950s out of the work of the US Department of Defense (COBOL, FORTRAN), IBM (PL/1, OS/360) and The University of Manchester (Leo I).\(^3\)

By computer science I mean the body of knowledge\(^4\) concerned with the use of the digital computer. The digital computer has itself changed very little since the 1960s. It is a stored program device that processes a data stream through the central processing unit (CPU). Data is held in binary digits (bits) either in fast, volatile memory (usually called RAM) that is lost when the program stops, or on slow, non-volatile memory (e.g. disks, tapes) that continue to exist after the program stops. This continues to be true of such as the Sony Playstation 2\(^5\), an Apple Mac mini, a Nokia mobile phone, the Cray supercomputer in the National e-Science Centre and all other computer-centric services.

However, the other part of CS – the creation of programs – has been in an almost constant state of upheaval. The earliest digital computers were programmed directly in the binary language of the computer’s CPU (e.g. the word ‘10011011’). In the mid-1950s these were translated into languages (assemblers) that stated the meanings of the binary words (e.g. ‘LOAD R1 1011’). By 1958 the US Department of Defense had defined two new languages that were non-computer specific for mathematical

---


\(^3\) the precise history of the appearance of the first digital, stored-program computers is, as for many inventions, disputed, in this case by the Americans and the British.


\(^5\) in the UK computing keywords are generally spelled in American English.

\(^6\) All trademarks and copyrights are acknowledged.
formulæ (FORTRAN, the formula translator) and data information (COBOL, the common business oriented language) processing purposes. The 1980s saw a proliferation of BASIC language dialects and database languages (e.g. Powerhouse, Oracle). This has now settled down into a field dominated by the commercial giants, such as Microsoft (C), Sun (Java) and Macromedia (Flash). In specific subfields, such as video gaming, programming is controlled and driven by major commercial companies that include Sony, Nintendo and Electronic Arts.

Although the computer qua computer has not changed its primary architecture (CPU + storage + data + program), its use has changed enormously over the past fifty years. Today's typical customer-facing applications can be purchased in the High Street as the home computer, video game console, mobile phone, Internet and personal digital music player. Similarly, the computer as a hidden device is central to white goods (e.g. washing machines, microwave ovens), digital subscription services (e.g. telephones, cable TV), industry (e.g. finance, retail, distribution and defence), black goods (e.g. TV, DVD player, cameras) and government (e.g. tax collection, immigration control, police).

In the personal sphere the computer has grown as a visible lifestyle device, from the home and games computer of the 1980s, through the mobile phone in the 1990s, to the entertainment-to-go iPod of the 21st century. New software applications (programs) have also changed how people, groups and societies interact through the use of such as e-mail, chatrooms, blogs, websites and texting.

It is against this backdrop of an ever-changing field that those working within the academic discipline of CS must seek to understand and teach.
The Computer Science Faculty and Students

As the field of CS has grown¹ along with student numbers and in accordance with UK Government funding policy, so faculty numbers have grown too. This growth was most evident in the late 1980s to late 1990s. Initiatives, such as the new blood scheme for the Scottish Central Institutions (CI's), the continued open-ended funding of students on CS degree programmes, and the unlimited funding of students on postgraduate information technology and software engineering programmes at the CI's, produced a boom in faculty numbers that was paralleled elsewhere in the UK and in other countries.

These boom times continued until 2001 when the year-on-year 15%–20% growth in applications to CS degree programmes was reversed to a similar year-on-year decline that has continued to date²,³. At the same time, the numbers of students dropping out of CS degree programmes in early years has become a serious problem as has the feedback indicating dissatisfaction from students taking CS degrees across the UK. Faculty have started to lose their posts - part-time, short-term contract and full-time contract – across UK CS departments⁴ and doubts are being openly expressed about the survival of CS as a major, pan-university subject beyond 2010. This trend has also been noted in the USA⁵,⁶ and Australia⁷.

⁴ For example, Bolton, Abertay and Paisley universities terminated casual faculty contracts and did not renew expired short-term contracts in their CS departments for 2005–6; Derby laid off 1/3 of their full-time CS faculty too.
⁷ Personal emails and conversations with CS faculty in both countries.
A few UK CS departments have diversified into CS applications fields including digital media studies and web-site design. Others are considering moves into digital security and forensics.

The problems that CS faces are encapsulated by opinions such as,

The dangers for Computer Science are acute ... for the first time in its history Computer Science is not expanding; other disciplines are encroaching on its territory ... we are now seeing 18-year-old students with at least 10 years computing experience, who can write flawless code while holding a dozen simultaneous e-conversations and watching TV ... they find “hello world” uninteresting ... they are natural collaborators – wires, extrovert and hyper-social; they are digital natives whereas their teachers are digital immigrants. Will Compute Science survive the current turmoil?

Keith Mander, University of Sheffield

I am always shocked at the low levels of intellectual rigour that seem to apply to much of the teaching of Computer Science/software engineering. We are good at teaching techniques ... but hopeless at encouraging our students to take a sceptical and informed perspective

Mike Holcombe, University of Sheffield

Of course, other disciplines and areas of knowledge have been, and continue to be, affected by changes in knowledge and in applications, and in their relationship to the labour market. However, CS is a field that is particularly vulnerable to rapid development, and where applications are emerging at speed, and often outside the Academy. It thus presents a particularly interesting case of the more general challenge

---

to the Academy presented by rapid changes of knowledge. At the same time, there are practical reasons for a focus on CS, that connect to the 'real world' problems of teaching and learning and employment noted above. The research proposed here, therefore, seeks to combine enquiry that may address some problems and issues in the delivery of CS, with a deepened understanding of the relationship between changing knowledge and the particular circumstances of CS.
The Questions

The overarching research questions are:

- Does the knowledge production process of CS relate to problems of recruitment and retention in the field?
- Does a deeper understanding of the knowledge base of CS enable us to better address these problems?

These overarching questions will be pursued through the following specific questions:

- What is happening to knowledge production in the 'cutting edge' fields in the Academy?
  - How is this best understood?
  - What kind of knowledge field is CS?
- What beliefs and assumptions about knowledge do CS faculty hold?
  - What processes of teaching and learning follow from these?
  - How far have these changed in recent years?
  - If there is a change, is it rooted in the field or in other factors (for example government policy?)
- What beliefs and assumptions about knowledge do CS students hold?
  - Where do they acquire these assumptions from?
  - Are these congruent with their experiences as undergraduate CS students?
  - What processes do the utilise to acquire CS knowledge?
- Where and to what extent are their alignments and conflicts?
- What practical lessons can be drawn from this enquiry?
A Short Review of the Literature

There are two contributing literatures: that dealing with recent and current changes in knowledge and that dealing with CS in particular.

On the nature and causes of 'new knowledge', Gibbons et al ¹ have become central figures in the debate. They are not without their critics; some welcomed the opening up of such a debate⁵, others view it as flawed⁶ while many have since taken it on board as an accepted position⁴.⁵. However, Gibbons et al's discussions on 'transdisciplinarity'⁶ – a clumsy word – do appear to be directly relevant to the current problem of CS as an “undisciplined discipline”.

It is clear that this new knowledge and the forces that have created it are a matter of intense debate in some academic circles. Delanty⁸ is critical of attempts by futurists to foresee the trends and effects. He doubts whether things are as inevitable and simplistic as are sometimes painted, and whether the move towards mode 2 from mode 1 knowledge is a good thing. His discussion around such issues as ‘democratic capitalism’, ‘managerial revolutions’ and ‘technological citizenship’⁹ are examples of

⁶ Gibbons M et al. op cit p27ff.
⁸ Delanty G. (2001), Challenging Knowledge: the University in the Knowledge Society, Buckingham: SRHE.
⁹ Delanty G. op cit p113.
how CS may have to engage with its changing place in the trans-disciplinary academic milieu.

Berger and Luckmann\(^1\) discussed the creation of accepted understandings of what knowledge is to a grouping of people, how they define and work within it and how this creates their realities. Although the text is old, it is a continued source of thought and publication that should inform the study further. Nelson Goodman\(^2\) also talks of how groups of people linked by a common knowledge field construct their realities. More recently, Mats Alvesson\(^3,4\) has become a reference point in how the acceptance of knowledge helps form the culture in work groupings.

This change in knowledge and perception of computing and computers by younger people is visible in the popular press:

"... tech know-how decreases with age. The younger you are, the more you inherently know in that (CS) realm. It's true that teens easily incorporate digital gadgets in nearly every aspect of their lives."

"College students probably spend more time than ever in front of a computer, but fewer seem to want to make computers their life work."

This gap between the CS generations has even been noted by Larry Ellison, the founder of Oracle Corporation and possibly the second-richest person in the field after Bill Gates of Microsoft Corporation. Ellison noted that, "If I were 21 years old I probably

wouldn't go into computing. The computer industry is about to become boring. I'd go into genetic engineering."  

There is evidence that a major – if not the – issue is the nature of CS itself. It is perceived as being dull. Sometimes HE CS faculty blame others, such as the schools or corporate greed. Students perceive a gap between their expectations and the teaching and learning realities. This is also notable in several studies on perceived poor teaching styles applicable to CS. Recently, Draper and Chickering et al note lack of faculty–student engagement, and later Draper points a finger at overpacked curricula. In such an atmosphere of get-it-all-taught there are dangers of skimming over the field without allowing students time to digest their learning and reflect upon it, particularly when some aspects are notoriously hard – for example, programming – to teach.

CS departments have an increasing issue in coming to terms with a field that is stretching,

---

"A challenging aspect of the computer science curriculum is to present the ever growing and ever changing topics of computer science within the time constraints imposed by the program of study."

"Computer Science ... has been shaped by a rapidly growing body of knowledge.""2

"... Computer Science ... [is] an ever-changing field.""3

"... a field as rapidly changing as computer science.""4

"... the rapidly changing character of the field (CS).""5

Yet, the same places emphasise how faculty also wish to keep their bloating subject grounded in the inherited truths of the foundations of CS,

"... our students gain a solid foundation in science and mathematics ...""6

"Our curriculum is built on the fundamental paradigms of the discipline: theory, abstraction and design."

282
When taken with probable knowledge- and intention-gaps between CS faculty and students, the lack of interface between the two groups cannot help in allowing teachers and learners to agree on a common curricular intention in the field.

The changes in CS are evident. At UAD, CS teaching is intent on moving away from a traditionalist, lecture-and-laboratory style towards problem-based learning approaches in the first three years of teaching in all CS degrees; only the final, honours year is to be clearly involved with Mode 1 knowledge¹.

Is this running away from deeper knowledge also an acceptance of the difficulty in dealing with its changing nature towards applications where the understandings are located in the student's domain? Indeed, are the CS faculty abandoning both the mode 1 and mode 2 knowledge domains, as the former is no longer something to be easily grasped by students and the latter is already within the domain of the students? If so, does the field of CS continue to exist or are CS faculty become facilitators of the students within their (that is, the students') knowledge domain.

If there is a perception gap as to the nature of the knowledge-base in CS between faculty and students, there may also be a further pushing-down effect upon the field by the social gap between the middle-class faculty and the new socially-inclusive FE and HE world, particularly in Scotland. Loss of interest by students in their studies may also be socially founded².

The nature of life for faculty in an atmosphere of syllabus and curricular change in a university has been a matter of debate since the inception of CS as a distinct academic discipline. Back in 1964 Miles³ found himself torn between the two opposites of inertia and change in educational innovation. By 1974 Nisbet⁴ had raised issues of how curriculum innovation brings 'problems of extra work-load, loss of confidence, [and] confusion' to the teacher. In 2001 Fullan could claim to begin to understand,

¹ MacKinnon L. (l.mackinnon@abertay.ac.uk) (18th October, 2005) Knowledge – where’s it taught? e-mail memo from Head of School to all SCCT staff at UAD.
'the nature of problems and principles of success and failure'. I am hopeful that Fullan's confidence will be illustrated and illuminated by others and by the enquiry.

Whilst Miles, Fullan and Nisbet talked within a general educational discourse (perhaps leaning more towards school education) others have taken time to consider change in the university context. There is a range of books from SRHE that could be considered. Such texts as Schuller and Warner and Palfreyman contain papers that open up discussions on a wide range of HE issues, including, teaching and learning patterns (that may be particularly relevant to CS with its high drop-out rate and low student satisfaction ratings), funding (CS has been historically an expensive laboratory-based subject but now finds itself with absent students who often prefer working from their home or flat) and transformation (has CS missed this, the next academic trend boat?)

Innovation takes place in a locus and by people. Hannan and Silver stated that in trying to understand it you need to know where and why it occurs within specific institutions, organisations and systems. While much of this has been discussed above, it is, as they continue, people and institutions who innovate, and it is within institutions that the management of innovation takes place. They identify the existence of frameworks that control the innovation which are driven by funding, reputation and survival.

"The 1990s have in fact been alive with the sound of innovation. Amidst, and often because of, unwelcome or unfamiliar pressures in higher education, inducements to innovate in teaching and learning have often, paradoxically, been strong and the response widespread."

Toohey is less optimistic, seeing the pressures to teach more hours, cut costs and increase income acting detrimentally upon good teaching and good curriculum design.

There remains the fact that despite all its innovative new programmes and courses, CS has rapidly and recently gone into decline as an academic field and it has struggled to

---

4 Hannan A, Silver H. (2000), Innovating in Higher Education: Teaching, Learning and Institutional Cultures, Buckingham: SRHE.
5 Hannan A et al. op cit p2.
6 Toohey S. (1999), Designing Courses for Higher Education, Buckingham: SRHE.
keep up with the changes in the applied field. Again, it looks on the surface to be related to such issues as knowledge modes, depths and viewpoints, and changing and dissipating knowledges and knowledge–bases. In his seminal work on the nature of life in the academe, Becher delineated a similar understanding between 'hard and soft, pure and applied, knowledge'\(^1\) as being critical to understanding the life of the academic. It is noteworthy that both he and Agassi discuss – in a truly reflexive sense – how much of where this debate on new knowledge is located is in a cross–disciplinary area which is itself 'a little explored border zone'\(^2\) and 'a relatively new academic discipline'\(^3\).

Yet, although it is widely accepted that CS has become part of a "highly interdisciplinary"\(^4\) domain and Hayes talks of 'intellectual migrants wandering back and forth across many academic frontiers'\(^5\), this change in the nature of a subject is not itself a new occurred since, as he wryly continues, "At one time, the term physics had a very broad meaning, roughly synonymous with natural science ... everything from astronomy and cosmology to meteorology, chemistry, zoology and botany (but not stamp collecting)."

---

1 Becher T. (1989), Academic Tribes and Territories: Intellectual enquiry and the cultures of disciplines, Buckingham: SRHE.
3 Agassi J. (1995) op cit
Methodology Discussion

By methodology, I mean,

Research methodology is the attempt to validate the rationale behind the selected research design and provide justification of why it is appropriate in solving the selected research problem.

Wikipedia\(^1\)

I am present in the study as a participant and as an observer. This allows me to bring to bear my understanding of knowledge change in CS as student, practitioner, teacher and researcher over the past quarter century and hence my knowledge of "the symbolic world of those studied"\(^2\). As such there will always be a degree of subjectivism in the enquiry, which I intend to embrace through appropriate and directed use of qualitative enquiry techniques.

As Robson states, such a real-world study rarely fits into neat structures as it 'takes place at particular times in particular places with particular people.'\(^3\) In this case the people are some of my students and colleagues. As the issue under investigation relates to the knowledge that faculty and students bring to the teaching and learning of CS, and where there are possible congruences and dissonances, I need to acquire my primary data directly from the teachers and learners directly. As Bryman\(^4\) states, the researcher needs "to see through the other people's eyes to understand their point of view."

Further, my interface with these students and staff requires care in handling potentially difficult emotions\(^5\) and even possible panic\(^6\). As I have some understanding and practical experience of counselling, and I would wish to only improve the situation, I would prefer to gather information from such a sensitive situation using face-to-face

---

5 Whyte WF. (1984) op cit 104,121.  
techniques rather than by, such as, spammed-out closed or semi-closed questionnaires.

With an open field to gather data from, it appears important to pilot the data gathering exercises in order to get a feel for the scope of data, its parameters and also to ensure that only enough data is gathered as is necessary and possible to analyse.

There are three techniques proposed: focus groups, interviews and observation. The questions I put will be informed by background knowledge and readings on knowledge change in CS.

A focus group (or ‘group interview’ or ‘group discussion’) is “a facilitated group discussion in which ideas, opinions and experiences are shared. They provide a unique and interactive way to gather information and share views and to gather data on experiences, perspectives, insights and understandings on group situations”. Brown et al. state that they give rise “to insights and solutions that would not come about without them”, in particular they can be used to bring out aspects of social relationships and patterns of interaction which may be particularly notable in a field, CS, that is studio (i.e. team) based in practice.

Standardised open-ended interviews (or ‘unstructured interviews’) will be a second means of data gathering. Their purpose is, “to find what is on someone else’s mind.”

---

4 Patton MQ. (1990) op cit p17.
12 Patton MQ. (1980) op cit p278.
Patton continues that this is a useful technique for capturing such as personal perspectives, experiences, thoughts, knowledge, and expectations. Open questionnaires are suitable as the interviewees are involved in the situation being investigated, have their own meanings of the situation and are affected by it.¹

Both the focus groups and open-ended interviews may need to be repeated² with groups and individuals if necessary. Whyte³ further talks of how interviewing takes place in stages, where issues of suspicion, unburdening and scoping arise at first, allowing the researcher to return and ask reflective and more focussed and informed questions later. I also hope that key individuals will arise from the focus groups who are perceptive, reflective and have significant experiences as the focus group does not easily allow identification of individual views.⁴

As stated above, I am bound to be a participant-observer⁵ in both the local and wider issue of how the CS Academy deals with knowledge-change. Inevitably there will be informal opportunities to gather data, consider, reflect upon and seek the reflections of others.

I propose to perform data analyses and feedback to those involved as the study proceeds. But, with appropriate care as issues may arise of censorship, covert activities⁶ and generally 'uncomfortable' conclusions.

The Method

The study will be a case study approach realised by data gathered by focus groups, open-ended interviews and participant observation for qualitative analysis with reference to issues of knowledge change in the field of CS and in the Academy. It will involve separate staff and students in the Computing Department at the University of ****. The focus groups will be small, around 4–6 people in each, and separate for staff and for students.

It is not intended to be overly prescriptive about schedules and deadlines, in order to allow the analyses to flow from the packets of data gathered. It may be that the study's form changes as significant indicators begin to appear that then require a change of tack.

Professor **** ******, the Head of Department, has specified that the focus group on students should start immediately (indeed, the first was in December, 2005.) These will be held in a comfortable, relaxed situation (probably the ***** meeting centre at *****) and recorded on audio tape. Questions will be initially set by the issues that are pre-identified in relation to knowledge change in CS. I intend to be advised by such as Robson1, Maxwell2, Yin3, Marshall and Rossman4, and, Denzin and Lincoln5.

Yin talks about how difficult analysis of data in case studies can be, how quickly it can become an overwhelming volume, and how disadvantaged a novice in the field can be6, so I intend to pilot (once each) the interviews and focus groups. For the pilot studies I have been particularly advised by Punch7, Denscombe8,1 and Gomm et al2.

3 Yin RK (1994) op cit.
6 Yin RK. (1994) op cit p102ff.
It may be that the focus groups will highlight individuals who may be suitable to interview. Robson\textsuperscript{3} discusses how real-life sampling for a case study provides criteria and advice and Gibbs\textsuperscript{4} provides a neat introduction to the approach with references to the work of other authors which may come in useful.

The analysis of the data gathered will be informed by such as Miles & Huberman\textsuperscript{5}, Silverman\textsuperscript{6} and Patton\textsuperscript{7} and will be related to the readings on knowledge change in CS.

I am also committed to ensuring the validity of the study by cross-checking the work in several ways:

1. The data gathering and methods are piloted to ensure maximum design correctness
2. the questions are informed by the CS and knowledge background analyses
3. data is gathered by both focus groups and questionnaires
4. data is gathered from both faculty and students
5. results are related back to the background analyses
6. the results are verified against faculty and students

---

\textsuperscript{4} Gibbs A. (1997) op cit.
The Draft Plan

<table>
<thead>
<tr>
<th>Year</th>
<th>Month</th>
<th>Activity 1</th>
<th>Activity 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005</td>
<td>Jan</td>
<td>Pilot study on students</td>
<td>Literature search on knowledge, knowledge societies and the changing nature of CS knowledge</td>
</tr>
<tr>
<td></td>
<td>Feb</td>
<td>Design of student focus group</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mar</td>
<td>Focus group on students</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Apr</td>
<td>Analysis of student focus group</td>
<td></td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>Pilot study on faculty</td>
<td></td>
</tr>
<tr>
<td></td>
<td>June</td>
<td>Analysis of pilot study</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Aug</td>
<td>Design of faculty focus group</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sept</td>
<td>Focus group on faculty</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Oct</td>
<td>Analysis of focus group</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Nov</td>
<td>Design of faculty questionnaires</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dec</td>
<td>Faculty questionnaires</td>
<td></td>
</tr>
<tr>
<td>2007</td>
<td>Jan</td>
<td>Analysis of questionnaires</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Apr</td>
<td>Design of student questionnaires</td>
<td></td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>Student questionnaires</td>
<td></td>
</tr>
<tr>
<td></td>
<td>June</td>
<td>Analysis of questionnaires</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sept</td>
<td>Feedback</td>
<td></td>
</tr>
<tr>
<td>2008</td>
<td>Jan</td>
<td>Final analyses</td>
<td></td>
</tr>
</tbody>
</table>

Draft literature review chapter: change in CS knowledge
Draft lit. rev. chapter: change in knowledge
Draft lit. rev. chapter: knowledge societies
Draft methodology chapter
Draft method chapter
Draft results chapter
Draft discussions chapter
Draft future work chapter
Finalise dissertation
Appendix B

Source Notes for Secondary Data Gathering on Undergraduate Computer Science Studies at Scottish Universities

Containing extracts from:-

* copies of notes from my log book taken at time of reading prospectuses at the universities
* my mapping of CS studies at university from prospectus information
* UCAS data on applications to study at universities
* UCAS data on acceptances to study at universities
26th July 1956

1. Project: The Glasgow Buddha
   - City of Asia (1958-59)
   - 3-4 years
   - 53, 54, 55, 56, 57, 58, 59, 60

2. Project: 1977-78 (Glasgow)
   - 53rd year
   - 54, 55, 56, 57, 58, 59, 60

3. Project: 1972-73 (Glasgow)
   - 53rd year
   - 54, 55, 56, 57, 58, 59, 60

4. Project: 1976-77 (Glasgow)
   - 53rd year
   - 54, 55, 56, 57, 58, 59, 60

5. Project: 1975-76 (Glasgow)
   - 53rd year
   - 54, 55, 56, 57, 58, 59, 60

6. Project: 1974-75 (Glasgow)
   - 53rd year
   - 54, 55, 56, 57, 58, 59, 60

7. Project: 1973-74 (Glasgow)
   - 53rd year
   - 54, 55, 56, 57, 58, 59, 60

8. Project: 1972-73 (Glasgow)
   - 53rd year
   - 54, 55, 56, 57, 58, 59, 60

9. Project: 1971-72 (Glasgow)
   - 53rd year
   - 54, 55, 56, 57, 58, 59, 60

10. Project: 1970-71 (Glasgow)
    - 53rd year
    - 54, 55, 56, 57, 58, 59, 60

The Queen's College, Glasgow, July 26th 1956

Glasgow, 13th July 1956

The Queen's College, Glasgow
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | AA | AB |
| 1 | 55-74 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 |
| 2 | Dough School |
| 3 |  |
| 4 | |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 | Glasgow Caledonian University |
| 14 |  |
| 15 |  |
| 16 | Glasgow College of Technology |
| 17 |  |
| 18 |  |
| 19 |  |
| 20 |  |
| 21 |  |
| 22 |  |
| 23 |  |
| 24 |  |
| 25 |  |
| 26 |  |
| 27 |  |
| 28 |  |
| 29 |  |
| 30 |  |
| 31 |  |
| 32 |  |
| 33 |  |
| 34 |  |
| 35 |  |
| 36 |  |

- Educational Technology Unit
- Course Lectures in Computing
- Computing lectures
- Computing for general IT
- All microcomputers
- Communications studio added
- Computing material in Hotel, Catering & Accommodation Management
- Computing in Home Economics
- Computer applications in Retail management
- BA Inter
- BA IT
- BSc
- BSc
- BSc
- BSc
- BSc
- BSc (Hons) CS
- BSc (Hons) Computer Sci
- BSc Information and OIC
- BSc Frame
- BSc Chemistry with Information Technology and In
<table>
<thead>
<tr>
<th></th>
<th>AC</th>
<th>AD</th>
<th>AE</th>
<th>AF</th>
<th>AG</th>
<th>AH</th>
<th>AI</th>
<th>AJ</th>
<th>AK</th>
<th>AL</th>
<th>AN</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>98</td>
<td>99</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>IS Development</td>
<td>IS Development</td>
<td>IS Development</td>
<td>IS Development</td>
<td>IS Development</td>
<td>IS Development</td>
<td>IS Development</td>
<td>IS Development</td>
<td>IS Development</td>
<td>IS Development</td>
<td>IS Development</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- e-Business:
  - MBA, MIS, IT5, IS
- IS Development:
  - e-b => electronic b
  - IS (IT) IS
- Computing & IT (no details)
- BSc Computing
- Games S/w dev
- BSc Computational Mathematics
- Maths & Comp
- Instrumentation
<table>
<thead>
<tr>
<th>Joint Academic Coding System</th>
<th>Degree (07)</th>
<th>Percent change</th>
<th>PDN (04)</th>
<th>Percent change</th>
<th>IBD (97)</th>
<th>Percent change</th>
<th>PDN (97)</th>
<th>Percent change</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME - Law By Area</td>
<td>64.149</td>
<td>4.9%</td>
<td>80</td>
<td>45</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>CS - Philosophy</td>
<td>77.413</td>
<td>3.9%</td>
<td>50</td>
<td>33</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>A1 - Pre-clinical Medicine</td>
<td>72.596</td>
<td>5.8%</td>
<td>50</td>
<td>33</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>A6 - Management system</td>
<td>62.189</td>
<td>11.1%</td>
<td>50</td>
<td>33</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>Q2 - English studies</td>
<td>53.861</td>
<td>7.6%</td>
<td>50</td>
<td>33</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>R7 - Nursing</td>
<td>53.177</td>
<td>7.0%</td>
<td>50</td>
<td>33</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>X1 - Teaching Teachers</td>
<td>41.798</td>
<td>10.6%</td>
<td>50</td>
<td>33</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>I7 - Business studies</td>
<td>48.510</td>
<td>5.5%</td>
<td>50</td>
<td>33</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>L3 - Social Work</td>
<td>48.541</td>
<td>16.1%</td>
<td>80</td>
<td>65</td>
<td>50</td>
<td>35</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>X7 - History of Physical</td>
<td>49.078</td>
<td>1.9%</td>
<td>50</td>
<td>33</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>L4 - Economics</td>
<td>49.080</td>
<td>3.0%</td>
<td>70</td>
<td>55</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>C6 - Science and Technology</td>
<td>43.200</td>
<td>10.0%</td>
<td>50</td>
<td>33</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>I9 - Computer Science</td>
<td>41.107</td>
<td>4.6%</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>M4 - Combined Science</td>
<td>40.694</td>
<td>0.6%</td>
<td>70</td>
<td>55</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>M2 - Combined Science</td>
<td>34.371</td>
<td>4.7%</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>L2 - Economics</td>
<td>37.322</td>
<td>10.6%</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>M1 - Design studies</td>
<td>35.140</td>
<td>18.7%</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>G1 - Mathematics</td>
<td>35.120</td>
<td>10.0%</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>B9 - Experts in Subjects out</td>
<td>31.813</td>
<td>20.5%</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>M3 - Economics</td>
<td>30.325</td>
<td>20.2%</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>M8 - Accounting</td>
<td>29.127</td>
<td>7.7%</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>B2 - Pharmacology, Physiology, and Pharmacy</td>
<td>26.451</td>
<td>9.4%</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>I8 - Economics</td>
<td>25.389</td>
<td>13.1%</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>X9 - Combined engineering and social science</td>
<td>24.036</td>
<td>4.2%</td>
<td>50</td>
<td>33</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>B1 - Anatomy and Physiology</td>
<td>25.383</td>
<td>13.1%</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>M7 - Materials science</td>
<td>25.685</td>
<td>3.4%</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>G4 - Management system</td>
<td>23.415</td>
<td>6.4%</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>E1 - Biological Engineering</td>
<td>33.561</td>
<td>6.6%</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>B9 - Experts in Subjects out</td>
<td>32.792</td>
<td>5.5%</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>M8 - Accounting</td>
<td>31.535</td>
<td>10.3%</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>M9 - Materials science</td>
<td>30.030</td>
<td>9.6%</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>X9 - Combined engineering and social science</td>
<td>29.000</td>
<td>8.2%</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>L1 - General Engineering</td>
<td>28.000</td>
<td>7.6%</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>M10 - Combined Subjects</td>
<td>25.389</td>
<td>13.1%</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
</tbody>
</table>

---
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>104</th>
<th>105</th>
<th>106</th>
<th>107</th>
<th>108</th>
<th>109</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>Nutrition</td>
<td>5.864</td>
<td>10.8%</td>
<td>100</td>
<td>106</td>
<td>82</td>
<td>92</td>
</tr>
<tr>
<td>B1</td>
<td>Nutrition</td>
<td>5.831</td>
<td>15.5%</td>
<td>69</td>
<td>52</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>C1</td>
<td>Combs of animals with</td>
<td>5.754</td>
<td>2.1%</td>
<td>89</td>
<td>72</td>
<td>52</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>engineering/technology</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1</td>
<td>Theology and Religious Studies</td>
<td>6.050</td>
<td>7.6%</td>
<td>76</td>
<td>76</td>
<td>76</td>
<td>76</td>
</tr>
<tr>
<td>E1</td>
<td>Combinatorial Mathematics</td>
<td>5.926</td>
<td>7.2%</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
</tr>
<tr>
<td>F1</td>
<td>Classical Literature</td>
<td>6.045</td>
<td>20.3%</td>
<td>34</td>
<td>34</td>
<td>34</td>
<td>34</td>
</tr>
<tr>
<td>G1</td>
<td>Classical Literature</td>
<td>6.045</td>
<td>20.3%</td>
<td>34</td>
<td>34</td>
<td>34</td>
<td>34</td>
</tr>
<tr>
<td>H1</td>
<td>Combs of medical sciences</td>
<td>5.467</td>
<td>3.2%</td>
<td>147</td>
<td>147</td>
<td>147</td>
<td>147</td>
</tr>
<tr>
<td>I1</td>
<td>Combs of medical sciences</td>
<td>5.467</td>
<td>3.2%</td>
<td>147</td>
<td>147</td>
<td>147</td>
<td>147</td>
</tr>
<tr>
<td>J1</td>
<td>Combs of medical sciences</td>
<td>5.467</td>
<td>3.2%</td>
<td>147</td>
<td>147</td>
<td>147</td>
<td>147</td>
</tr>
<tr>
<td>21</td>
<td>Other Sciences</td>
<td>5.663</td>
<td>15.1%</td>
<td>496</td>
<td>500</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>C2</td>
<td>Combs of medical sciences</td>
<td>5.467</td>
<td>3.2%</td>
<td>147</td>
<td>147</td>
<td>147</td>
<td>147</td>
</tr>
<tr>
<td>C3</td>
<td>Combs of medical sciences</td>
<td>5.467</td>
<td>3.2%</td>
<td>147</td>
<td>147</td>
<td>147</td>
<td>147</td>
</tr>
<tr>
<td>C4</td>
<td>Combs of medical sciences</td>
<td>5.467</td>
<td>3.2%</td>
<td>147</td>
<td>147</td>
<td>147</td>
<td>147</td>
</tr>
<tr>
<td>C5</td>
<td>Combs of medical sciences</td>
<td>5.467</td>
<td>3.2%</td>
<td>147</td>
<td>147</td>
<td>147</td>
<td>147</td>
</tr>
<tr>
<td>D2</td>
<td>European Languages, etc.</td>
<td>1.068</td>
<td>9.2%</td>
<td>54</td>
<td>54</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>E2</td>
<td>European Languages, etc.</td>
<td>1.068</td>
<td>9.2%</td>
<td>54</td>
<td>54</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>F2</td>
<td>European Languages, etc.</td>
<td>1.068</td>
<td>9.2%</td>
<td>54</td>
<td>54</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>G2</td>
<td>European Languages, etc.</td>
<td>1.068</td>
<td>9.2%</td>
<td>54</td>
<td>54</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>H2</td>
<td>European Languages, etc.</td>
<td>1.068</td>
<td>9.2%</td>
<td>54</td>
<td>54</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>I2</td>
<td>European Languages, etc.</td>
<td>1.068</td>
<td>9.2%</td>
<td>54</td>
<td>54</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>J2</td>
<td>European Languages, etc.</td>
<td>1.068</td>
<td>9.2%</td>
<td>54</td>
<td>54</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>Joint Academic Coding System (JACS) Subject</td>
<td>Degree 2008</td>
<td>Degree 2009</td>
<td>Percent change</td>
<td>FDG 2008</td>
<td>FDG 2009</td>
<td>Percent change</td>
<td>HND 2008</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------</td>
<td>------------</td>
<td>----------------</td>
<td>---------</td>
<td>---------</td>
<td>----------------</td>
<td>---------</td>
</tr>
<tr>
<td>MI - Law by Area</td>
<td>14.952</td>
<td>15.252</td>
<td>1.9%</td>
<td>31</td>
<td>35</td>
<td>6%</td>
<td>22</td>
</tr>
<tr>
<td>02 - Design studies</td>
<td>13.901</td>
<td>13.235</td>
<td>-5%</td>
<td>1.144</td>
<td>1.644</td>
<td>44%</td>
<td>1.333</td>
</tr>
<tr>
<td>CR - Psychology</td>
<td>13.003</td>
<td>12.702</td>
<td>-2%</td>
<td>1.15</td>
<td>1.15</td>
<td></td>
<td>1.15</td>
</tr>
<tr>
<td>PG - Management studies</td>
<td>11.358</td>
<td>11.872</td>
<td>-4%</td>
<td>1.254</td>
<td>1.044</td>
<td>-16%</td>
<td>1.372</td>
</tr>
<tr>
<td>MI - Business studies</td>
<td>10.094</td>
<td>10.948</td>
<td>-8%</td>
<td>1.125</td>
<td>1.203</td>
<td>7%</td>
<td>1.253</td>
</tr>
<tr>
<td>GA - Computer Science</td>
<td>9.387</td>
<td>10.709</td>
<td>13%</td>
<td>1.207</td>
<td>1.207</td>
<td></td>
<td>1.290</td>
</tr>
<tr>
<td>G3 - English studies</td>
<td>8.853</td>
<td>9.212</td>
<td>4%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
</tr>
<tr>
<td>A1 - Pre-clinical Medicine</td>
<td>8.011</td>
<td>7.922</td>
<td>-1%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
</tr>
<tr>
<td>LS - Social Work</td>
<td>7.795</td>
<td>7.393</td>
<td>-5%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
</tr>
<tr>
<td>FM - Sports Science</td>
<td>7.759</td>
<td>7.859</td>
<td>1%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
</tr>
<tr>
<td>KI - Training Teachers</td>
<td>7.425</td>
<td>7.826</td>
<td>5%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
</tr>
<tr>
<td>Y - Courses of social studies/other</td>
<td>7.214</td>
<td>7.680</td>
<td>6%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
</tr>
<tr>
<td>VI - General by Period</td>
<td>7.244</td>
<td>7.436</td>
<td>2%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
</tr>
<tr>
<td>N9 - Others in Fields</td>
<td>7.156</td>
<td>6.139</td>
<td>-16%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
</tr>
<tr>
<td>RN - Combinations within Business &amp; Arts</td>
<td>6.664</td>
<td>6.632</td>
<td>-0%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
</tr>
<tr>
<td>Studies</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y - Courses of languages</td>
<td>6.626</td>
<td>7.291</td>
<td>9%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
</tr>
<tr>
<td>RT - Law</td>
<td>6.396</td>
<td>5.271</td>
<td>-18%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
</tr>
<tr>
<td>WR - Accounting</td>
<td>5.927</td>
<td>5.570</td>
<td>-5%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Joint Academic Coding System (JACS) Subject</th>
<th>Degree 2008</th>
<th>Degree 2009</th>
<th>Percent change</th>
<th>FDG 2008</th>
<th>FDG 2009</th>
<th>Percent change</th>
<th>HND 2008</th>
<th>HND 2009</th>
<th>Percent change</th>
</tr>
</thead>
<tbody>
<tr>
<td>MI - Law by Area</td>
<td>14.952</td>
<td>15.252</td>
<td>1.9%</td>
<td>31</td>
<td>35</td>
<td>6%</td>
<td>22</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>02 - Design studies</td>
<td>13.901</td>
<td>13.235</td>
<td>-5%</td>
<td>1.144</td>
<td>1.644</td>
<td>44%</td>
<td>1.333</td>
<td>1.466</td>
<td>9%</td>
</tr>
<tr>
<td>CR - Psychology</td>
<td>13.003</td>
<td>12.702</td>
<td>-2%</td>
<td>1.15</td>
<td>1.15</td>
<td></td>
<td>1.15</td>
<td>1.15</td>
<td></td>
</tr>
<tr>
<td>PG - Management studies</td>
<td>11.358</td>
<td>11.872</td>
<td>-4%</td>
<td>1.254</td>
<td>1.044</td>
<td>-16%</td>
<td>1.372</td>
<td>1.682</td>
<td>23%</td>
</tr>
<tr>
<td>MI - Business studies</td>
<td>10.094</td>
<td>10.948</td>
<td>-8%</td>
<td>1.125</td>
<td>1.203</td>
<td>7%</td>
<td>1.253</td>
<td>1.308</td>
<td>4%</td>
</tr>
<tr>
<td>GA - Computer Science</td>
<td>9.387</td>
<td>10.709</td>
<td>13%</td>
<td>1.207</td>
<td>1.207</td>
<td></td>
<td>1.290</td>
<td>1.218</td>
<td>-5%</td>
</tr>
<tr>
<td>G3 - English studies</td>
<td>8.853</td>
<td>9.212</td>
<td>4%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>A1 - Pre-clinical Medicine</td>
<td>8.011</td>
<td>7.922</td>
<td>-1%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>LS - Social Work</td>
<td>7.795</td>
<td>7.393</td>
<td>-5%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>FM - Sports Science</td>
<td>7.759</td>
<td>7.859</td>
<td>1%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>KI - Training Teachers</td>
<td>7.425</td>
<td>7.826</td>
<td>5%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Y - Courses of social studies/other</td>
<td>7.214</td>
<td>7.680</td>
<td>6%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>VI - General by Period</td>
<td>7.244</td>
<td>7.436</td>
<td>2%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>N9 - Others in Fields</td>
<td>7.156</td>
<td>6.139</td>
<td>-16%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>RN - Combinations within Business &amp; Arts</td>
<td>6.664</td>
<td>6.632</td>
<td>-0%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Studies</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y - Courses of languages</td>
<td>6.626</td>
<td>7.291</td>
<td>9%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>RT - Law</td>
<td>6.396</td>
<td>5.271</td>
<td>-18%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>WR - Accounting</td>
<td>5.927</td>
<td>5.570</td>
<td>-5%</td>
<td>9.5</td>
<td>11</td>
<td></td>
<td>9.5</td>
<td>11</td>
<td></td>
</tr>
</tbody>
</table>

Table 9 - Acceptances by subject
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Y Comps of Humanities</td>
<td>1.995</td>
<td>3.712</td>
<td>-47%</td>
<td>135</td>
<td>54</td>
<td>-60%</td>
<td>5.0</td>
<td>85</td>
<td>-40%</td>
</tr>
<tr>
<td>B1 - Anatomy, Physiology and Pathology</td>
<td>3.192</td>
<td>3.719</td>
<td>-20%</td>
<td>99</td>
<td>93</td>
<td>-6%</td>
<td>29</td>
<td>31</td>
<td>-7%</td>
</tr>
<tr>
<td>F1 - Chemistry</td>
<td>3.581</td>
<td>4.505</td>
<td>+23%</td>
<td>109</td>
<td>109</td>
<td>0%</td>
<td>14</td>
<td>14</td>
<td>0%</td>
</tr>
<tr>
<td>FR - Physical &amp; Forensic</td>
<td>3.192</td>
<td>3.719</td>
<td>-20%</td>
<td>99</td>
<td>93</td>
<td>-6%</td>
<td>30</td>
<td>31</td>
<td>-7%</td>
</tr>
<tr>
<td>HS - Civil Engineering</td>
<td>5.411</td>
<td>3.719</td>
<td>-34%</td>
<td>41</td>
<td>54</td>
<td>+32%</td>
<td>105</td>
<td>109</td>
<td>+4%</td>
</tr>
<tr>
<td>Y Comps of physical science with social</td>
<td>5.059</td>
<td>3.719</td>
<td>-20%</td>
<td>72</td>
<td>115</td>
<td>+60%</td>
<td>44</td>
<td>60</td>
<td>+40%</td>
</tr>
<tr>
<td>Studies/Sciences</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G5 - Information Systems</td>
<td>3.082</td>
<td>3.405</td>
<td>+10%</td>
<td>128</td>
<td>124</td>
<td>-3%</td>
<td>10</td>
<td>10</td>
<td>0%</td>
</tr>
<tr>
<td>HS - Marketing</td>
<td>5.349</td>
<td>3.719</td>
<td>-34%</td>
<td>117</td>
<td>143</td>
<td>+21%</td>
<td>486</td>
<td>357</td>
<td>-27%</td>
</tr>
<tr>
<td>F5 - Physics</td>
<td>2.929</td>
<td>2.917</td>
<td>-0.4%</td>
<td>35</td>
<td>35</td>
<td>0%</td>
<td>10</td>
<td>10</td>
<td>0%</td>
</tr>
<tr>
<td>K2 - Building</td>
<td>2.849</td>
<td>3.022</td>
<td>+5.7%</td>
<td>172</td>
<td>121</td>
<td>-26%</td>
<td>456</td>
<td>337</td>
<td>-26%</td>
</tr>
<tr>
<td>H1 - General Engineering</td>
<td>2.812</td>
<td>2.814</td>
<td>0.1%</td>
<td>51</td>
<td>53</td>
<td>4%</td>
<td>166</td>
<td>114</td>
<td>-27%</td>
</tr>
<tr>
<td>Y Comps of soc studies/Sciences with business</td>
<td>2.799</td>
<td>3.081</td>
<td>+9.8%</td>
<td>24</td>
<td>26</td>
<td>8%</td>
<td>16</td>
<td>20</td>
<td>+25%</td>
</tr>
<tr>
<td>H8 - Tourism, Transport and Travel</td>
<td>2.712</td>
<td>2.743</td>
<td>+1.3%</td>
<td>355</td>
<td>342</td>
<td>-4%</td>
<td>450</td>
<td>502</td>
<td>+11%</td>
</tr>
<tr>
<td>Y Comps of social studies/Sciences with</td>
<td>2.717</td>
<td>2.808</td>
<td>+3.4%</td>
<td>293</td>
<td>136</td>
<td>-53%</td>
<td>96</td>
<td>68</td>
<td>-27%</td>
</tr>
<tr>
<td>L1 - Human and Social Geography</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L2 - Combinations within Social Studies</td>
<td>2.644</td>
<td>2.594</td>
<td>-1.8%</td>
<td>15</td>
<td>15</td>
<td>0%</td>
<td>1</td>
<td>1</td>
<td>0%</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Y Comps of physical sciences with</td>
<td>2.290</td>
<td>2.435</td>
<td>+6%</td>
<td>105</td>
<td>109</td>
<td>+4%</td>
<td>105</td>
<td>115</td>
<td>+10%</td>
</tr>
<tr>
<td>art/humanities</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y Comps of physical sciences</td>
<td>2.290</td>
<td>2.435</td>
<td>+6%</td>
<td>105</td>
<td>109</td>
<td>+4%</td>
<td>105</td>
<td>115</td>
<td>+10%</td>
</tr>
<tr>
<td>Y Comps of physical sciences with</td>
<td>2.290</td>
<td>2.435</td>
<td>+6%</td>
<td>105</td>
<td>109</td>
<td>+4%</td>
<td>105</td>
<td>115</td>
<td>+10%</td>
</tr>
<tr>
<td>art/humanities</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y Comps of physical sciences</td>
<td>2.290</td>
<td>2.435</td>
<td>+6%</td>
<td>105</td>
<td>109</td>
<td>+4%</td>
<td>105</td>
<td>115</td>
<td>+10%</td>
</tr>
<tr>
<td>Y Comps of physical sciences with</td>
<td>2.290</td>
<td>2.435</td>
<td>+6%</td>
<td>105</td>
<td>109</td>
<td>+4%</td>
<td>105</td>
<td>115</td>
<td>+10%</td>
</tr>
<tr>
<td>art/humanities</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y Comps of physical sciences</td>
<td>2.290</td>
<td>2.435</td>
<td>+6%</td>
<td>105</td>
<td>109</td>
<td>+4%</td>
<td>105</td>
<td>115</td>
<td>+10%</td>
</tr>
<tr>
<td>Y Comps of physical sciences with</td>
<td>2.290</td>
<td>2.435</td>
<td>+6%</td>
<td>105</td>
<td>109</td>
<td>+4%</td>
<td>105</td>
<td>115</td>
<td>+10%</td>
</tr>
<tr>
<td>art/humanities</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

UCAS
### Table 2e - All HE students by level of study, mode of study, subject of study(#6), domicile and gender 2004/05

In order to clarify the terms used in this table, please read the Definitions.

#### Downloadable: subjectHE.xls

<table>
<thead>
<tr>
<th>Subject/Mode/Location</th>
<th>Total HE students</th>
<th>United Kingdom</th>
<th>Other European Union</th>
<th>Non-European-Union</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PTUGs</td>
<td>FTUGs</td>
<td>PTPGs</td>
<td>FTUGs</td>
</tr>
<tr>
<td>Medicine &amp; dentistry</td>
<td>55960</td>
<td>38395</td>
<td>8160</td>
<td>180</td>
</tr>
<tr>
<td>Pre-clinical medicine</td>
<td>14490</td>
<td>13610</td>
<td>245</td>
<td>30</td>
</tr>
<tr>
<td>Pre-clinical dentistry</td>
<td>1190</td>
<td>1110</td>
<td>25</td>
<td>20</td>
</tr>
<tr>
<td>Clinical medicine</td>
<td>34340</td>
<td>19705</td>
<td>5160</td>
<td>86</td>
</tr>
<tr>
<td>Clinical dentistry</td>
<td>5280</td>
<td>3680</td>
<td>520</td>
<td>10</td>
</tr>
<tr>
<td>Others in medicine &amp; dentistry</td>
<td>690</td>
<td>90</td>
<td>210</td>
<td>40</td>
</tr>
<tr>
<td>Subjects allied to medicine</td>
<td>300140</td>
<td>148290</td>
<td>8335</td>
<td>11035</td>
</tr>
<tr>
<td>Anatomy physiology &amp; pathology</td>
<td>1165</td>
<td>705</td>
<td>0</td>
<td>460</td>
</tr>
<tr>
<td>Pharmacology &amp; pharmacy</td>
<td>16325</td>
<td>11945</td>
<td>1610</td>
<td>1385</td>
</tr>
<tr>
<td>Complementary medicine</td>
<td>17840</td>
<td>10945</td>
<td>1480</td>
<td>750</td>
</tr>
<tr>
<td>Nutrition</td>
<td>5685</td>
<td>3435</td>
<td>105</td>
<td>845</td>
</tr>
<tr>
<td>Optics</td>
<td>4580</td>
<td>2610</td>
<td>520</td>
<td>875</td>
</tr>
<tr>
<td>Aural &amp; oral sciences</td>
<td>3360</td>
<td>2210</td>
<td>490</td>
<td>230</td>
</tr>
<tr>
<td>Nursing</td>
<td>191425</td>
<td>90950</td>
<td>1245</td>
<td>90680</td>
</tr>
<tr>
<td>Medical technology</td>
<td>8540</td>
<td>5110</td>
<td>355</td>
<td>710</td>
</tr>
<tr>
<td>Others in subjects allied to medicine</td>
<td>47325</td>
<td>16960</td>
<td>2475</td>
<td>13880</td>
</tr>
<tr>
<td>Biological sciences</td>
<td>149520</td>
<td>100050</td>
<td>14235</td>
<td>21115</td>
</tr>
<tr>
<td>Botany</td>
<td>2820</td>
<td>1730</td>
<td>2805</td>
<td>3810</td>
</tr>
<tr>
<td>Zoology</td>
<td>845</td>
<td>305</td>
<td>330</td>
<td>200</td>
</tr>
</tbody>
</table>

[http://www.besa.ac.uk/education holland/universitysubjectHE.xls](http://www.besa.ac.uk/education holland/universitysubjectHE.xls)

Page 1 of 7
| Table 2c: All HE students by level of study, subject of study, domicile and gender 2004/05 |
|---|---|---|---|---|---|
| Mathematical sciences | 25655 | 17925 | 2100 | 3915 | 1615 |
| Mathematics | 21910 | 8275 | 13635 | 2660 | 1075 |
| Operational research | 965 | 190 | 315 | 326 | 140 |
| Statistics | 3445 | 1440 | 650 | 615 | 740 |
| Others in mathematical sciences | 15 | 15 | 0 | 0 | 0 |
| Others in mathematical & computing sciences | 1300 | 310 | 170 | 705 | 120 |
| Computer science | 131280 | 73515 | 13358 | 33265 | 10985 |
| Computer science | 108280 | 26495 | 81785 | 5515 | 1020 |
| Information systems | 81065 | 50620 | 6090 | 14335 | 5910 |
| Software engineering | 41440 | 17700 | 3495 | 16310 | 3940 |
| Artificial intelligence | 7220 | 4820 | 765 | 760 | 1070 |
| Others in computing sciences | 550 | 445 | 80 | 0 | 35 |
| Engineering & technology | 137825 | 75720 | 21265 | 22100 | 18480 |
| Broadly-based programme | 6355 | 3460 | 550 | 265 | 25 |
| Civil engineering | 21910 | 8275 | 13635 | 2660 | 1075 |
| Mechanical engineering | 19700 | 10545 | 2990 | 3520 | 2695 |
| Aerospace engineering | 14500 | 2175 | 2780 | 1760 |
| Naval architecture | 7600 | 5610 | 590 | 650 | 760 |
| Naval architecture | 245 | 370 | 60 | 10 | 40 |
| Producing & manufacturing engineering | 34500 | 19565 | 6645 | 4740 | 3835 |
| Production & manufacturing engineering | 2808 | 1985 | 18095 | 2935 | 290 |
| Chemical, process & energy engineering | 7980 | 3765 | 1685 | 1560 | 1050 |
| Chemical, process & energy engineering | 6080 | 3330 | 1445 | 170 | 1135 |
| Others in engineering | 120 | 120 | 50 | 50 | 50 |
| Minerals technology | 1520 | 355 | 420 | 485 | 280 |
| Minerals technology | 225 | 120 | 50 | 50 | 50 |
| Metallurgy | 665 | 110 | 390 | 5 | 160 |
| Ceramics & glass | 125 | 50 | 55 | 55 |
| Polymeric & textiles | 2405 | 1845 | 365 | 115 | 140 |
| Materials technology not otherwise specified | 3215 | 1690 | 865 | 165 | 480 |
| Materials technology | 1960 | 710 | 150 | 35 | 165 |

http://www.araa.ac.uk/office/patients/subjectnumber0005.htm
Appendix C

Exemplars of Requests and Replies for Data from
Scottish Universities on Undergraduate Computing Science
Applications.
### Applications to study Information Systems at Paisley

<table>
<thead>
<tr>
<th>Session</th>
<th>Year of entry 1</th>
<th>Year of entry 2</th>
<th>Year of entry 3</th>
<th>Year of entry 4</th>
<th>Year of entry 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1995-1996</td>
<td>0</td>
<td>26</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1996-1997</td>
<td>0</td>
<td>16</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1997-1998</td>
<td>24</td>
<td>15</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1998-1999</td>
<td>19</td>
<td>20</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1999-2000</td>
<td>13</td>
<td>22</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2000-2001</td>
<td>15</td>
<td>8</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2001-2002</td>
<td>19</td>
<td>11</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2002-2003</td>
<td>17</td>
<td>5</td>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2003-2004</td>
<td>13</td>
<td>5</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2004-2005</td>
<td>18</td>
<td>3</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2005-2006</td>
<td>13</td>
<td>3</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Applications to study Software Engineering at Paisley

<table>
<thead>
<tr>
<th>Session</th>
<th>Year of entry 1</th>
<th>Year of entry 2</th>
<th>Year of entry 3</th>
<th>Year of entry 4</th>
<th>Year of entry 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1995-1996</td>
<td>0</td>
<td>35</td>
<td>26</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1996-1997</td>
<td>0</td>
<td>18</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1997-1998</td>
<td>64</td>
<td>17</td>
<td>24</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1998-1999</td>
<td>62</td>
<td>41</td>
<td>24</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1999-2000</td>
<td>66</td>
<td>20</td>
<td>36</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2000-2001</td>
<td>65</td>
<td>32</td>
<td>37</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2001-2002</td>
<td>58</td>
<td>20</td>
<td>33</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2002-2003</td>
<td>55</td>
<td>25</td>
<td>41</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2003-2004</td>
<td>70</td>
<td>24</td>
<td>31</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2004-2005</td>
<td>50</td>
<td>35</td>
<td>37</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2005-2006</td>
<td>27</td>
<td>12</td>
<td>28</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

### Applications to study Computer Games Technology at Paisley

<table>
<thead>
<tr>
<th>Session</th>
<th>Year of entry 1</th>
<th>Year of entry 2</th>
<th>Year of entry 3</th>
<th>Year of entry 4</th>
<th>Year of entry 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001-2002</td>
<td>59</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2002-2003</td>
<td>287</td>
<td>35</td>
<td>31</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2003-2004</td>
<td>347</td>
<td>27</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2004-2005</td>
<td>229</td>
<td>15</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2005-2006</td>
<td>217</td>
<td>15</td>
<td>14</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
From: "PCI Strathclyde" <doi@strath.ac.uk>
Date: Mon Jul 24, 2006 11:47:59 AM Europe/London
To: "John N Sutherland" <akademos@biinternet.com>
Subject: RE: Computing Science

Dear John,

Thank you for your request (below) under the Freedom of Information (Scotland) Act 2002, regarding Computer Science data.

Data is available from various sources including:

http://www.hesa.ac.uk/products/home.htm
and
http://www.ucas.ac.uk/

The University statistics obtained from the Strathclyde University Student Record system is as follows:

Applications to the BSc Hons Computer Science and the MEng Computer Science (introduced in session 2002/03):

<table>
<thead>
<tr>
<th>Year</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>1995/96</td>
<td>546</td>
</tr>
<tr>
<td>1996/97</td>
<td>513</td>
</tr>
<tr>
<td>1997/98</td>
<td>543</td>
</tr>
<tr>
<td>1998/99</td>
<td>546</td>
</tr>
<tr>
<td>1999/00</td>
<td>631</td>
</tr>
<tr>
<td>2000/01</td>
<td>592</td>
</tr>
<tr>
<td>2001/02</td>
<td>612</td>
</tr>
<tr>
<td>2002/03</td>
<td>613</td>
</tr>
<tr>
<td>2003/04</td>
<td>419</td>
</tr>
<tr>
<td>2004/05</td>
<td>326</td>
</tr>
<tr>
<td>2005/06</td>
<td>313</td>
</tr>
</tbody>
</table>

Over the last ten years the University has also run additional courses within the department of Computer Science such as BSc Computer Science with Law, BSc Computer Science with a Modern Language, BSc Business and Software Development, BSc Software Engineering and BSc Business Information Systems. The department also run two courses jointly with the Engineering Faculty - BEng and MEng in Computer and Electronic Systems.

The statistics for these courses are not included in the above totals.

I hope this information is of help to you.

Yours sincerely,

Jeanette Leiper
Freedom of Information Office
University of Strathclyde

Copyright:
This information is supplied under the Freedom of Information (Scotland) Act 2002 and copyright in it belongs to the University of Strathclyde or to another party. Releasing this information to you under the Freedom of Information (Scotland) Act 2002 does not convey a right to reuse that information in a way that would infringe copyright. Copyright material must not be copied, distributed, modified, reproduced, transmitted, published (including published on the internet or intranet) or otherwise made available in whole or in part without the prior written consent of the copyright holder.
Appendix D

Exemplar Interview Transcript - Preamble

The transcript is one of the individual interviews which were the primary data in the analysis. All proper names have been changed to protect the individuals named, those being interviewed, and the places mentioned. In almost all cases a pretend name has been inserted (this is the same name for the same person or institution in each of the transcripts). In some cases a generic title has been used where the individual holds a rank and it is the rank that is being discussed. The real name has been retained where identifying the individual person, university, place or organisation does not affect the necessary anonymity or it is also necessary to identify the proper name to better understand the data.

The data in the transcripts can be reused by someone performing a secondary analysis, and can be made available on request in PDF, Pages or DOC format. After the passage of a suitable period of time, possibly after June 2018, the author can be approached for the raw data recordings.

The Universities re-named can be categorised as follows (this is pertinent to the discussions):

- pre-1960’s: A, B, G
- 1960’s: C, F, H, J
- 1990’s: D, E
People are named by colours. There is no significance to names chosen, these are purely arbitrary.

The naming of degrees or areas of study are as spoken and are not altered.

All words are transcribed as spoken. Inaudible parts of the recording are indicated by the double-symbol <>. Gaps in speech - silences - are indicated by three dots ‘...’.

Grammar is added as the author thought it should be implied. At the start and end of some of the interviews the text has not been transcribed as the speaking was not relevant to the study; this is indicated because it shows the complete content of the recordings and that what is recorded in the appendices is a true indication of the entire interviews.

The Interview Transcript – Dr ‘Blue’

August, 2006.

<preamble about the recording technology>
<preamble about the research>
W: you won’t be able to do this in five years time
B: no

W: they'll be away

B: they'll be away

W: how did you get into Computing then, Dr Blue?

B: <laughs> now there's a question. yes. Originally at school. We were one of the few schools who actually had a bit of Computing and Maths. <>filling in coding sheets. Sending them away to the college of education. Getting back the results the next week. And then at university, after a year, decided to change courses from theoretical physics to Computing. Cos I enjoyed it so much.

W: did you have old Jack Cole?

B: Professor Cole was my head of department at that time. Jack. Yeh. He was the head of department. Dr Blue-Green was still there.

W: he was a wee boy then wasn't he?

B: must've been. yeh.

W:<laughs>

B: so, actually, I don't know how long it had been running at the University of G, but we were certainly one of the first half a dozen cohorts who did Computational Science.

W: what year was that, then?

B: graduated 1977. Started 1972, graduated 1977. The school, actually I had two experiences of Computing before that. one was at school, as I've said, that would have been 70-71. And also I had a summer school at the University of F in 71 doing what they called Computing, but was really numerical analysis. Cos in those days Computing was basically Maths. You know, that's what you used Computing for. Doing numerical analysis problems and things like that.

W: how old were you in 1970?

B: 70, I would have been about 16.

W: you're a bit older than me. 12 back then. That's about when the roots of Computing begin to appear, as a subject.

B: yeh yeh.

W: I did Computing, I did it as a filler at the University of B. And didn't have a Computing degree. Hadn't started yet then. So, the University of G was quite quick off the mark.
B: Yes it was. of course it wasn’t called Computing; computational science. It had an aura of theoretical

W: yeh. I remember Dr Blue-Green used to get quite upset of you called it Computing science. Did you ever get a handle on what was the difference between computational and Computing

B: no. I was never actually so interested in the theory as I was in the practice. I never really pursued it that much. I mean, that was the last time I ever looked at computer science, per se, was during my degree. Basically, the degree was useless in terms of getting a job. You went out, I mean, it was useful in that it got you a job. When you actually went into industry, the Computing they were teaching, certainly in the University of G in those days, was miles away removed from reality in Computing. You almost had to relearn, your education, what you learned, how to learn and the art of learning, being at university, were what made you able to go into Computing in the real world. It wasn’t the actual content of what you’d learned at university. I didn’t even know what a file was when I left university.

W: what was in my mind there was, I remember leaving the University of B and starting at Rolls Royce as a Fortran programmer and I had to read a file and I had no idea what a file was.

B: no.

W: it was a shocker

B: and yet you got into industry, as a trainee programmer sort of thing, your first job, usually in those days, your first job would be to produce a program to produce a report. reading from a file, paginating it, you know, x lines per page, with columns, and maybe with subtotals and totals, and that was totally alien. I was actually in the middle of all this. Through a contact I actually started working in Timex in their data processing department, as a programmer, in 1975. So, before I’d finished my course. So on the one hand I was in the real world and they were saying on one sense quite rightly, what a load of rubbish you’re being taught at university, that’s of no use to anybody. Then you went into university and they were saying, how much rubbish the real world was and they weren’t doing Computing properly. You know, they shouldn’t be writing in assembler and drawing flowcharts. They should be doing this.

W: its funny, by definition, old Jack Cole and Dr Blue-Green didn’t come from a Computing background.

B: they were all engineers, physicists and mathematicians.

W: How did they define what Computing was back then, do you think? There must have been a way to say, that is Computing.

B: I think in the 70s, that era in the mid 70’s, there was work going in the American ACM in terms of defining a Computing curriculum. But, I think, when you look back at it, the course didn’t actually begin until 2nd year. In 1st year you did Maths. you did a double Maths module in which there was a little bit of Computing. Then Computing didn’t start
till 2nd year, and, I’m not actually sure, I couldn’t be absolutely sure about this, but I think you had to have done 1st year Maths in order to do 2nd year Computing.

W: until very recently that was <>

B: And then I think it was more a case of what the actual academic staff were interested in. What was actually taught. I mean, it would be interesting, for example, I had Dr Grey-Black who taught me a lot about the underlying ideas of Computing with his functional programming and a language called SASL. Which eventually became Miranda. When he moved down to Kent. And he also did some Maths and some denotational semantics as well. And I had always been keen to know what happened when Dr Grey-Black left and went to Kent. Did that subject matter continue, or did the replacement academic bring his favourite topic and replace it with that? I don’t know, but I always wondered about that.

W: it would be interesting to see in other degree, I remember in Maths at the University of B, which was the main thing I did out there, beautifully structured, ores structure, the notes were amazing that they gave. And people didn’t teach their favourite bits of Maths, they taught the syllabus all the way through. So whether Computing is a kinda gorgeous subject where because it started from nothing you could teach anything you felt like.

B: of course, being a student, in those days I wasn’t particularly interested. I don’t know if our students are particularly interested in it either, the syllabus. I mean, we learned what we were taught. But certainly, unlike nowadays, we were never handed out a syllabus, a curriculum, a course handbook, so you never actually knew whether there was a syllabus or whether every year they just decided what they were going to teach. I don’t think that is true, we certainly were, but I certainly never saw it. And I couldn’t see how everything fitted together. There had been thought put into it because everything did actually link together, and I would certainly say, while it didn’t bear any resemblance to the real world whatsoever, the theories it gave you, gave you a superb foundation for learning. And, in fact, a lot of what I learned as a student still survives today and I still use. I mean, one, an example in Timex, when I was working there full-time, after I graduated I went there on a full-time job, we were looking at a new programming language called PL/1. IBM had brought out. Now, in those days, travel was much more expensive and the training courses weren’t quite so frequent, so it wasn’t a case of going somewhere nice to learn how to PL/1 program, it was a question of bringing in this whole library of kinda video cassettes, goodness knows, they weren’t’ beta and they weren’t VHS, some weird learning machine video, and there was another couple of folk who came at the same time who had come from K College, I think, from College anyway, they had HND’s, and those were the days HND’s were quite valid technical level, but I certainly noticed the huge difference between their ability to learn that language and mine. You know, because I had all the basics, and I was used to learning. You know, I just zoomed though these tapes. You know, ba-ba-ba-ba-bang, dead quick. Where, Id say the two people who came from college with HND’s they were very much slower, very much more pedantic, they didn’t have a basic structure on which to build this new language. You know, it was like every programming language was a new language to them. <>learning everything from scratch again. Whereas I had been given the basic sort of ideas of blocks and variable, procedures and stuff like that. And remember, of course, this was also the time when in academia, it was very much structured programming. The only exception to that would have been when you had to write some kind of assembler.
Whereas, in the real world, it was still flowcharts and goto statements, even though you were using a structured language. I'll give you an example, this was about late 79 at this time, late 79. I'd decided to come to work at the then L College of Technology as a lecturer in Computing. So, I put my notice into Timex. And I'd been working on a big project that had kinda come to an end anyway. So, what was I going to do. So, they had this kinda one-off program that they were needing written in PL/1. You can knock that off in a week or two, you know. So, sure enough, it was just a week long program, it wasn't a big one at all. And I decided to do it going back to my university knowledge, going back, doing it properly, so I wrote it in PL/1, parameters, procedure calls, there wasn't one goto statement in the whole program, while loops, beautifully structure. And I thought, an academic couldn't have faulted it any way, between the looping, the parameter passing, all the rest of it. Compiled it, ran it, it worked perfectly. Fine, Cheerio, Thanks, Been great working for you. Came to work here. About a month later my old boss telephoned me and said, we're needing you program and we've got this same situation again, but your program is almost what we want, except, it's that badly written that none of us have a clue how it works. Now, that was his words: badly written. And yet I would still maintain it was beautifully written. But they were used to flowcharts, GOTO's, they didn't understand parameter passing, cos they'd never met it, you know. I sometimes wonder, like, that's one of the reasons I've been able to move. I think, the science, if you like, the underlying principles that I got in the 70's, I don't think have changed, until today. Except for object orientation, which we didn't, although, even that you did actually get a sniff of, because there was a language called Simula-67, which was object oriented and had virtual functions and everything in it. So we had a sniff of it there. But, everything else. I'd hate to see some of these programmers who were programming in the real world in assembler and in PL/1 with GOTO's going all over the place, how they coped with going into a language like even C, but certainly going up into the C++'s and the Java's, where you were forced to be structured. That must have been a huge conceptual jump for them.

W: In the University of B we did Algol-W.

B: as did we, as did we.

W: Came out into the horrors of the real world with Fortran IV, COBOL-66, RPG2. God, a nightmare, RPG2. And of course, if you feel like it, you might put a CALL or a PERFORM statement in <>GO TO's <> GOTO's all over the place. Then started doing the research masters at the University of G, and it was all in s-algor. By that time they'd moved onto s-algor. Its the only time it has happened to me. I sat down and, what I sued to do was write the programs on paper, and then I would type them in

B: punch cards, yeh.

W: put it in, this entire program, part of the suite of programs I was writing, compiled it, right, ran it, right. And I didn't have to change anything. And I was astonished. And, suddenly, I was in awe of Jack and Dr Blue-Green who had designed a language so clean that, if it compiled, it ran. And there was a good chance you could write it so clean and neat and tidy. Very compact language, so, I understand that. On the other hand, I remember teaching one year, do you remember that the Telecomms Inc. software engineers used to do the MSc part-time?

B: yeh
W: and I picked up Dr Mauve's module one year when he was on sabbatical on quality. And, standing chatting, we were talking about the horrors of using global variables, whatsoever, and they were all really restless and really upset, and I remember stopping dead and saying, well, what is it? And they said, that's just rubbish. That's rubbish. If you want the fastest, most efficient program, just use global variables and repeat the code as often as you want. They worked in a different world. There was a complete paradigm clash, if you like, between what I was saying to them and what they were saying to me.

B: I think, though, that they just didn't get it. Those people had been brought up in industry, because, one of the biggest differences between working in industry and here is that, when we knock a program together, we usually run it. And that's it. Whereas, there, you were writing a program that was run either daily, weekly or monthly, it was always a regular basis, and it always needed changing. And the more it had global variables and unstructured programming, the more difficult to maintain the program. And, of course, you always have to remember that in those days the computers were so slow that, sometimes, speed was an issue. But, you're right, global variables was a classic one where everybody included <> at the beginning of your program. I don't think I was ever in that kind of vein because it was the way I had been brought up with algol-w.

W: I found also that when I wrote in COBOL, this was back in machine <> you were overlaying stuff and ran out of memory.

B: There's a subject that you don't teach any more, overlaying.

W: no, actually, buffering I hadn't talked about for donkeys' years. Until the students started doing graphics in Java. And I suddenly realised they had to start buffering their display stuff which was flickering on the screen. So, what you were saying about underlying principles, it always comes back. So, the funny thing about buffering, OK, I'll tell you about buffering, this is what buffering is.

B: even though it was 20 years ago you;'d done it.

W: it hasn't changed

B: I think, it's a lot of that low-level, I think the high-level, the low level stuff, and you had to know both,. Whereas now, you don't have to know the low-level stuff. We used to. I think it depends. An anecdote. I think, I was never a database person. And we maybe had to look at a bit of databases, then latterly, OK, you know, every program has to speak to a database. But, you look up the curriculum, you look up the textbooks, and all you see is normalisation. 1st order form, 2nd, you know, all this stuff. And I was looking through it, and I was thinking it was this big deal. I thought, this is useless, this is irrelevant, this is silly, this is stupid, I don't understand this. But then I suddenly realised why I thought that. Because of the way that I'd been brought up with the principles, when I designed a database, it would be in normal form. Without thinking - just naturally. I would not duplicate data. So, all the rules that they applied for normalisation, were irrelevant to me because my data would be normalised, just by the way I'd been brought up. Thinking about it. But, someone who hasn't been brought up that way, yeh, you know, you have to go through that. And why I am able to move from one technology to another. I don't think they are. I mean, I look at the web nowadays,
and I see huge similarities. You talk to someone like Dr Lemon who is heavily into HTML. He’s talking about this happens and that happens with the browser. And that’s no different from what we were talking about in the 70’s between processes and the operating system. It has just moved to a different vehicle with a different syntax around. But, a lot of things they are talking about, the browsers, a lot of the event handling, its exactly the same. The problems you have with windows, when you start designing windowing systems.

W: I think that the only that has appeared that really challenged me and stopped me in my tracks was moving into kinda telecoms stuff. When suddenly electronics and physics begin to appear. <> lectures on satellite comms, and things. Suddenly you’re aware of the fact that, we’re into wavelengths here, data loss, we’re into data packets, medium wave and FM, we’re into reflections off the sea, we’re into, suddenly the real world began to appear. I must admit, I found it very difficult to hold my virtual Computing world and my real world in my head at the same time.

B: my comms were very practical comms that I do. You know, it was more like fault finding, going through a modem, going through a BT line. So, you had to understand phase and frequency modulations. but, you never actually taught it at that stage. I just used it and did it. It was more of a question of I never had. The biggest issue that I had, I’d say the only thing that my head really stuck at, because, functional programming I got through, at university, so you went on the things like ML, denotational semantics and formal methods, <> the big thing in the 80’s, formal methods, that wasn’t a big deal because that was just functional programming taken on a bit. What really blew my mind was object orientation. And I think I was very lucky I had a colleague, I had Dr Violet, and he’d come into Computing late. And he’d come into Computing as object orientation. And that was one of his first. I don’t think it was his first, but it was one of his first exposures. So he was into Smalltalk and quite heavily into object orientation. I was being, if you like, the devil’s advocate, so we would bounce, you know ideas, ‘thats a load of rubbish’, why would you want to do it that way, what’s wrong with procedural, it would give you this this and this advantage. And he would say: but then you’ve got this this. So, we would have the ability to bounce. But, even with that I started teaching object orientation, then I started teaching Java, I think I was the first one who introduced Java here. I thought I had it, but I’d say it took me two years. it was two years I realised I didn’t have it. I almost had it. But, I didn’t. I think object orientation is something that is vastly underestimated. As I say to my students, I say, its a head thing. Its not a technical thing. its not a learning thing. To learn that this does that. Its a head thing. You have to get your head round what’s happening. And I would say that took me two to three years to really get my head round how object orientation was working. And I think the only reason I did that was to do multithreading. And I think it was thinking about objects in a multithreading environment that really open that windows to say, ah!, there’s the link, I’ve got it now.

W: but, do you think you view it as Dr Violet does, or do you think you have a handle on it? - you’re handle?

B: eh

W: Dr Violet gets it. Dr Violet really gets it.
B: Oh yeh, I got it. i get it. But, i think it is a similar kind of question to asking if you are a fluent French speaker: are you thinking in French or English? You know. And I think my answer to that would be, I get it, I get it in both perfectly,, and i can think in both. but, what I;m actually thinking in at any particular time depends on what I am doing at that particular time. You know, what my context and what my environment is. You know, if I'm busy working on a particularly procedurally oriented programming problem or language then that's the way I'm thinking. I can always translate into OO. But, if I;m heavily into OO then I'll think in OO. I'll think in classes.

W: I must admit, when I think of attributes - data, methods - functions. I find it very, I do the translation bit.

B: what you have lost there is the object.

W: yeh.

B: <> state and behaviour into one.

W: yeh. My handle on object orientation is because of all the work I did on, I had a job writing a column for monthly magazines at one time, it produced some very useful money when the family were wee, an extra 200 quid every, I wrote about CASE about every month. And I think it Shlaer and Mellor, Sally Shlaer, a rare woman in Computing, and she did early work on what became UML. Rational Rose stuff. I remember reading through her stuff was really clear on what object oriented analysis and design was. And I now find I am now a native object oriented analysis and design person. I can see no reason to do it

B: any other way

W: yes. I am absolutely convinced it is the only way to do it. But, in programming terms I still think procedurally. And I think it is because the world is naturally consisting of encapsulated objects. There is you and what you know and what you do. And there is me. And we communicate with messages. But, computers are procedural. <laughs>

B: ah well, <> don't. We had a research project a few years ago where we got a, forgotten its name, it was a board where you plugged into a PDP. And it was actually object oriented architecture. So the actual computer

W: the Rekursiv

B: the Rekursiv. With a K. made by the Linn people. Now that was actually an object oriented architecture.

W: that's what Dr Violet did his PhD on. I remember he <> the room

B: Rekursiv with a K. So a couple of machine, there's a couple of LISP machines the States. Which, if you were into esoterica, you know, the sort of, ah, so there is a different way from doing it procedurally. But, yes it is, I think nowadays you are more and more, I think, the other big change in Computing is, I always think there's a spectrum between down in the bottom there's electricity, and right up at the top there's a problem you're trying to solve using that electricity. Now, back at the 60's the problem
solver had to of course cover the whole range, the whole spectrum, the electricity right through to the problem. And gradually as the years have progressed the problem solver is now much more problem oriented. You know, what i had to know as a programmer in the 70’s about how the computer worked was quite a lot.

W: an awful lot

B: whereas, now someone can solve problems on the computer, programming, “programming” because it might not be what we think is programming, using a spreadsheet, or whatever, program. But, someone can program and solve a problem without knowing one thing about how the computer works underneath. I mean, our concepts of files and folders are being reshaped. You know, VISTA’s going to give us a new view of that, for example. But we don’t, for example, have tot each computer organisation like we used to.

W: yeh, but how much? I used to teach the computer organisation on the Computing degree, ten years back. I picked it up from Dr Light. I loved it because you just sat and taught assembly language to people. the actual guts of the beast. This what it is doing. And, for the computer games tech degree, I forgot to write it in. And it wasn’t there. then we realised we had to add it back in. Because the games tech stuff is really nasty on the metal.

B: so they have to know a bit about

W: they have to know down there. but that is an exceptional thing.

B: I mean, if you are programming in Java or things like that

W: you don’t need to know that

B: so, eh, what’s even more interesting these days is that the machine that you need know about isn’t even a real machine. For example, if you are programming against the dot-net environment what your needing to look at is the CLR, the dot-net intermediate language, the dot-net assembly language doesn’t exist, because its run against a virtual machine.

W: I know nothing about dot-net. I’d never heard of it. Because I don’t need to. I mean, there’s something out there called dot-net. Like I care. Its a bit like, before I bought my diesel car I didn’t care about diesel.

B: does it go? does it cost less?

W: yeh. miles per gallon. So, I thought, OK, I ran my petrol car until somebody told me you could get 75 miles per gallon out of the diesel. I get 68 mile per gallon out of the diesel. its not bad. There’s some things you don’t need to know.

B: Do you think there’s a way we’ve moved onto a programming language, because you moved onto it, not because you believed in it. because it did the job better. Then you suddenly realised, oh, the bigger picture’s better as well. In one respect we move backwards because we were lucky, as you said, we had algol-w. We were in academic institutions in the 70’s ahead of our time. So we learned algol-w then you went
backwards. you went backwards in the real world into assembler, perhaps then forwards into COBOL, FORTRAN and then into C. And that was still way below algol-w in terms of, you know, programing language theory. And then you moved into what I wold call the real good sound theoretical languages which were pascal and java. Now, on the interim, we now had C++ appearing. Which I absolutely detest. As I say to students, C++ is a bastard language. Using that word in its true meaning. It doesn’t have any real parents. Its a bit of everything, you know. Its a bit of object oriented, a bit procedural, there’s no protection from memory management or hacking or anything else like that. And I think C++, when the history of Computing is writ, I would say C++ will be an interesting aside.

W: and yet its core to games

B: I think that’s purely through historical and through ignorance by the games producers. I can show them software in dot-net for example that is written in COBOL that runs faster than C++ programs. And that’s always what they get to, they always get to that C++ is fast.

W: COBOL and FORTRAN, you can’t get faster because they sit on top of the machine code. ADD A TO BE GIVING C ON SIZE ERROR, and you can actually see the instruction, flags been set, and there’s an overflow, a very very clean simply language actually. Even with your GOTO’s you’re just simply changing the program counter.

B: that’s exactly what goto was. PC- something or other.

W: one of the very simple languages. Thinking then about what’s in the Computing degree, because I haven’t taught it for some time, is there anything significant gone out of it in terms of theory?

B: yeh yeh yeh. I don’t think there is very little theory in it any more, as such. I think theory is one of these hidden curriculum things. The better students will get, they’ll get it as they go through. There is still a bit of theory, i mean, I do a course in 4th year on languages and compilers. But, that’s not in order to teach them language theory or compiler theory. Its actually about, here’s a way you can actually write a syntax diagram. And here’s an automatic way. I only touch one technology, recursive descent, I don’t talk about any of the other compiler technologies. And I say here’s a very simple way of going from that syntax to the compiler. Why would you need to? Well, you might be writing a game. You might want to find some way to interact with sentences. OK, that’s a small language. You might be writing a distributed system on the web where you want to pass messages. You design a protocol. Well, design it using BNF and you can use compiler technology to process it.

W: you actually teach them BNF?

B: yes. yes. yes. I’ve just finished writing them a tool to help them as well.

W: its interesting, if you take the games tech degree, what they lack in the games tech degree is game design theory. Because, they have to learn so much Computing and so much Maths to become a 3D graphics programmer that there’s actually no room to learn anything about games.
B: I think they're just as ignorant as our Computing students are. They certainly don't know any more about Computing theory either. I think they learn about one particular bit of Computing, and how to program in one particular way, in a graphics way, as you say, but I don't think they understand, I mean, if you start talking to them about coupling, cohesion, if you ask the questions, why? Object orientation is a better way of programming. Why? I don't think they could answer. Just like our Computing students couldn't answer.

W: Coupling and cohesion, that's your classic Ian Somerville stuff, that your classic software engineering stuff.

B: But it's also programming stuff.

W: Would you draw a line between software engineering and comp sci? Do you think there's a line?

B: I draw lines all over the place. I think there's a spectrum. Somebody ask me what I do, I'm just a simple programmer. I think that word has become dirty almost. And I think it goes back to the days of the 70's when the perception was that programming would be a technical <>, it would be done by technicians. Not by degree students. But, I'm sorry, 30 years later and we still don't know how to program. And I think programming, I'd use that word to describe the whole ten yards, as the Americans like to say. You know, its design, its quality, its testing. I think software engineering was simply a term that was conjured to, to do two things. To make it into something that would give it status. Like engineering. This was a status thing; it wasn't just playing with computers. And also, I think, to open the door to a number of people who technically, I know this sounds awful, but I know software engineers who can't program. I just found that amazing. They can wave hands. These business analysts, systems analysts, systems designers, who can't program. But they can wave hands and they've got big words for things that programmers do naturally because that's the way to do them. And I think that's one end of the spectrum. And the computer science is really <sigh> I'm not interested in the computer science except in so far as I know tis there and I trust it. What I'm interested in is applying it. I don't see there have been many advances in computer science in the last 30 years.

W: I'd agree with you there. I don't see the subject has changed.

B: And I don't think we have to teach it because its there and we just use it.

W: I don't think I really mentioned before, a smart chap, a really smart chap, John Smith, on the games tech. I think he came 5th in his country's annual leaving results in his year. His brother is coming here either this year or next year, and he came 4th. And he said, when he finishes, his parents are well-to-do, and he wants to go to Princeton to do an MSc in Computing Science because he reckons he hasn't got the underlying principles. I reckon he has more than he thinks.

B: Yeh. I think that sometimes what we do is to put labels on what we know. <> from what it is. Oh, that's what you call it. That's what I did naturally.

W: Coupling
B: coupling and cohesion. Oh that's called coupling. That's called cohesion. That's the theory underlying it. Fine. I mean, I gave up worrying about it when I saw it wasn't going anywhere at the time. I think the favourite one was in the quality era, the metric. The formal specifications. And there was all sorts of theory flying around and about, but they never went anywhere. I think denotational semantics was one approach, and that certainly gave me a further insight. But, it didn't go anywhere.

W: I've often got this model that Computing is a very tall tree with tiny little branches sticking out that don't go anywhere.

B: yeh.


B: But, having said that, things like fourth generation languages have disappeared, almost entirely. Whereas, databases, they didn't go anywhere, but they're now just accepted, they're there, full stop. And any real application, there'll be a relational database at the back of it. But, we don't worry about it any more.

W: not many people would even use the prefix relational. I remember teaching about object oriented databases.

B: what was there, there was object oriented, the was relational databases, what was the other one, used pointers, based on set theory. A database theory based on set theory. But yeh, there was, wasn't there. And those other ones have certainly gone up blind alleys. And relational is just the database, isn't it. But, I think the reason why we can change with it, these underlying principles are the same. I mean, if you look back, one of the biggest revolutions, I remember being gob-smacked at the time, but theoretically it was nothing. It was really just a visualisation of computer memory. The spreadsheet. Visicalc. I remember when I saw Visicalc running for the first time on an Apple, and I thought, what a brilliant idea. That is just so cool. You know, it was unbelievable.

W: dBase as well.

B: dBase, dBase II, dBase III. I just thought, wow.

W: I can remember the Data General machines

B: oh yes.

W: they didn't come with an index. They had a separate index. And the index produced a pointer. And the pointer gave you the block number and the record number within the block. And you had to manually maintain this index, because it would overflow. <> a three level database. And when it filled up you had to take ages and restructure it, and do it by hand. And so the concept of dBase was, you just typed it and it was there. I didn't have to do it by hand. I didn't have to manually do it. But, the fact that I had to do that meant that databases were simple. I knew what they were because I'd done them by hand.
B: I'm never quite sure whether <> good or bad <>. The other example where you had to do things by hand, I remember in every computer science course, whether in 1st or 2nd year, would have a course on data structures. Now, funnily enough, the HND still has, which shows how bad the HND has got. Because we haven't taught data structures for about 10 years. But, you used to have stacks, didn't you? Now, I always felt very uncomfortable teaching that, because, 1. you were teaching them what a stack was, the concept of a stack. Now, very trivial. Push pop. That was OK. Secondly, you were asking them what you might use a stack for? Algorithm design. That was OK. I didn't have an issue with that. But, thirdly, you were saying, here's some ways that you could write a stack. I still cringe to think we used to teach them to write stacks, to give them examples, to say here's writing a stack using an array, here's writing a stack using a circular array, here's implementing a stack using a linked list. So you're doping that but, remember, you're doing this in 1st or 2nd year. So, fourthly, your teaching them how to program because they are about becoming programmers. So, at the same time you are trying to teach them about programming, about how to implement stacks, about how use stacks, about algorithms and about what a stack actually was. And, no wonder they got confused. Nowadays, who cares how you implement a stack. Has anybody ever implemented a stack for the past 15 years. I don't think so. You know, you go into java, there's a class called stack. GO into C-sharp, dot-net, there's a class called stack. There's a class called queue. So, now you might concentrate upon, what might you use these for in an algorithm.

W: a lot of things don't matter. things like saving 2 bytes here and there. used to be critical.

B: absolutely, yeh.

W: make sure your words are left aligned

B: don't use a double use a float. Don't use a double there because that value will never be bigger than that.

W: don't use four bits, use two bits, but if its a four bit wide word extend it out to 4 bits

B: byte aligned. it might be more efficient if it is byte aligned rather than halfword aligned.

W: even things like 2s complement. That gone entirely.

B: floating point implementation with mantis exponent. Who needs to know about that any more. The trouble is that, sometimes, sometimes, you need it. I mean I try to teach them a little bit of Boolean algebra. Because thats still, fundamentally, what's going on. the logic flows form that because its still two-value logic that you are dealing with. But, they don't get that,. I mean, I had a student a year before, in the honours year project. it wasn't for you was it? Where they were trying to hide data inside images. Can't remember the word for that, but there's a word for that kind of thing. So they would take a JPG image and use the the bottom bit to actually store the first bit of the information they wanted to store. And so on. Anyway, they were having to get down to bit level. But, this guy actually created a data structure that was a string of characters. Each character was actually the character zero or the character 1. Ok, from the efficiency point of view it was awful. That was OK nowadays. But, it just illustrated
the fact that he just didn’t understand the bits, you could actually do things with bits on
a computer. It is so far removed from them now it never occurred to him that he could
actually do bits. You try to explain to students the difference between && and a single &,
between the logical AND and the bitwise AND. They’ve lost it now.

W: yes, you’re right. There are actually a lot of concepts which they can’t handle now
because the don’t <>

B: yeh, you’re mentally

W: it often makes me wonder, what do they not teach in a three year English degree that
we do teach in a four year degree. Are their graduates any different from ours? the
other side of that is, and English chap said to me, why does it take you a year longer to
teach what we teach?

<post-amble rundown>

B: looking back, i think that maybe we are doing a disservice to our students by making
them ready for market, and we’re giving them some basic principles, but we’re not giving
them basic theory. We’re giving them basic principles. And looking back on how have i
coped with change, it was probably because i was taught basic theory, all these year
back.

W: Giddens et al talked of mode 1 and mode 2 knowledge. We’re giving students mode
2 knowledge because that’s the fashion of the day.

B: we have to be vocational. No, I’m sorry, I’m sorry, and you can quote this as an old
<>, I’m sorry, but I am totally against this whole movement towards hand-waving, what
i call, and you have my permission to use this term in your work, edubabble.

W: edubabble.

B: edubabble. And the handwavingness, and that’s been linked in with this wonderful
word called employability. yeh. I;m sorry, do the employers not have to take some
responsibility. You know. For example, these transferrable skills. The student has to
be able to present the work. Why? How many of our students are going to be standing
out there in industry presenting their work to an audience. very few of them. How many
of our students will eventually become senior managers. Few of them. And yet, I’ve
seen students going through hell to do a presentation.


B: why are we doing this? Its up to employers to say, John, you seem to have quite a
good personality, your good at <> Or, Dr Blue, we’re sending you on a course for
presentation. you know, eh, we teach them silly things at silly times. group working,
you know, that’s one of the favourite in-things at the moment. team-based problem
solving. group hugs, you know. And what’s even worse, its becoming non-technical. I
mean, we used to have to work in technical groups, but now its in non-technical groups.
I;m sorry, but, you know, why does someone have to know how to manage a group.
Manage a project. they have to know that these things have to be done. You’re not

320
going to be a project manager till ten years down the line. then, that little thing about, yeh, there is project management. That’s when you can then learn the skills you need.

W: yes. <> you’ll not understand it for years
B: yeh.
Appendix E

Example Extracts of Data Encoding

Containing:

* first encodings after recording listenings

* groupings of codings under headings

* crib-sheet of codings names and meanings

* codings applied to interview transcripts

* common-by coding groupings of text portions with analysis and selection notes
INITIAL CODING THOUGHTS BASED ON LISTENING (READING ***1)

other - varied backgrounds, CS as Maths,

*** - programming is CS, knowledge, curriculum, CS theory, practice, labels, thought processes, underlying knowledge, real life v uni CS, new terms, loss of old CS knowledge in a few years, ugly v clean languages, thin knowledge in some named CS degrees, old v new unis, who controls the field? (uni v industry), CS K boundaries

*** - purpose, background, difficult concepts, abstractness of CS, programming is CS, creativity, OO, underlying knowledge, managerialism, field teaching v knowledge, lack of time for scholarship, real life v uni CS, applied CS knowledge, K surprises, new terms, stability of underlying CS theory, students enter with no CS knowledge, lack of new CS teachers, loss of old CS knowledge in a few years, innovation fatigue, ugly v clean languages, thin knowledge in some named CS degrees, BSc CS lost in 5 years, old v new unis, who controls the field? (uni v industry v management)

*** - student issues, university issues, change in CS, programming, CS K boundaries, who controls computing (uni v industry), underlying static CS knowledge, old & new CS staff, huge application range of CS, users, new K in CS, locating CS knowledge, obsolete K in CS, K gaps in CS academics, non-knowledge & lies.

*** - can't define CS, CS k can be obsolete or underlying, programming is difficult, CS is Maths, CS k change is hard to keep up with, CS k is huge & unstable, inaccurate info out-there from ignorant dabblers, CS theory is essential, graduate employability, CS k is industry-driven & unstructured, students' CS k is poorer, students thinking is undisciplined, students do OO better than academics, CS k is held by older academics, UAD doesn't do CS k or good academic degrees, change in academia, enjoyable job, managerialism, lack of research, corruption, illogical teaching decisions, lack of younger CS academics, soft V hard CS k, loss of Maths, CS k rate of change emulated in other subjects, lies / geekspeak & CS change problems of understanding, applied CS always improves

*** - increasing managerialism in academia, change as an aspect of our lives, CS is a western subject field, we do it for money & perception (but these are in reality poor), enjoy teaching & job security, CS k change, change in academia, political interference, financial limitations of academia, academic freedoms, students are punished / rewarded, CS k change is interesting, academic change is illogical, concentrate on a small aspect of CS k, political lies about academia, CS k change produces constant teaching change, illogical academic field management, corrupt job / task allocation, effects of poor management, CS k is different at new and old unis and new ones produce most useful graduates.

excited:

*** - non-programming in CS, edubbable, skills taught unnecessarily at uni, stupid industry decisions, pretentiousness

1 three asterisks indicate a deleted academic's name
*** - managerialism, K change, programming is fun, thin degrees (e.g. WDD)

*** - non-K (e.g. WDD), deception and lies, programming is my field

** - inaccurate knowledge in outside field and students polluted by this; extremely poor management of CS departments and general academia.

*** - political interference in academia, poor university management, illogical field management.
CODING BY MAJOR GROUPING v2

CUR curriculum - PRO programming, EDB edubabble, POL political forces, ABS abstract field, TLOS (loss of) theory, NAT (nature of c) thin/thick/easy/hard, APP applied v THE theoretical cs k, XTRA added to degrees, IMP important in a(ny) degree study, ANAL Systems Analysis, FUN enjoyable, BORE boring

KNO knowledge - CON control, OBS obsolescence, TERM new terms, SCOP scope/scale, PROV provenance, LOC location (KSTU student, KSTA staff, KACA academia, KIND industry, KOTH other, KUSER users), RATE rate of change, STAB (in)stability

STU students - THNK thinking styles, INK inherent CS knowledge, STVAC v academics’ k

UNI university - MGT new management styles, PCON control of personal work, OLDNEW old v new unis, AGPRF staff age profile, UOBS obsolete sectoral things, SELL selling degrees

PERS personal - HATE hates, LOVE loves, PAGE age, BACK background, CHNG changes, JOBS job security, FIN financial, SHOCK things that exasperate, HARD things that are hard to come to terms with, TIME the feeling of lack of time

CNTXT wider context - MAT Maths, CS, SE, PHYS Physics, ENG Engineering, AST Astronomy, BA Arts, ACAD the CS academy, IND industry, GEMP employable graduates.
CODING BY CODE NAME:-

ABS - abstract nature of the field in the curriculum
ACAD - the CS academy
AGPRF - universities staff age profile
APP - applied knowledge in the curriculum
BACK - personal background
CHNG - personal changes
CON - control of CS knowledge
CS - Computer Science curriculum
CUR - curriculum
EDB - edubbable in curriculum
ENG - Engineering
FIN - personal financial issues
GEMP - graduate employment/ability
HARD - things that the academic find hard to come to terms with
HATE - personal hates
IND - industry
IMP - what is crucially important in a(ny) degree study
INK - students’ inherent CS knowledge
JOB - personal job security
KACA - academia CS knowledge
KIND - industry VS knowledge
KNO - knowledge in CS
KOTH - CS knowledge not student/staff/industry/academy/users
KSTA - staff CS knowledge
KSTU - student CS knowledge
KUSER - CS knowledge in the users
LOC - location of CS knowledge
LOVE - personal loves
MAT - maths curriculum
MGT - university management style
NAT - nature of knowledge in the curriculum thin/thick/easy hard
OBS - obsolescence of CS knowledge
OLDNEW - old versus new universities
PAGE - personal age issues
PCON - personal control of academic work
PERS - personal academic stuff
PHYS - Physics
POL - political forces acting on the curriculum
PRO - programming curriculum
PROV - knowledge provenance in CS
RATE - rate of change of CS knowledge
SCOP - the scope or scale of knowledge in CS
SE - software engineering curriculum
SHOCK - things that exasperate
STAB - stability of CS knowledge
STU - students
STVAC - students’ versus academics’ knowledge
TERM - new terms for knowledge in CS
THE - theoretical knowledge in the curriculum
THNK - students thinking styles
TLOS - loss of theory in the curriculum
UNI - university
UOBS - obsolete things in the sector
XTRA - subfields added to degrees
gramming I got through, at university, so you went on the things like ML, denotational semantics and formal methods, the big thing in the 80s, formal methods, that wasn't a big deal because that was just functional programming taken on a bit. What really blew my mind was object orientation. And I think I was very lucky I had a colleague. I had Louis, and he'd come into computing late and he'd come into computing as object orientation. And that was one of his first. I don't think it was his first, but it was one of his first exposures. So he was into Smalltalk and quite heavily into object orientation. I was being, if you like, the devil's advocate, so we would bounce, you know ideas. 'That's a load of rubbish', why would you want to do it that way, what's wrong with procedural, it would give you this this and this advantage. And he would say, but then you've got this this. So, we would have the ability to bounce. But, even with that I started teaching object orientation, then I started teaching Java. I think I was the first one who introduced Java here. I thought I had it, but I'd say it took me two years. It was two years I realised I didn't have it. I almost had it. But, I didn't. I think object orientation is something that is vastly underestimated. As I say to my students, I say, its a head thing, its not a technical thing. Its not a learning thing. To learn that this does that. Its a head thing. You have to get your head round what's happening. And I would say that took me two to three years to really get my head round how object orientation was working. And I think the only reason I did that was to do multithreading. And I think it was thinking about objects in a multithreading environment that really open that windows to say, ahh, there's the link, I've got it now.

j: but, do you think you view it as does, or do you think you have a handle on it? - you're handle?

a: eh

j: Louis gets it. Louis really gets it.

a: Oh yeh, I got it. I get it. But, I think it is a similar kind of question to asking if you are a fluent French speaker. are you thinking in French or English? You know. And I think my answer to that would be, I get it, I get it in both perfectly, and I can think in both. but, what I'm actually thinking in at any particular time depends on what I am doing at that particular time. You know, what my context and what my environment is. You know, if I'm busy working on a particularly procedurally oriented programming problem or language then that's the way I'm thinking. I can always translate into OO. But, if I'm heavily into OO then I'll think in OO. I'll think in classes.

j: I must admit, when I think of attributes - data, methods - functions I find it very, I do the translation bit.

a: what you have lost there is the object
a: So, yes, I went from that, to coming here, which was 83 84 when I did the postgrad IT. The first cohort through the postgrad IT. And, of course, we used BBC's, modified BEM's, the Torch machines.

j: I remember the Torch machine, yeh.

a: which was BBC BASIC

j: they were dual processor

a: dual processor, there was a Z80 as well as a 6502.

j: I remember, coming from a minicomputer mainframe background, I had a terrible time getting my head round the microcomputers when they came out. What on earth these things were. Because they were single user machines. I mean, I had no concept of multithreading, of simultaneous multithreading. They had no concept of a batch job. There were a whole lot of concepts missing with these little boxes that sat on your desktop. Quite difficult actually to make the mental leap.

a: I didn't find that. I just went from these multiuser machines. Cos, at poly, our module on computing gave us access to the P200, what was it, ICL 4200, which you could actually get terminal access to if you were on the computing course. The Maths course. But, we could just submit batches and get results back overnight. So, I'd seen that type of machine as well. I mean, I'd see out little Data General.

j: that's still multi-user. On a little Nova you could maybe 8 terminals on it. I suppose its a multiuser machine which you could use as a single user machine.

a: yeh. So I'd seen both and was more familiar with single user.

j: yeh, that's interesting. Cos I remember, he lived on micros, I mean, his whole life was micros. He had no knowledge of minis and mainframes.

a: <laugh>

j: And I found it really strange, this micro thing. I mean, we all shared the same room together, these two different computing knowledge-bases and to ... I mean, the minis and the mainframes disappeared anyway.
<table>
<thead>
<tr>
<th>HARD</th>
<th>But, tis a very straightforward language, Java is not.</th>
</tr>
</thead>
<tbody>
<tr>
<td>a:</td>
<td>no</td>
</tr>
<tr>
<td>j:</td>
<td>Java is much, I mean, would say it is a more complex language than FORTRAN or Algol?</td>
</tr>
<tr>
<td>a:</td>
<td>The trouble with object oriented languages and, to some extent, other languages as well, is that you not only need to know how to put together the flow of control to make your thing work, but you have to know the hierarchy of objects, or hierarchy of functions which you've got in this library. So you have effectively two things you've got to know. If you take BASIC, I'm not talking about Visual BASIC here.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TERMS</th>
<th>Visual BASIC</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>OBS</th>
<th>j: old gee whizz BASIC</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>HARD</th>
<th>a: old gee whizz BASIC. You had a few things that helped you manipulate strings, whatever, but most of stuff you had to do yourself. You had to work out how to do things yourself. To manipulate these sets of variables. But, with something like Java, the whole point is that you need to take, to use a class and you know what the classes are and the methods are so that you can tackle each problem. SO, here's this particular problem and I want, which class do I go to, and there's this vast array of classes. Which one do I go to? Oh bugger, I'll just do it myself. And so, you've got this extra load that you've got to get inside your brain. To say, I need to do this. The whole point of Java and the object oriented thing is that you take what's already been done. It's the class. Just take and change what you need to and &lt;&gt; make it work for your particular brand or whatever the class is.</th>
</tr>
</thead>
<tbody>
<tr>
<td>j:</td>
<td>its ... why has something as complex as Java caught on? Why do you think that is?</td>
</tr>
<tr>
<td>a:</td>
<td>Its a hard question to answer. Whether society in general has moved into a mode whereby people don't bother to evaluate things, they just say, here's the new thing, it must be better. The hype that came in with Java said, this is so much better, and people just said, OK, lets see if tis the thing to do we'll do it. But why that worked with hard-nosed professionals who've been doing C++</td>
</tr>
</tbody>
</table>

<p>|       | j: they're much more like real programming languages, while Java, its really most strange thing. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MGT</td>
<td>The senior managers want to do their job, I'm not blaming them, but the senior managers want to an academic job, to manage the university, but their political masters don't give them enough money and they tell them, you have to give &lt;&gt; performance figures and indicators and things like that, because they say, politically they have to account for how much money's being spent.</td>
</tr>
<tr>
<td>POL</td>
<td>The senior managers want to do their job, I'm not blaming them, but the senior managers want to an academic job, to manage the university, but their political masters don't give them enough money and they tell them, you have to give &lt;&gt; performance figures and indicators and things like that, because they say, politically they have to account for how much money's being spent. Knowing very well that they don't care about how they spend themselves. But others cannot stand it, and, hence this curtails academic freedom by the back door under the guise of things like costs, and things like that. They are controlling everything.</td>
</tr>
<tr>
<td>FIN</td>
<td>The senior managers want to do their job, I'm not blaming them, but the senior managers want to an academic job, to manage the university, but their political masters don't give them enough money and they tell them, you have to give &lt;&gt; performance figures and indicators and things like that, because they say, politically they have to account for how much money's being spent. Knowing very well that they don't care about how they spend themselves. But others cannot stand it, and, hence this curtails academic freedom by the back door under the guise of things like costs, and things like that. They are controlling everything.</td>
</tr>
<tr>
<td>PCON</td>
<td>The senior managers want to do their job, I'm not blaming them, but the senior managers want to an academic job, to manage the university, but their political masters don't give them enough money and they tell them, you have to give &lt;&gt; performance figures and indicators and things like that, because they say, politically they have to account for how much money's being spent. Knowing very well that they don't care about how they spend themselves. But others cannot stand it, and, hence this curtails academic freedom by the back door under the guise of things like costs, and things like that. They are controlling everything.</td>
</tr>
</tbody>
</table>
What the CS academics feel last