1992

The A.D.E. taxonomy of spreadsheet application development

Maria Jean Hall

USE OF THESIS

THE A.D.E. TAXONOMY OF SPREADSHEET APPLICATION DEVELOPMENT

by

Maria Jean Johnstone Hall B.Sc., Grad. Dip.

Applied Science Computer Studies

A Dissertation submitted in Fulfilment of

the Requirements for the Award of

Master of Applied Science

at the Faculty of Science and Technology,

Edith Cowan University

Date of Submission: June 1992

ABSTRACT

Spreadsheets are a major application in end-user computing, one of the fastest growing areas of computing. Studies have shown that 30% of spreadsheet applications contain errors. As major decisions are often made with the assistance of spreadsheets, the control of spreadsheet applications is a matter of concern to end-user developers, managers, EDP auditors and computer professionals.

The application of appropriate controls to the spreadsheet development process requires prior categorisation of the spreadsheet application. The special-purpose A.D.E. (Application, Development, Environment) taxonomy of spreadsheet application development was evolved by mathematical taxonomic methods to categorise spreadsheet development projects to facilitate their management and control.

Data was collected on a sample of Australian developed spreadsheet applications. The sampled spreadsheets exhibited a very low level of managerial, I.T. department and auditor control. The data was analysed both by hierarchical cluster analysis using average linkage with the Euclidean distance measure, and by partitioned cluster analysis using the kmeans algorithm. The A.D.E. taxonomy of spreadsheet application development was developed in three sections from these analyses, categorising: A - the spreadsheet application, D - the developer and E - the development environment. A diagnostic key was developed for each of the three sections.

The A.D.E. taxonomy was validated by inter-rater comparison of the same spreadsheet and by two categorisations by the same rater three months apart. The validity of the clusters, used to develop the taxonomy was established and the taxonomy was also validated under a 'usefulness' criterion. A follow-up study to develop a spreadsheet development 'control model' was foreshadowed.

DECLARATION

I certify that this thesis does not incorporate, without acknowledgment, any material previously submitted for a degree or diploma in any institution of higher education; and that to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where due reference is made in the text.

Date ....29/6/92....

ACKNOWLEDGMENTS

The writer wishes to express her sincere appreciation to those who have assisted, either as participants or as advisers throughout the duration of this study.

Special thanks are extended to the study's supervisor, Associate Professor Ronald Hartley for his professional advice and encouragement. With his assistance and continual gentle encouragement, this study was completed on schedule.

My thanks is extended to Associate Professor Anthony Watson and Dr Jim Millar for their guidance in the early days of this study. I also thank the University research consultancy, Associate Professor Phil Clift and Jane Armstrong for their suggestions.

The interest shown by many colleagues, and the donation of their valuable time by survey participants are also gratefully acknowledged.

The purchase of statistical software and survey costs were funded through an Edith Cowan University internal research grant and thanks is extended to the University for this support.

Last but not least, I thank my husband William and son Andrew for their patience, consideration and support, and my son James for his invaluable contribution as a good listener, and sometime Devil's Advocate, when considering some of the ideas canvassed in this dissertation.

Table of Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

1.1. Chapter Overview

This chapter introduces the context of the study. The rapid growth in the use of PCs (Personal Computers) in Australia is outlined as is the importance of spreadsheet output as an aid to management decision making. Other studies reporting spreadsheet errors, and reports of business losses due to spreadsheets are used to establish a need for the control of spreadsheet development.

Two justifications for the study are given: The need for computer professionals to be concerned about quality assurance and control of end-user computing and the necessity first to measure before applying control.

Primary and secondary goals of the study are established involving the derivation of a special-purpose taxonomy of spreadsheet application development for use in the control of end-user created spreadsheets. Some theoretical and practical implications of a taxonomy are canvassed and subsequent chapters of this dissertation are outlined.

1.2. Spreadsheet Applications

Electronic spreadsheets, based on the familiar accountant's financial ledger, are a major application in end-user computing, the fastest growing area of computing. Schmitt (1988, p. 1) defines end-user computing to be "all forms of computing that originate outside the DP (data processing) department's control" or less broadly "that which occurs when an employee, usually not a DP professional, develops a computer application that aids the employee in the performance of his or her job".

A spreadsheet program is considered to be any commercially available personal computer based software application package that allows the user dynamically to manipulate text, numbers and formulae stored in a row by column format in a matrix of cells. The contents of the cells are held electronically and displayed on a computer screen.

A spreadsheet application is a model or template developed using a spreadsheet package. Such applications are usually, but not solely, developed by end-users.

1.3. Background to the Research Problem

Over the last ten years, there has been a rapid expansion in the use of PCs in Australia and more end-users than ever before are developing spreadsheet applications. Many of these applications are developed with no input or control from EDP (electronic data processing) auditors or managers. Studies have shown that one in three spreadsheet applications contain errors. This is of concern when considering spreadsheet usage in the support of management decision making.

Clearly spreadsheet development control is required, however it is unnecessary and not cost-effective to control all spreadsheets. A taxonomy of spreadsheet application development would allow the classification of spreadsheet development projects. Those requiring control could then be identified and controls appropriate to that class in the taxonomy could be selected.

1.3.1. The Growth in End-User Computing

End-user computing has experienced rapid growth in the last twelve years. In 1981, Rockart and Flannery reported in Benson (1983, p. 35) made some predictions based on their measured growth of end-user computing in seven large American companies. At that time, traditional data processing was growing at the rate of 5 to 15% a year while end-user computing had a growth rate of between 50% and 90%. They forecast that end-user computing would occupy up to 75% of corporate computing resources by 1990.

Guimares and Ramanujam (1986, p. 179) report on a) the Boston based Yankee Group's estimate of 2.7 million microcomputers in the United States in 1982 rising to 5.4 million in 1984 and b) Booz Allen Hamilton's estimate of 2.6 million in 1982, 4.6 million in 1984 reaching 13 million by 1990.

Benson (1983, p. 35) reported that International Data Corporation estimated that four out of five administrative workers would be using personal computers by 1990. Udell (1990) reported that by that year, 30 million microcomputers were using DOS world-wide. Udell's estimate did not include the number of personal computers using alternative operating systems.

Table 1: Estimates of Worldwide Growth in Personal Computing

BY YEAR	SOURCE	ESTIMATION
1971		First microprocessor
1975		First microcomputer
1982	Booz Allen Hamilton(1986)	2.6 million microcomputers in U.S.A.
1984	Booz Allen Hamilton(1986)	4.6 million in U.S.A.
1984	Yankee Group	5.4 million in U.S.A.
1989	Wright (1990)	1 in every 36 Australians
1990	Booz Allen Hamilton(1986)	13 million microcomputers U.S.A.
1990	Udell (1990)	30 million DOS users worldwide
1990	Rockart & Flannery (1981)	75% of corporate computing resources
1990	Benson (1983)	4 out of 5 administrative workers
1993	Wright (1990)	1 in every 6 Australians

This phenomenal growth pattern has been replicated in Australia. PCs gained respectability in Australia in 1983 with the introduction of IBM's Personal Computer. In 1987 the Australian PC market was worth $678 million. Two years later the market was worth $1.68 billion. By 1989 One in thirty six Australians used a PC, and by 1993, this figure is expected to rise to one in six. (Wright, R., 1990, p. 102)

1.3.2. Thirteen Years of Spreadsheet Software

Spreadsheets do not have a long history. Their evolution over the last few years has been so rapid, that it has outstripped the efforts of management, auditors and DP professionals to exert control over end-user created templates.

The first electronic spreadsheets, then called 'row column manipulators', were developed in the late 1960s for large mini and mainframe computers. They did not receive a wide usage as access to them was largely restricted to the Computer Services department due to complex operating systems and expensive use of valuable mainframe computer time. (Goss, Dillon and Kendrick, 1989, p. 20)

VISICALC, the first microcomputer spreadsheet was introduced for the Apple II in 1979 and quickly became the de facto standard. It was developed by two MIT graduates, Bob Frankston and Dan Bricklin, and marketed by their Harvard Business School marketing student colleague, Dan Flystra. Licklider considers that the spreadsheet was the catalyst for the change of the microcomputer from "a hobbyist's novelty into an essential tool for financial analysts". (1989, p. 324)

Context MBA, the first integrated spreadsheet, with the addition of windows, graphics, file management, and word processing was introduced in 1981. Stand-alone spreadsheets continued to gain in popularity and a survey by Benson in 1982 found VISICALC in use in over 80% of the PCs surveyed, and the primary or exclusive software on 60% of those PCs. (Benson, 1983, p. 39)

Lotus 123 entered the market in 1982, introducing the concepts of natural-order recalculation and macros. Within a couple of years Lotus had displaced VISICALC as the de facto standard. By 1984 spreadsheet software had become popular with over a million packages sold that year, in the U.S.A. alone. (Brown & Gould, 1987, p. 258)

Integrated packages containing spreadsheets also increased in popularity with Ashton-Tate's Framework, Lotus Symphony, Apple's Apple-works and Visi-corp's

VisiON leading the way. Microsoft's Excel extended GUI (graphical user interface) spreadsheets to a wide audience and became the predominant spreadsheet on the Apple Macintosh. This popular spreadsheet was later ported to the IBM P.C.

By 1985 Lotus compatible programs had appeared; Mosaic's TWIN, Paperback software's VP-Planner, Borland's Quattro Pro, Javelin Software's Javelin, Computer Associates Supercalc and the Software Group's Enable. Three dimensional spreadsheets were pioneered by Supercalc and Enable.

Lotus 123 version 3.0 extended spreadsheets to the OS/2 environment. Supercalc 5 appeared on IBM mainframes and spreadsheets such as Lotus Improv appeared on UNIX, PICK or VAX platforms benefiting from such features as virtual memory, transparent networking, multi-user capabilities and multi-tasking. (Yager, 1990, p. 147)

Ware (1986, p. 63) reports that spreadsheets, and VISICALC in particular, have been credited with much of the early growth in microcomputers. Spreadsheets gave users their first taste of PC user-friendly functionality, which had no counterpart on the mainframe. Connors (1984, p. 16) reported that 90% of PC users, who responded to an American National Association of Accountants survey, used spreadsheets and the availability of spreadsheet software was the main reason for respondents computer purchase. A 1986 survey reported by Ware (1986, p. 63) showed that spreadsheets were used on nearly 80% of all microcomputers.

During this rapid expansion phase, spreadsheet popularity has not been confined to accountants, and this writer's recent inquiry of the Sydney Lotus Users' group solicited the response that most spreadsheet users in that large group of spreadsheet enthusiasts, were administrators rather than accountants or engineers.

With the relatively recent introduction of three dimensional spreadsheets and spreadsheets running in WIMP (Windows, Icons, Mouse and Pull-down menus) and GUI (graphical user interface) environments, the continued popularity of this type of application software seems assured. New generation spreadsheets such as LOTUS 123 for Windows and EXCEL are placing a heavy emphasis on presentation and WYSIWYG (what you see is what you get). They are attracting a new generation of enthusiasts. Graduates of many disciplines from business colleges, TAFE colleges and Universities have been exposed to this type of software and the new generation of computing courses in many of our high schools has introduced a vast audience to the by now, not so humble, spreadsheet.

1.3.3. The Use of Spreadsheets as an Aid to Decision Making

Spreadsheets are used in the work-place for many purposes including the presentation, reporting and communication of information. They can transform manually tedious and time consuming tasks into quick and easy electronic tasks. Forecasting, trend analysis, "what if" analysis and goal seeking or optimiser models have been developed by many end-users to assist management decision making. A survey conducted by Aggarawal and Ob ak (1987) reported by (Goss, Dillon and Kendrick, 1989, p. 21) found that spreadsheets were the most popular type of software employed for strategic decision making.

Managers, not spreadsheets, make decisions out as Paxton (1991, p. 20) points out, "A manager's decisions will be no better than the data on which they are based." There is an unfortunate trend not to question computer output too deeply. Beitman reports that

> Many executives tend to accept electronic spreadsheet print-outs as 'gospel' without questioning their accuracy or validity. (Beitman, 1986, p. 8)

Moskowitz confirms this:

> Ever since the first computer crunched the first number, users have shown a proclivity to respect computerised output much more than it probably deserves. (Moskowitz, 1987a, p. 40)

Why is this so? Paxton (1991, p. 20) argues that users of traditional mainframe computer generated output have learned to trust such data as it is normally subjected to stringent EDP controls. This trust is misplaced when considering PC generated output which has not been subject to EDP department or audit control.

In many organisations, end-users develop personal spreadsheet based systems to automate some of their manual job functions. These informal or personal systems run alongside the corporate computer system without being subjected to the control, quality assurance or formal development methodologies of the latter. Parker (1988, p. 16) suggests that it is only a small step for such personal systems to be legitimised as part of the corporate computer system. This can occur by default when other employees learn to rely on having access, on a regular basis, to the output of some-one else's personal system.

Managers and decision makers who rely on spreadsheet data produced by others on personal rather than corporate systems, are vulnerable in three ways; (Paxton, 1991, p. 23) a) data may not be available when it is required, b) data may be available but erroneous and c) data may be available and valid but not in a form the decision maker understands. These spreadsheet problems arising out of uncontrolled end-user developed systems, expose an organisation to risk, when the spreadsheet output is required to support major economic or strategic decision making.

1.3.4. Errors in Spreadsheet Applications

Howitt identified the one major cause of problems in end-user computing:

> The computer's remarkable power to get more work done faster also creates the opportunity to make more mistakes and multiply them rapidly. (Howitt, 1985, p. 26)

This is particularly relevant to spreadsheets, which often are developed so quickly and easily, that many users fail to use a consistent and thorough design methodology, or test and document their product. Spreadsheet amendments compound this problem, as they are frequently made in an ad hoc manner often with no documentation of the changes.

Kee (1988, p. 55) reports that the typical spreadsheet developer is a "manager with limited knowledge of programming standards". and Edge and Wilson (1990, p. 36) point out that end-users, who are not IT Specialists, may be unaware of the need for controlling spreadsheet development.

What portion of spreadsheet applications are flawed?

Are spreadsheet applications really such a major source of error in the personal computing environment? Over the last five years, much has been written in both the academic journals and trade press, concerning the prevalence of errors in spreadsheet models. Guimares and Ramanujam (1986 p. 179) conducted a field study of 400 top American firms. They reported that one of the most critical problems seen in end user computing was the need to assure the integrity of both data and applications.

Other researchers have conducted surveys and experimental studies in an attempt to quantify the proportion of flawed spreadsheet applications. Bryan (1986, p. 39) reports that one in every five spreadsheets has errors. Creeth (1985, p. 92) reports that some industry experts consider that errors are present in one in every three spreadsheet applications. Ditlea (1987, p. 60) reports that this statistic has been confirmed by two Silicon Valley consultancies, Input and Palo Alta Research. Howitt (1985, p. 26), and Greenberg (1986) reported by Paxton (1991, p. 21) have also confirmed this one in three error rate.

Experimental studies on errors in personal computing have been conducted by Card, Moran and Newell, Brown and Gould and Davies and Ikin.

Card, Moran and Newell (1983) conducted a series of experiments at the Xerox Palo Alto Research Centre on subjects using word processors and text editors. They were interested in identifying the causes of errors. They found that even skilled operators made a substantial number of data entry errors.

Brown and Gould (1987, p. 259) conducted an experimental study of nine IBM employees, all experienced Lotus 123 users who carried out three identical spreadsheet application development tasks. All participants were confident of the accuracy of their spreadsheet templates, however Brown and Gould conservatively determined that 44% of the applications contained errors. Only 18% of the total errors could be attributed to petty typing errors.

The Australian experience has been similar. Davies and Ikin from the Tasmanian Institute of Technology analysed nineteen worksheets from experienced Lotus 123 users spread across ten companies. Again all developers were confident of the error-free status of their applications, yet 83% of the applications contained some form of error and 14% of the spreadsheets contained significant errors (Davies and Ikin, 1987, p. 54).

Incidences of spreadsheet error

Berry (1986, p. 36), Ditlea (1987, p. 60) and Stone and Black (1989, p. 131) report on one celebrated case of spreadsheet error. A Fort Lauderdale construction company, James A Cummings Inc. eventually dropped a lawsuit against Lotus Development Corporation and IBM for millions of dollars of damages it claims were caused by an error in LOTUS SYMPHONY. The company controller and application developer created an error when he inserted an extra row at the top of a range addressed by a @SUM function for expenses of $254,000. These expenses were subsequently not included in the range summation of the total costing of a bid for the construction of a 3 million dollar office complex for a local utility. The Lotus 123 Application packaging now contains advice to users to verify their work.

Parker (1988, P. 16) and Paxton (1991, p. 20) report on the termination of employment of six Dallas oil and gas company executives who made an incorrect substantial investment decision based on erroneous spreadsheet output, costing their company several million dollars during a major acquisition. Parker also reports on a $36 million underestimation of the size of a market for computer aided design equipment due to the 'rounding up' of a .06 inflation rate to 1.00 (Parker, 1988, p. 16). The press has reported many additional 'disasters' in recent years.

Ballou, Pazer, Belardo and Klein (1987, p. 13) also express concern about the lack of spreadsheet control procedures to ensure data quality as does Sato who reports that end-user computing is expanding at a faster rate than corporate information systems as a whole. This is causing control problems, not least because end-user spreadsheet development is often distributed and geographically distant from the EDP department. End-user computing is essential for an organisation to retain its competitive edge, however it has to be controlled "to attain integrity of data, information and decision making" (Sato, 1989, p. 7).

Moskowitz (1987b, p. 51) sums up the lack of control thus:

> The situation may be a universally shared but generally unspoken nightmare of the corporate world: thousands of employees devote millions of hours to electronic templates used to calculate the flow of billions of dollars - yet much of the exercise is wasted because the calculations are dangerously flawed.

1.3.5. The Computer Professional's Responsibility

Naomi Karten, computer consultant and lecturer on end-user computing is the editor of Auerbach Publishers' Managing End-User Computing. She reports that spreadsheets are the greatest potential internal source for data processing errors within an organisation:

> Users and systems developers are in the best (or worst) position to damage perhaps inadvertently, their companies' systems, the business data they contain and the business decisions that depend on that data. (Karten, 1989, p. 29)

She considers it the responsibility of computer professionals, particularly user support personnel, continuously to educate and remind end-users of the potential problems.

> Educating users is an important step in maintaining spreadsheet sanity. (Karten, 1989, p. 30)

Steenbergen (1989) in an editorial in the September 1989 W.A. Offline magazine, mouthpiece of the Australian Computer Society, expresses the concern the computer professional should feel about the lack of quality assurance being taken in personal computing with the continuing flow of application development away from DP professionals to end users. He suggests that:

> DP professionals have a part to play in educating users and management in personal computing quality assurance . . . . We have a job to do. Maintain the standard!

There have been some efforts in this area by Data Processing and other related Professionals. Flower (1989, p. 852) recognises the problem and asks who holds the responsibility for assuring the quality and integrity of spreadsheet output. Ashworth (1987. p. 136) finds the problem all too familiar:

> DP professionals have been coping with similar problems for years. The absence of standards for programmers to work to, has always lead to varying degrees of chaos. Over time the DP profession has developed methodologies to assist in the regulation process.

He suggests controlling spreadsheet application development with software engineering methodologies similar to those applied to programming. Other authors (Stone and Black, 1989, p. 131), (Simkin, 1987, p. 130), (Ghosal and Caster, 1990, p. 40), (Ware, 1986, p. 63) suggest structured spreadsheet development methodologies and spreadsheet development standards. Paxton (1991, p. 22) approaches the problem from an accountant's viewpoint and suggests that spreadsheet development is best controlled by the AIS (Accounting Information Systems) function.

The study described in this dissertation, is the first part of a response to Karten's and Steenbergen's pleas for DP professionals to accept their responsibilities with regard to personal computing:

> If the potential of the computer is to be realised, then human error must be controlled. (Bailey, 1983, p. 11)

1.3.6. Do all Spreadsheets Require Control?

Early surveys conducted by a) Aurbach publishers and Schultz and Redding in 1982, reported in Schultz and Hoglund (1986, p. 46), b) Price Waterhouse reported in Grant, Colford and Daly (1984), c) Schultz and Hoglund (1986), and d) Hoglund (1984) unpublished thesis, all concluded that whereas management usually imposed controls on the selection and purchase of software and hardware within their organisations, less than one third imposed controls on user developed applications.

Since the early eighties various control measures have been proposed with a wide range of degree of rigour. Whilst most authors agree that a significant problem does exist (Flower, 1989, p. 852), (Ashworth, 1987, p. 136), opinions as to what to do to control the situation are divided. The background and professional discipline of the author may have an influence in determining the degree of control proposed.

Pro-control

Many reports in the literature, mostly represented in the accounting, auditing and professional management journals are concerned with the management control of spreadsheet models. There is a frequently expressed concern that major business decisions are based on model output that has a probability of 30% of being flawed. Their answer is a rigid set of controls. (Kee and Mason, 1988, p. 46), (Williams, 1989, p. 46). However Kee and Mason do soften this stance by suggesting that "as many controls as feasible should be delegated to the user". (1988, p. 47)

Auditing sources such as Gaston (1986, P. 47) are concerned about the difficulties of controlling spreadsheet templates that may seem simple to the end-user, however Ghosal and Carter place the responsibility for control, on the developer: "Developing spreadsheets is no longer a private art form." (1990, p. 39) Other authors get rid of the problem altogether, by suggesting that, frequently, spreadsheets are an inappropriate tool and should be replaced by specialist decision support or accounting software. (Edge and Wilson, 1990, p. 38), (Howitt, 1985, p. 29)

Some authors extend the design and control techniques used in other more traditional areas of data processing. Bromley (1985, p. 136) and Goss, Dillon and Kendrick (1989, p. 23) based spreadsheet layout on the divisions of a COBOL program. Ashworth (1987, p. 137) and Hayen and Peters (1989, p. 31) suggest controlling spreadsheet development using a software engineering software development life cycle, while Ronen, Palley and Lucas (1989, p. 84) propose a spreadsheet development life cycle and spreadsheet flow diagrams.

Laissez-faire

A smaller number of articles take an opposing view. Computer trade articles, the hobbyist press and a few academics promote the freedom, creativity and user seductiveness of spreadsheet software. Ronen, Palley and Lucas (1989, p. 84) note that the tool's simplicity and transparency allow the end-user an easy expression of a model that might not have been considered worthwhile if rigid control was mandatory.

The middle ground

These authors recognise that a varied degree of control is necessary in some circumstances. Schultz and Hoglund (1986, p. 49) feel that users must be permitted to be creative with their personal computers and this could be hampered by applying strict controls to all worksheets. They recognise however that some worksheets do require control:

> It is neither desirable nor effective to stifle user creativity by enforcing burdensome controls over all types of microcomputer applications. However some programs are particularly critical to the firms success and therefore must be subject to sufficient controls to ensure that they are free from error. . . . . This degree of control enforced over user-developed applications should be a function of the potential for material harm that an invalid application presents. (Schultz and Hoglund, 1986, p. 50)

Canning (1984, p. 2) surveyed the views of information systems executives, concluding that they too were concerned with controlling spreadsheet development while wishing to retain an environment with the necessary degree of freedom for developers.

Chambers and Court (1986, p. 93) suggest that control should be determined by application function:

> The extent to which computer operations should be controlled, should be a function of what the computer is asked to do, not of how much it costs.

Paxton (1991, p. 21) agrees that not every spreadsheet needs to be fully controlled, and suggests that control procedures be limited to applications where there is a "favourable cost / benefit relationship". Gerrity and Rockhart (1986, p. 31) concur, and suggest a different degree of control for different types of spreadsheet models. Krull (1986, p. 36) suggests that control, where necessary, be distributed to the end-user.

There appears to be a need for an extensive spreadsheet application taxonomy to categorise projects. The availability of a taxonomy would allow the easy identification of spreadsheet development projects that do require control. This taxonomy would also facilitate comparisons of the design and control recommendations proposed by different authors. The two opposing viewpoints regarding spreadsheet controls may not be so far apart as they initially seem. They may be controlling different categories of spreadsheet applications.

Lack of suitable taxonomies in the literature

Some attempts to develop taxonomies for end-user computing in general and spreadsheet development in particular have been documented in the literature. Most of these are either incomplete or not suitable to be used with a control model to suggest application appropriate controls. Chapter two discusses these partial taxonomies.

1.4. Study Focus

The researcher proposes a two part project to develop tools to assist spreadsheet application developers ensure that they design quality, secure applications of integrity. It is necessary first to categorise and measure what one seeks to control. Only then can appropriate controls be determined.

This dissertation describes the first stage of the project, which will derive and validate a taxonomy of spreadsheet application development. The second stage of the project (outside the scope of this current study) will develop an end-user spreadsheet control model. Use of this model will further validate the taxonomy under the criteria of usefulness. The taxonomy, with a check list of security, design and control mechanisms will be used to suggest appropriate design criteria and control mechanisms to a spreadsheet application developer. A future study, comprising the second stage of the project, is foreshadowed in the final chapter of this thesis.

A taxonomy of spreadsheet application development will be of value to developers for the categorisation of proposed or existing spreadsheet projects, to managers and EDP auditors who seek to control spreadsheet development and to other researchers who may wish to compare reports from the literature regarding the control of spreadsheet application development.

1.4.1. Primary Research Goals

This study had two primary research goals:

a) Improve the planning and management of spreadsheet application development

b) Develop a special-purpose classification - Taxonomy of Spreadsheet Application Development for use in controlling spreadsheet development

1.4.2. Secondary Research Goals

The study had many secondary research goals. They can be considered in three broad areas: a) concerning collection and analysis of a data sample, b) concerning the cluster analysis process and c) concerning the validation of the taxonomy.

Collection and analysis of the data sample.

- Identification of a suitable sampling frame and primary collection of data on spreadsheet application development.
- Sample Data reduction / simplification. Through exploratory data analysis and data reduction, gain a better understanding of the underlying data structure.
- Generation of hypotheses for future testing

Cluster analysis

- Achieve well structured clusters
- Achieve Intuitive Clusters
- Achieve clusters from which a suitable taxonomy can be developed

Validation of the Taxonomy

- Demonstrate Taxonomic Stability - Adding few cases or attributes to the analysis does not appreciably change the taxonomy
- Demonstrate Taxonomic Robustness - Removing one or two objects or attributes does not disturb the classification
- Demonstrate Taxonomic Replicability - Agreement between different multivariate methods
- Demonstrate agreement with taxonomies from the literature
- Demonstrate agreement with own a priori expectations
- Demonstrate the usefulness of the taxonomy
- Validation of the diagnostic key of the taxonomy

1.5. Significance of this Study

This study is theoretically significant as it produces a new method of categorising the development of spreadsheet applications, which should be of interest to end-user developers, EDP auditors, managers and other researchers. The taxonomy is also of theoretical interest as it was developed by applying the methods of classical mathematical taxonomy to the new fields of end-user computing in general and spreadsheets in particular.

The study also has some practical significance as it develops a sampling frame of spreadsheet developers that could be reused. It goes some way towards defining the variability of Australian spreadsheet development practice.

1.6. Scope and Limitations of this Study

The study is limited to aspects of end-user computing in Australia involving the development of applications using spreadsheet software. It is restricted to the development and validation of a taxonomy of spreadsheet application development designed for the special purpose of the management control of spreadsheet usage.

It is recognised that the primary research goal of improving the management and control of spreadsheet development projects, will only be satisfied when a 'control model' is produced to be used in tandem with the taxonomy to suggest application appropriate design and control criteria. This dissertation describes a study that goes some way towards achieving this goal, however it stops short of producing a control model. The final chapter of this thesis outlines how this current study could be extended to produce a model for the control of spreadsheet development.

1.7. Outline of Subsequent Chapters of this Dissertation

The second chapter reviews the literature for articles of relevance to this study. The history of categorisation is outlined, leading to the development and use of taxonomies both in other fields and in computer science. Taxonomies with particular relevance to the broad area of end-user computing are canvassed as are the more specific partial taxonomies of the spreadsheet development process.

Exploratory data analysis methodologies are discussed together with an overview of mathematical taxonomic methods. The view of a taxonomy as one of many possible models of reality, and criteria for selecting the 'best' model are established. Reports from the literature are used to justify the selection of appropriate attributes of the spreadsheet development process to be used in the development of this special-purpose taxonomy.

The third chapter details the study methodology and design. A data collection survey is described. Methods are outlined for multivariate data analysis using hierarchical cluster analysis and partitioning kmeans techniques. The evolution of the three-part A.D.E. taxonomy of spreadsheet application development and its diagnostic keys are described.

The fourth chapter reports on the results of the survey, and one hundred and fifty cluster analysis runs with variable parameters. The development of the three part A.D.E. taxonomy, its cluster profiles and diagnostic keys are described.

Chapter 5 covers the validation of the A.D.E. taxonomy and the survey data collection instrument. Chapter 6 concludes this dissertation, makes some recommendations and outlines future research directions extending this study. In particular, the development of a spreadsheet 'control' model is foreshadowed.

Material in appendices A-E support the methodology, result and validation chapters.

1.8. Summary of this Chapter

This chapter introduced the problem of spreadsheet errors and placed it in a context of concern both to Australian managers and IT professionals. A broad research focus was determined, involving improvement in the management of spreadsheet application development. The need first to measure what requires control was established, leading to the study research goal of developing a special purpose taxonomy of spreadsheet application development for use in the quality assurance and control of spreadsheet projects.

CHAPTER TWO:

REVIEW OF RELATED LITERATURE

2.1. Outline of this Chapter

This chapter reviews the literature for articles of relevance to this study. Initially, the history of categorisation and mathematical taxonomy are briefly considered. This is followed by a discussion on clusters and models.

Some examples of the use of taxonomies in computer science are reported. Taxonomies with particular relevance to the general areas of end-user computing and software development environments are discussed, as are the more specific partial taxonomies of the spreadsheet development process. The chapter concludes with a justification for the selection of the spreadsheet development attributes that were used to evolve the special-purpose A.D.E. taxonomy, the subject of this study.

2.2. Literature Sources

Articles published in academic journals and books, computer magazines, the computer trade press, newspapers and material from unpublished masters dissertations and conference papers were used in the preparation of this review. To identify sources of these articles, searches were conducted of abstracts held on CDROM particularly ABI/INFORM, ERIC, C-DATA and MATHSCI. On-line searches of the American DIALOG (INSPEC, Microcomputer Index, Compendex Plus, Philosopher's Index and MATHSCI) and Australian STAIRS and URICA databases also yielded useful material. The bibliography lists of located articles, in turn helped locate further material. Articles were also found through the suggestions of colleagues and students, the library staff of Edith Cowan University, the American Information Office, the Australian Consumer's Association and several spreadsheet vendors.

2.3. Classification as a Human Endeavour

Everitt (1980, p. 3) quotes Linnaeus:

> All the real knowledge we possess, depends on methods by which we distinguish the similar from the dissimilar.

Classification is the important basis of much of our lives. We classify everything around us, often subconsciously. We continuously improve and revamp these classifications and on them we base our responses to the stimuli we receive.

Schiffman, Reynolds and Young note the assistance classification provides to understanding.

> The rate of increase of human understanding has depended on organising concepts that allow us to systemise and compress large amounts of data. Systematic classification generally precedes understanding. (1981, p. 3)

It is understandable therefore, that Classification is one of the oldest scientific pursuits. The first classifications or taxonomies categorised the natural environment, people, animals and plants and the occurrences that affected them such as disease.

As early as 3000 BC, the Egyptian Imhotep classified physical and behavioural disorders. The early Hindus classified people into six types based on gender, physical and behavioural characteristics. Hippocrates (460-377 BC) classified diseases according to fever and chronicity

The Greek philosopher and naturalist, Aristotle (384-322 BC) was the first to propose a comprehensive classification scheme for animals. This continued in use with only minor changes, for nearly 2,000 years. He first divided animals according to whether they had red blood or not. Subsequent subcategories where based on how the animal's young were produced, live, egg, pupa etc. Theophrastus, sometimes called the first ecologist, extended Aristotle's ideas and classified plants relating them to their habitat.

The Swedish naturalist, Professor of Botany at Uppsala University, Carolus Linnaeus (1707-1778), established classification principles that have been extended to modern taxonomies. In 1753 he published Species Plantarum, and five years later Systema Naturae. These books introduced a binomial system for the classification of plants and animals e.g. Homo sapiens.

Charles Darwin's The Origin of Species, first published in 1859, developed his theories of evolution based on natural selection and a scheme postulating hierarchical links between taxa. These theories stimulated advances in Biology particularly Palaeontology and Comparative Anatomy. They had a tremendous impact on religious thought and Sociology and influenced Karl Marx in his ideas about the class struggle. Mendelyev in the 1860s published the periodic table of the elements which influenced later work on underlying atomic structures. Both classifications have had a profound effect on the subsequent development of their own and many other disciplines.

The twentieth century has seen the extension of classification to non-biological entities. Hertzprung and Russell classified stars based on their surface temperature and light intensity. (Kaufman & Rousseeuw, 1990, p. 1) Archaeology serration studies in the first quarter of this century, and the more recent marketing classification into market segments consisting of customers with similar needs have continued this trend. (Kaufman & Rousseeuw, 1990, p. 2)

Taxonomies have also proved popular with educators. Bloom in consultation with a group of experts developed a taxonomy of educational objectives. (Bloom, Engelhart, Furst, Hill and Krathwol, 1956), Steinaker and Bell (1979) produced a Gestalt educational taxonomy extending beyond just the cognitive, psychomotor and affective domains. Biggs and Collis (1982) developed the SOLO taxonomy which assessed the quality of student's work retrospectively. These taxonomies have been used extensively in education in areas including curriculum planning, student assessment, teacher training, evaluation and in-service.

The earlier methods of devising classifications were subjective, relying on the perception and judgement of the researcher. The classifications produced were usually no more than three dimensional, so eye-brain judgement was satisfactory to identify the clusters. (Kaufman & Rousseeuw, 1990, p. 2) The relatively new discipline of mathematical taxonomy has formalised the development of classifications using mathematical algorithms rather than relying solely on the subjective opinion of the developer. Arabie, Douglas and Desararbo (1987), also promote mathematical clustering and go as far as to suggest in their monograph, their three only valid excuses, for relying on visual clustering:

a) the researcher has read an out-of-date book

b) computational laziness

c) a very large data-set

Subjective opinions should not be ignored entirely however. They still have an important part to play choosing the input to the Cluster Analysis process and interpreting the results.

Early Cluster Analysis

In 1894, K Pearson published the first paper related to numerical taxonomy: "Contributions to the Mathematical Theory of Evolution". In a follow-up paper in 1901, he defined statistical procedures for detecting clusters. The first mathematically based non heuristic algorithm was published in Colloquia Mathematicae 2 in 1951 by K. Florek, J. Perkal and their colleagues. The algorithm developed classifications using similarities and graph theoretic concepts.

The more formal and objective modern methods of numerical taxonomy are now in vogue. Kaufman and Rousseeuw acknowledge that Cluster Analysis is "a very young scientific discipline in vigorous development". (1990, p. 3)

They suggest that there are three driving forces behind this;

a) the need to classify data described in more than three dimensions

b) the advent of the computer

c) the objectivity standards of modern science.

The ready availability of desk-top number crunching computer power coupled with user-friendly software has made the algorithms of mathematical taxonomy readily accessible to researchers.

Since it was first published in 1984, the Journal of Classification has successfully promoted modern classification techniques, made them available to a much wider audience and given them an increased visibility and credibility. The International Federation of Classification Societies founded in 1985 has established the validity of Classification as a discipline.

Today, Mathematical or Numerical Taxonomy covers many techniques and methods including Q-analysis, R-analysis, typology, typological analysis, Cluster Analysis, botryology, grouping, clumping, automatic classification, numerical taxonomy and unsupervised pattern recognition.

Taxonomists now apply these principles to many diverse fields. Godehardt (1990, p. 28) lists applications in the fields of anthropology, archaeology, astronomy, biol ogy, business, chemistry, computer science, economics, engineering, geography, geology, information and library science, linguistics, marketing, medicine, political science, psychology, sociology and soil sciences.

The classifications derived using mathematical taxonomy have been used widely. They have established a frame-work for information storage and retrieval and simplified the understanding of the relationships between their members. Practitioners can now communicate in the sure knowledge that they are talking about the same thing. Taxonomies have also suggested hitherto unsuspected common properties of classified entities.

2.4. Clusters, Models and Reality

Clusters

What is a cluster? The first attempts at mathematically defining clusters were by graph theorists in the early fifties. Kaufman and Rousseeuw (1990, p. 3) report that there is still no generally accepted definition of a cluster. The composition of a cluster is very much an individual decision. The cluster is bound primarily in the eye of the beholder.

Romesburg stressed this view:

> A cluster is a set of one or more objects that we are willing to call similar to each other. It may seem strange to use the word 'willing' but that is exactly the right word. To call two or more objects similar, we must be willing to neglect some of the detail that makes them non-identical. We must be tolerant of some of their differences. (1984, p. 15)

A cluster is a group of similar entities. Entities within a cluster are similar to each other and dissimilar to entities in other clusters. Cluster analysis defined by Kaufman and Rousseeuw as "the art of finding groups in data" (1990, p. 1) seeks to identify clusters or groups within a data-set. Objects are placed in groups so that groups contain similar objects, and groups are as dissimilar from each other as possible i.e. objects are allocated to promote within group homogeneity and between group heterogeneity.

Cluster Analysis divides a multivariate data-set into groups or classes. The familiar criteria for 'good' structured design of computer programs include 'within module cohesion' and 'loose coupling between modules'. These criteria are similar to the 'intra-cluster homogeneity' and 'inter-cluster heterogeneity' criteria of Cluster Analysis i.e. internal cohesion and external isolation.

Groups or clusters can be compact i.e. spherical, globular or ellipsoidal. Compact clusters have each member more like all other members of the cluster than they are like those who are outside the cluster. Alternatively, the clusters can be extended, serpentine or chained. Each cluster member is more like at least one other member than any outside the cluster. Clusters can be well separated or close together.

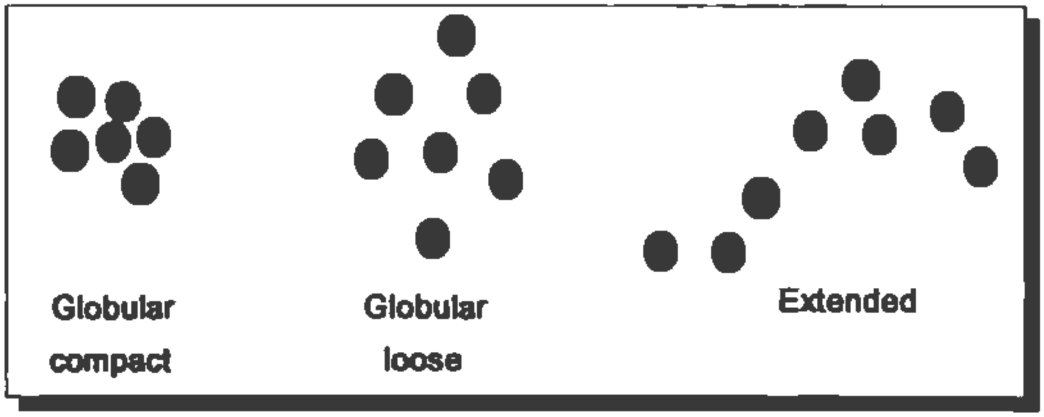


Figure 2.1: Three types of well separated clusters. Globular compact, globular loose, and extended.

Clusters can be overlapping or exclusive. Overlapping clusters allow an object to belong to two clusters. The concepts of Zadeh's fuzzy logic, conceptual clustering, probability clustering and some ideas expressed about language and categorisation by Lakoff (1987) explore the idea of introducing a probability function to model the likelihood of an object being placed in a particular cluster. This type of cluster has limited use in developing a taxonomy and will not be considered further.

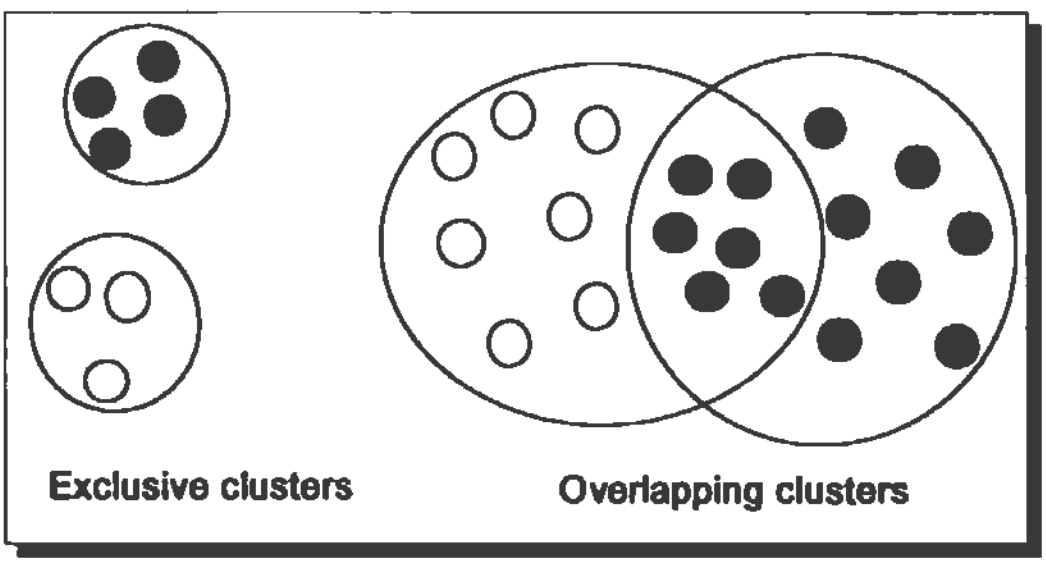


Figure 2.2: Exclusive and overlapping clusters.

Clustering criteria can be monothetic i.e. based on a single characteristic, or polythetic based on many characteristics. Polythetic exclusive clustering was the basis for the development of the A.D.E. taxonomy, the subject of this dissertation.

Models

Taxonomies are models of whatever they categorise, just as a map is a two dimensional representation of a three dimensional terrain.

Troy and Moawad (1982, p. 28) define a model as " a simplified representation of the behaviour (or structure) of a real system or process". Stopher and Meyburg (1979, p. 23) define a model as "an abstraction of reality" i.e. a simplified representation rather than a replica of reality. Godehardt (1990, p. 7) also considers a model as "the image of our understanding of reality". These authors suggest that a model should be valid, as accurate as possible and useful. They point out that it will never be perfect. It will always have errors due to incompleteness, biological variation and measurement inaccuracies. It will comply only within certain tolerance limits.

Godehardt (1990, p. 30) balanced the loss of precision and information in a model with the benefits of clearness and economy it provides. He differentiated between the quality of models. (Godehardt, 1990, p. 5) There are good models for technical systems which we well understand. There are poorer models for complex biological systems as there is so much available data that only some of it can be in use at any one time. During the abstraction process, some details are discarded to keep the model within manageable bounds. It follows that there can be many different valid models of the same reality.

Several authors illustrate this concept with a pack of playing cards. (Jackson, 1983) (Anderburg, 1973, p. 17) The fifty two cards in a pack could be modelled or clustered into groups:

- Four clusters of thirteen: Clubs, Diamonds, Hearts, Spades
- Thirteen clusters of four: aces, twos, threes etc.
- Two clusters of twenty six: red cards and black cards

- Two clusters of twenty six: major and minor suits
- Two clusters: twelve face cards and forty number cards
- Three clusters: Queen of Spades, thirteen Hearts, all other cards
- Twenty six clusters: matched pairs of the same rank and colour

All clusters are valid. All provide a good general model. A keen card player plays Patience with two packs of cards combined. One pack is ten percent wider than the other. The cards are old and the combined packs contain three twos of Diamonds and only one two of hearts. One of the Aces of Spades has the corner missing and is clearly recognisable even when face down. The packs have two jokers. All of the models above provide a useful representation of the reality of this pack of cards.

Which is the 'best' model? There is no absolute answer to this question. The answer depends on the use to which the cards will be put. Bridge, Poker, Rummy, Bezique, Pelmanism, Patience and Snap players would select different models. Criteria to establish the 'best' model will depend on its intended use.

The 'best' clustering is the one that is of most use for a pre-specified purpose. The taxonomist's task is to select the 'best' model for a specified purpose. This is not only a scientific endeavour but also an art. The decision has both objective and subjective elements. Godehardt summarised this:

> We can say:
> (a) Scientific modelling is an art
> (b) All models are wrong
> (c) Some models are better than other ones
> (d) Our task is to find the best ones (Godehardt, 1990, p. 6)

There is a need to evaluate the adequacy of a model to determine its validity within set parameters and whether it is the 'best' model for the specific circumstances where it will be used. These concepts are considered further in chapter 5.

2.5. Mathematical Taxonomy

Cluster Analysis is a method of exploratory data analysis. Its purpose is to uncover from the data, hitherto unknown phenomena and groupings. Cluster Analysis is very different from inferential or confirmatory statistics, which allows a decision between different models of the null hypothesis. (H_0 and H_1) Exploratory statistics is used to generate, rather than test models or hypotheses, hence its usefulness in developing a taxonomy. Unlike inferential statistics, the sample rather than the underlying population is the prime source of interest:

> Every researcher, however, must note that cluster analyses are very subjective even if we use 'objective' mathematical methods to outline the different groups. This holds since the resulting clusters depend not only on the computational procedure, but also on the choice of attributes to be measured. And since the researcher . . . decides on the basis of his or her personal knowledge which attributes and objects should be drawn from a sample, this choice may be biased. Therefore the results of a cluster analysis are chiefly valid for the specific sample only and we cannot generalise them to a larger population without careful inspection. (Godehardt, 1990, p. 24)

There is always a temptation to generalise the results of a Cluster Analysis from the sample to the underlying population. This was resisted in this study. Generalisation and extension would require the use of inferential statistics. To do this, the model would require validation with confirmatory statistics and new data collected on a probability based sample.

> Model validation on the basis of exploratory methods alone is impossible. The purpose of confirmatory statistics (together with careful experimental design) on the other hand, is to validate phenomena and hypothesis from investigations . . . Its aim is at least to keep the probability of wrong decisions as low as possible . . . This confirmation is necessary. At the same time, pure confirmation is not sufficient for progress . . . . Exploratory methods are indispensable for the advance of scientific research. (Godehardt, 1990, p. 16)

Cluster Analysis differs from Multi-dimensional scaling. The latter is also a procedure for finding groups in data, but produces an answer mapped to n dimensional space. Cluster Analysis is a dimensionless grouping procedure.

There are many different Cluster Analysis algorithms including:

a) Hierarchical, both agglomerative and divisive (Lorr, 1983, pp. 83 - 120) (Dunn and Everitt, 1982, p. 77), (Everitt, 1980, p. 32)
b) Optimisation / partitioning (Kaufman & Rousseeuw, 1990, p. 113), Kmeans (Hartigan, 1985), (MacQueen, 1967)
c) Density or mode seeking - Hill and Valley methods (Jackson, 1983, p. 171) TAXMAP method of Carmichael and Sneath (Everitt, 1980, p. 47)
d) Clumping (Everitt, 1980, p. 54)
e) Q Factor analysis (Everitt, 1980, p. 54)
f) Geometric methods including Graph theory (Lorr, 1983, p. 80) (Clifford and Stephenson, 1975, p. 123), Minimum spanning trees (Clifford and Stephenson, 1975, p. 123), (Diday and Simon, 1976, p. 66), Metroglyphs (Gordon, 1981, p. 81) and Principal Co-ordinates Analysis (Gordon, 1981, p. 83)
g) Q mode or R mode analysis (Gordon 1981, p. 82)
h) Principal coordinates analysis (Gordon 1981, p. 83)
i) Non metric multi-dimensional scaling (Gordon, 1981, p. 91)
j) Probabilistic clustering (Clifford & Stephenson, 1975, p. 118)
k) Fuzzy clustering (Gordon, 1981, p. 58)
l) Conceptual clustering (Michalski & Stepp, 1983 a and b)

This study used the first two of these algorithms; hierarchical and partitioning Kmeans. These two algorithms were chosen as they implemented different philosophies of cluster structure, and were readily available on a personal computer using SYSTAT software. Further details of these algorithms and their variable input parameters can be found in chapter 3.

Uses of Taxonomies

Taxonomies have been used to predict reaction to stimuli from the earliest times. Galen (129-199 AD) related a person's susceptibility to various diseases to nine temperamental types. Today, taxonomies are still used in this way.

Everitt (1980) describes some other uses of Cluster Analysis including;

- finding a true typology
- model fitting
- develop a taxonomy
- hypothesis testing
- data exploration and hypothesis generating (must test with new data)
- data reduction and simplification

Romesburg (1984) agrees with the above but splits the taxonomy development into the development of general and special purpose taxonomies and adds the further use of assisting planning and management.

- develop general taxonomy
- develop special purpose taxonomy
- assist planning and management

This study has as its primary research goals two of Romesburg's uses of Cluster Analysis i.e. assist planning and management and develop a special purpose taxonomy.

Romesburg also discusses the value of classification and taxonomies to the research process. (1984, p. 225) Taxonomies can act as a catalyst to memory and thinking. They become the building blocks for scientific theories. They assist in the discovery of inductive generalisations and the prediction of values of specific variables. They assist in the organisation and retrieval of objects and improve planning.

Kaufman and Rousseeuw (1990, p. 2) identify two common purposes of taxonomies. They are primarily used to identify a structure already present in data. They can also impose structure in a 'fair' way, where necessary, on almost homogeneous data, e.g. divide a country into telephone areas.

Romesburg (1984, p. 6) generalises the different motives for taxonomy usage in science, planning and engineering. Scientists are motivated by a curiosity to discover how nature works, they do not require this knowledge for the benefit of society. Scientists validate their models by agreement with experimental facts. Planners on the other hand are motivated by making the world materially better. This involves management decisions on the best way to achieve a goal. Planners validate their work on how well the implemented plan improves the human condition.

Taxonomies are of use to both scientists and planners. Scientists use taxonomies to improve their understanding of the subject under study and to communicate with other scientists. Planners use taxonomies to assist in the management, evaluation and control process. A taxonomy of the spreadsheet development process would support the goals of both scientists and planners.

2.6. Problems and Benefits of Cluster Analysis

Benefits of Cluster Analysis

Gordon (1981, p. 140) discusses the benefits of Cluster Analysis, the most significant being the reduction of a large volume of data to a summary of manageable size. The implementation of a Cluster Analysis procedure also forces a researcher to specify precisely, important factors in assessing the data. Once programmed, computers work without bias and the researcher's preconceived ideas are ignored unless programmed in explicitly, when they can be identified.

Problems of Cluster Analysis

Everitt (1980, p. 59) discussed a major problem of this discipline i.e. the lack of a universally recognisable definition of exactly what constitutes a cluster. Twelve years later, there are still many distinct but often vague definitions used by different authors. This situation does not promote scientific objectivity.

There is also the difficulty of deciding how many clusters are present in data or indeed if any clusters are present at all i.e. if the data is non-homogeneous. Cluster Analysis algorithms force clustering on data, i.e. they do not have a possibility of returning a result that no clustering exists. This point has been noted by many authors (Sneath & Sokal, 1973), (Everitt, 1980), (Romesburg, 1984).

The criteria for accepting or rejecting clustering solutions are also ill defined and usually depend on the subjective judgement of the practitioner.

Many clustering algorithms give hierarchical solutions. Hierarchical solutions have their own particular problems. It could be inappropriate to force a hierarchical structure on a particular data-set. Everitt (1980, p. 65) shows that in hierarchical clustering, there is no relocation of entities once they have been placed in a cluster. An element may be placed in the wrong branch early on upsetting the solution with no chance of a re-assignment. There is doubt also how many clusters are represented in a hierarchical solution. The researcher has to decide this by looking at the tree. In addition, use of the single linkage algorithm may cause chaining, a phenomenon described in Section 3.6.2.

Cluster Analysis does not automatically lead to a taxonomy. This still requires interpretation, skill and insight by the numerical taxonomist to select characters, coefficients of similarity and difference and clustering method:

> These methods (Cluster Analysis) are best seen as tools for data exploration rather than for production of a formal classification. . . . These conclusions however are not to be interpreted as criticisms of numerical methods but are merely intended to imply that one cannot replace careful thought by automated computerised methods. (Dunn & Everitt 1982, p. 105)

2.7. Software Engineering Taxonomies

The field of Computer Science has its own models and taxonomies. The activity of programming involves the preparation of an abstract and general model of reality, and then its particular implementation. All possible values of variables, all relevant objects and all possible environmental situations have to be considered. Taxonomies can prove useful to computer scientists.

In many respects, spreadsheet development (by whatever name - application, template or worksheet) is similar to the development of other software applications. Both can be described by attributes such as size, complexity, developer expertise, development time and software used. Developer characteristics are the major source of difference between spreadsheets and other software. Spreadsheets are usually developed by end-users, who are not computer professionals and often work outside the direct control of DP departments. Kee notes that "spreadsheet templates are typically developed by managers with limited knowledge of standards or the consequences of not applying them" (1988, p. 55).

2.8. Selection of Spreadsheet Attributes for use in Cluster Analysis

Selection of spreadsheet attributes for input to the Cluster Analysis process was based on attributes mentioned in the published software engineering taxonomies reviewed below. Attributes used to distinguish between membership of categories in the various taxonomies of end-users, software applications, development environments, software usage and criticality, were drawn from the reports of many different authors.

2.9. Categorisations of Relevance to the Spreadsheet Development Process

Many authors have described taxonomies and categorisations of relevance to software application development. In this literature review, emphasis is placed on those categorisations that can be used to describe general end-user computing or spreadsheet development. Chapter 3 describes how some of the variables described in these taxonomies were used to derive the A.D.E. taxonomy of spreadsheet applications development.

2.9.1. End-Users

Several authors have proposed taxonomies describing spreadsheet developers or more general end-users. Tucker (1987) took a simple view. He categorised people involved with spreadsheets as 'Builders', 'Users' and 'Readers'. 'Builders' create spreadsheets, 'Users' run spreadsheets and 'Readers' use their output. Frequently the 'Builder', 'User' and 'Reader' are the same person.

Rockart and Flannery (1983, p. 777) noted the CODASYL end-user facilities committee categorisation of end-users as 'Direct', 'Intermediate' and 'Indirect'. 'Direct' users work with terminals or PCs. 'Intermediate' users specify the information requirements for reports which they ultimately receive and 'Indirect' users use computers through others e.g. an airline passenger requesting a flight booking.

Rockart and Flannery (1983) cite Martin (1982) and McLean (1974), who expanded on the CODASYL committee definition of end-users. They further broke down 'Direct' users into:

a) DP professionals who write code for others

b) DP amateurs who write code for their own use

c) Non DP trained users who use code written by others

Rockart and Flannery (1983) stressed the diversity of end-users and defined their own taxonomy which was rearranged by Kasper and Cerveny (1985). Their categories of end-users included:

Supporter of end-users

a) Functional support personnel who work predominantly in their own functional areas while retaining a sophisticated supporting role to the end-user computing activities of their work-mates

b) End-user computing support personnel often in an Information Centre.

c) Professional DP programmers

End-user

a) Non programming end-users who use software provided by others

b) Command level end-users who can use the software well and generate unique reports and queries

c) End-user programmers who develop their own applications.

Cotterman and Kumar (1989, p. 9) further evolved this definition. They produced an end-user cube graphical taxonomy based on the ideas of morphological analysis as propounded by Zwicky (1967). They aggregated Rockart and Flannery's six classes of users into two: those who develop systems for use by others and those who develop systems only for their own use. They also categorised end-users in three dimensions, 'Operation', 'Development' and 'Control'. 'Operation' involves the running, 'Development' the creation, and 'Control' the authorisation of the application. They coded each dimension on a binary dichotomous scale leading to a categorisation such as (0,1,0) for an organisation or individual who did not operate or authorise an application but had the responsibility for developing it, i.e. Cotterman and Kumar's category of 'User-developer'. They used their cube to classify and assess end-user computing risks.

Other authors categorise developers by expertise. Shneiderman (1987) divided end-users into 'Novice', 'Knowledgeable intermittent users' and 'Frequent or Power users'. Page-Jones (1990) extended this categorisation. He developed his taxonomy primarily for use in categorising software engineering expertise but stressed that it had a much broader usage. It is pertinent to spreadsheet developers:

a) Innocent

b) Aware

c) Apprentice

d) Practitioner

e) Journeyman

f) Master

g) Expert

2.9.2. Application Areas

Spreadsheets are rather specialised software applications and accordingly there have been few reports in the literature covering the areas where they are used. Spreadsheets can be considered as a subset of decision support systems. Eom and Lee (1990, p. 68) surveyed journal articles about decision support systems published between 1971 and 1988. They categorised these by application area. Most applications (66%) were in the corporate financial management area. Their categories included:

a) Corporate financial management including accounting, auditing, finance, human resource management, international business, information systems, marketing and transportation and logistics, production and operations management, strategic management
b) Agriculture
c) Education
d) Government
e) Hospital and health care
f) Military
g) Natural resources
h) Urban and community planning
i) Miscellaneous

2.9.3. Application Function

Many authors have classified software by function. Such categorisations concentrate on the use of the application. General functional taxonomies have been developed for software applications. More restricted functional categorisations of decision support systems have been reported and there are some papers and articles which attempt a limited categorisation of spreadsheets from a functional perspective. Some of these classifications are general purpose but more often the classification has been developed with a specific purpose in mind.

Ballou and Pazer (1985, p. 1985) categorised information systems as either 'Transaction processing' or 'Model based decision support'. Spreadsheet applications can belong to either category. Prototyping is a common development methodology for spreadsheets. West (1986) developed a taxonomy of prototypes. His categories of 'Transaction system' and 'Decision support' were similar to those of Ballou and Pazer, with the additional category of 'Data integration' software. He extended his taxonomy to consider different implementation technologies and development environments.

Eom and Lee (1990) in their survey of published articles (1971 - 1988) on decision support systems, noted spreadsheets as one of the types of software used to develop decision support systems. They were concerned about the impact of decision support systems on decision making. They divided the applications in their survey into four kinds.

a) Deterministic models. Once the input is determined the output is assured.
b) Stochastic models involving a measure of probability about their outcome.
c) Forecasting and statistical models.
d) Other applications

Eom and Lee (1990) also considered the capacity of the output of a decision support system to influence a decision. They extended Alter's taxonomy to model this aspect of software applications. Alter's (1980) taxonomy as reviewed in Eom and Lee (1990) had the following categories:

a) File drawer systems - on-line access to a particular item
b) Data analysis systems - on-line data retrieval, manipulation and display
c) Analysis information systems - manipulate the internal data from transaction processing augmented with data from other sources
d) Accounting models - use balance sheets, estimate of income etc.
e) Representational models - estimate future consequences on variable parameters

f) Optimisation models - generate optimal solutions within a series of constraints

g) Suggestion models - leave no room for judgement

Fox published his well known software application taxonomy in 1982. He categorised the function of software in two dimensions: (Fox, 1982, p. 35)

a) Types: 'Application', 'Support' (programmer tools) or 'System' software

b) Classes: 'Product' or 'Project' (used to develop a Product).

Macro (1990, p. 71) added a third class of software to b) - the 'Prototype'. Using Fox's taxonomy, spreadsheets (applications, worksheets or templates) are 'Product', 'Application' software while the parent spreadsheet software is 'Support', 'Project' software. Frequently spreadsheet applications are 'Prototypes' that have migrated to become 'Products' without the checks and balances normally associated with software 'Products' developed by DP professionals.

Rockart and Flannery (1983, p. 779) surveyed end-user computing in seven large American and Canadian companies. Their survey covered all types of end-user computing and was not restricted to spreadsheets. 50% of the applications involved complex analysis, and a further 21% simple analysis or inquiry. Other types of systems developed involved report generation, operational systems and miscellaneous systems.

Schneider and Hines (1990) also classified software applications. Their classification was a special purpose taxonomy for medical software, developed to assist in ensuring patient safety. It was of particular interest to this study as it classified software applications from a control perspective. It considered all types of applications and control, and spreadsheets were not mentioned explicitly in their article. Schneider and Hines considered two aspects of medical software requiring control, 'Patient Safety' and 'Patient Vulnerability'. 'Patient Safety' involved protection from harm by a medical device. 'Patient Vulnerability' involved protection from indirect harm due to erroneous data entering a system.

Schneider and Hines' taxonomy was also three dimensional considering 'Function' (data or device driven), 'Mode' (actively change data or report only) and the concept of a 'Controlled or Uncontrolled environment'. They recommended points of control for each classification within their taxonomy. Their concept of environmental control was used in the development of the A.D.E. taxonomy and their suggestion of basing control on the application category within a taxonomy is considered further in chapter 6.

2.9.4. Application Criticality

A further aspect of the use of a software application is how critical it is to the organisation where it is developed. Weber (1986) considered the criticality of end-user developed systems. He gave suggestions on the assessment of criticality including:

a) Effect on the organisation should the system be withdrawn
b) Scope of effect of the system
c) Use of corporate data

Eom and Lee (1990) classified published articles on decision support systems by the level of management involvement: 'Strategic', 'Tactical' or 'Operational'. Their paper did not restrict itself to a discussion about spreadsheets but considered decision support systems in general. However their classification is also useful to categorise spreadsheets and would assist in giving an indication of how critical a spreadsheet is to an organisation.

Karten (1989) looked at spreadsheet applications from a control perspective and the criticality of the application to the organisation. Her classification of spreadsheet applications was restricted to those types she considered worthy of control:

a) Used for making business decisions especially financial that have a permanent and significant effect on the organisation
b) Users or creators of corporate data
c) Complex (logical or content)
d) Rushed development

e) Catastrophic consequences if in error

f) Developed in an organisation with a heightened sensitivity due to past experiences of errors

Eom and Lee (1990) considered task interdependency in their survey of articles on decision support systems. They were concerned about the sharing of data between decision makers and the impact a particular decision support task exerts on other tasks. They classified their surveyed decision support journal articles by task interdependency

a) Personal support only

b) Group support - using corporate data and relating to each other

c) Organisational support - creating corporate data

Rockart and Flannery (1983) also considered how critical end-user computer systems were to an organisation. They categorised the scope of systems as 'Personal', 'Single department' or 'Multi-departmental' and expressed surprise at the percentage of systems which were not confined to personal use (69%). They also categorised the frequency of use of the applications as 'Daily', 'Weekly', 'Monthly', 'As needed' and 'One-shot'. Their classifications were used to help identify suitable spreadsheet attributes for input to the clustering process. A comparison of the results of the survey of spreadsheet applications described in this dissertation with Rockart and Flannery's findings for general end-user computing, can be found in chapter 6.

2.9.5. Data

Data used in an application is a major contributor to its criticality. Rockart and Flannery (1983, p. 778) reported on the source of data used in their survey of end-user computing applications. Approximately one third was transferred electronically, a further third was keyed in and most of the remaining third was generated by the end-user.

Nesbit (1985, p 80) identified categories of data usage that can cause integrity problems:

a) Multiple purposes - same data used again
b) Mixed time frames - currency for one use may be different for another
c) Big categories small analysis - data aggregated so that useful data is no longer explicit
d) Misunderstood definitions
e) Corporate rather than private data

Buckland (1989, p. 196) distinguished between 'Public', 'Corporate' and 'Non-corporate' data (Private data). His categories considered data from the perspective of its source. 'Corporate' data was considered as either data that effected the finances of the company and was kept as part of its records or data on which routine management decisions were based. He considered 'Private' data to be either "transient or short lived" data or "data developed from analytical work without adequate controls" and 'Public' data as data from public sources. These concepts of data categorised by its source are relevant to spreadsheets and were used in the development of the A.D.E. taxonomy.

2.9.6. Program Implementation

Halstead (1977) was concerned with algorithms and their implementation. He was interested in algorithmic properties that could be measured directly or indirectly, statically or dynamically including 'Length', 'Program Level', 'Modularity', 'Purity' (lack of double negatives, aliases etc.), 'Size', 'Intelligence content' and 'Programming effort'. Fox (1982) also considered the three major attribute categories of software: 'Scale', 'Complexity' (subdivided into 'Technical' and 'Logical') and 'Clarity'. These properties have relevance for spreadsheets.

Lehman (1980) cited by Macro (1990, p. 74) classified programs according to their S, P, or E properties:

a) S - Specified formally

b) P - Problem oriented with an inexact formulation

c) E - Embedded in the real world so likely to change formulation

Macro (1990) extended this classification of programs to software and changed E to mean 'Evolvable'. Few spreadsheets belong to Lehman's category S. Most spreadsheets can be categorised as P with a few in category E. The prevalence of spreadsheet error reports in the literature, outlined in chapter 1, and the current extended spreadsheet usage in many organisations, promotes the case for more spreadsheets being developed in category S i.e. with formal specification (and control).

Other classifications according to program size and temporal properties ('Batch', 'On-line', 'Real-time') are given by Macro (1990).

2.9.7. Complexity

Macro (1990, p. 80) pointed out the "many faceted" nature of software complexity. He considered three aspects:

a) Complexity of Intention - software scope and requirements

b) Complexity of Interaction - dynamic software operation

c) Complexity of Implementation - design and programming

The remainder of this discussion is restricted to 'Complexity of implementation' as this has most bearing on spreadsheet development. This facet of software complexity is an attribute of the implementation of software rather than an attribute of its function or operation. Several different authors have defined aspects of software complexity (Fox, 1982), (Halstead, 1977), (Shneiderman, 1980), (Macro, 1990), (Gilb, 1977, p. 88).

However Macro reports that:

> There are no established and generally accepted metrics for measuring the complexity of a software system, although there is much research into this topic. Macro (1990, p. 86)

Shneiderman (1980) postulated three types of software complexity: 'Logical', 'Structural' and 'Psychological'. 'Logical' complexity was involved with measuring the number of possible paths through a program. He suggested measuring this using either the number of logical IF statements or McCabe's (1976) graph theoretic complexity metrics. Gilb (1977, p. 162) also discussed metrics for measuring 'Logical' complexity.

Shneiderman's (1980) 'Structural' complexity involved 'Absolute' and 'Relative' structural complexity. 'Absolute' was concerned with the number of modules and objects while 'Relative' was concerned about the coupling and links between them. 'Psychological' complexity was concerned with software characteristics that are difficult for humans to understand and had much in common with Macro's (1990) concept of 'Complexity of interaction'.

Meyer and Curley (1989) considered the complexity of computer applications with particular relevance to expert systems. They considered complexity in two parts: 'Knowledge' and 'Technology' complexity. 'Knowledge' complexity was concerned with measuring the domain and information characteristics of the expert system, i.e. the complexity of content. 'Technology' complexity was concerned with the implementation of the system i.e. hardware platforms, programming effort, database and networking.

Miller (1989) discussed the complexity afforded by linking worksheets. He discussed modularisation and linkage within a worksheet, one time consolidation of worksheets, multiple open worksheets linked e.g. Windows D.D.E., three dimensional spreadsheets and multi-dimensional databases.

Based on these ideas about the complexity of general software applications, spreadsheet complexity will be considered in terms of:

a) Design complexity - worksheet layout
b) Formula complexity - functions and formulas used
c) Link complexity - structural links to other entities
d) Logical complexity - number of options in the spreadsheet, controlled by logical IF and LOOKUP functions.

2.9.8. Software Development Environments

Macro (1990, p. 64) defined four paradigms of application development: 'Computation', 'Data-processing', 'Process-oriented' and 'Rule-based'. The 'Computational' paradigm involves complex calculations and differs from the 'Data-processing' paradigm which involves heavy volume simple transaction processing. 'Process oriented' involves calculation in real-time and 'Rule-based' incorporates the artificial intelligence principles of heuristic adaption and the ability to learn.

Sommerville (1985, p. 381) categorised software development environments as:

a) Programming language independent, best used for small systems
b) Programming language specific, used for exploratory programming and prototyping
c) Software Engineering - IPSEs (integrated project support environments)

When considering spreadsheet security, integrity and quality assurance, it is insufficient to consider development environments solely in terms of the software used. Account needs to be taken of the people and procedures involved (as in Sommerville's IPSE), i.e. not just the programming but also the whole software development project.

Dart, Ellison, Feiler and Haberman (1987) of Carnegie Mellon University considered this when they produced a taxonomy of software development environments. They differentiated between 'programming' and 'software development' environments. The former consisting of 'programming in the small' i.e. coding, compilation etc. and the latter a combination of 'programming in the large' and 'programming in the many' i.e. extending into areas such as configuration and project management. Their taxonomy considered basic operating facilities such as memory and data, and state of the art enhanced functionality, such as browsers, windowing and multi-tasking.

Their taxonomy had four categories:

a) Language centred environments - one language only, highly interactive with poor support for programming in the large
b) Structure oriented environments - tools for direct manipulation of structures, language independent generators
c) Toolkit environments - including support for programming in the large activities. No environmental controls
d) Method based environments - support programming in the large and programming in the many, design methodologies etc.

Spreadsheets were not referred to explicitly in this paper, but have aspects of language centred and structured oriented environments.

Perry and Kaiser (1991) produced a general three dimensional model of software development environments looking at 'Structures', 'Mechanisms' and 'Policies'. They placed this in a sociological metaphor of 'State', 'City', 'Family' and 'Individual'. 'Structures' are objects that represent the software under development. 'Mechanisms' are the languages and tools involved. 'Policies' are user requirements that are imposed during the development process. They compared their taxonomy to that of Dart et al. Their concept of policies is pertinent to the control of spreadsheet development.

Schmitt (1988) developed a partial taxonomy of end-user development environments which is also relevant to spreadsheets.

a) Basic, used for decision making within a department. No DP data provided. Application within the scope of the normal functional job of the developer.

b) Sophisticated end-user. Corporate data downloaded from the main-frame and used locally.

c) Distributed programming. Developed for others to run.

2.9.9. Spreadsheet Categorisations

Several partial categorisations of aspects of the spreadsheet application development process have been published.

Moskowitz (1987b. p. 51) categorised spreadsheet templates in the popular computer press primarily by whether the developer was a computer professional:

a) Large templates prepared by programmers usually debugged and validated with care.

b) End user error-prone templates, often adapted by others with no real understanding of the underlying constraints.

Anderson and Bernard (1988) and Ronen, Palley and Lucas (1989) examined types of spreadsheet application. Creeth (1985, p. 92) looked at the type of models he considered were suitable for spreadsheet implementation concluding that accounting packages or financial modelling packages were often the more appropriate tool. Creeth felt that spreadsheets should only be used for very simple models:

a) Models that are solely used by their developer

b) Models that may be used by others but are unlikely ever to require formula changes

c) Models that will seldom be updated

Hassinen, Sajaniemi and Väisänen (1988) reviewed more than one hundred spreadsheets in use in Finnish government and industry and produced a taxonomy of spreadsheet physical and logical data structures.

Anderson and Bernard (1988, p. 42) categorised spreadsheets from an accountant's perspective with the required documentation and controls in mind.

a) Simple spreadsheets developed for and by the same person.

b) Complex spreadsheets developed for and by the same person.

c) Spreadsheet created for another user.

Ronen, Palley and Lucas (1989, p. 87) categorised spreadsheet models in a similar way, but focused on the model reusability as well as whether the developer was also the user of the model.

a) Developer is the user too. One shot throwaway model.

b) Developer is the user too but frequent model runs.

c) Developer not the formal user.

They also categorised spreadsheet applications in terms of information systems as:

a) Transaction processing.

b) Management Information Systems.

c) Decision Support Systems - personal use only.

d) Decision Support Systems designed for others.

Their class d) further considered models designed for few or many users, the expertise of the user and the number of times the model was run.

This review of the literature did not identify a complete taxonomy of all aspects of the spreadsheet development process. The most suitable categorisation pertinent to spreadsheets, was provided by Rockart and Flannery's (1983) extensive taxonomy of end-user computing. A comparison of the A.D.E. taxonomy of spreadsheet application development with Rockart and Flannery's taxonomy was used to validate the former and can be found in section 5.5.5 and chapter 6.

Rockart and Flannery classified end-user applications in several dimensions:

a) By primary purpose e.g. reports, operational systems
b) By systems scope - multi or single department, personal
c) By primary source of data
d) By who developed them
e) By who uses them
f) By frequency of use
g) By inclusion of graphics

Ronen, Palley and Lucas (1989) and Anderson and Bernard (1988) went one step further, suggesting appropriate design and control criteria could be developed for different spreadsheet categories.

There is a need for a more extensive yet generalised spreadsheet application taxonomy to enable comparisons of the design and control recommendations proposed by different authors. Cotterman and Kumar (1989), the developers of an end-user taxonomy, justify its use by pointing out the dangers of comparing research results where groups have not been fitted into such a taxonomy. They used their taxonomy to assess risk caused by end-users. The same point can be made to support the development of a taxonomy of spreadsheet applications. Chapter 6 includes a discussion on how such a taxonomy, with a checklist of matching design and control criteria, could assist a spreadsheet application developer in building worksheets with the appropriate security and integrity controls.

2.10. Summary of this Chapter

This chapter discussed some reports in the literature of relevance to developing a special purpose taxonomy of spreadsheet application development. The concepts of the representation of reality with different models, and the criteria for choosing the 'best' model were considered. A brief history of classification and numerical taxonomy was developed. Finally the literature was reviewed for categorisations and taxonomies of the spreadsheet development process and allied activities.

CHAPTER 3: STUDY METHODOLOGY AND DESIGN

3.1. Outline of this Chapter

This chapter sets out the rationale behind this study and its design in sufficient detail to allow its replication by others. Initially, the study is framed by the goals of the research. A survey of spreadsheet application development and the subsequent exploratory data analyses are described, leading to the construction of a taxonomy of spreadsheet applications development and its diagnostic key. The chapter concludes with a discussion of ethical considerations.

3.2. Framing of the Study

This study was framed by the primary research goal of the development and validation of a special purpose taxonomy of spreadsheet application development. The A.D.E. (Application, Developer, Environment) taxonomy was evolved for use in categorising spreadsheet application development projects.

In a future study, a 'Spreadsheet Control Model' will be developed. A spreadsheet development project's category within the A.D.E. taxonomy could then be input into the control model to ascertain appropriate spreadsheet design and control measures. Thus the long-term research goal of providing assistance for the planning and management of spreadsheet application development, also contributed to the framing of this current study.

The selection of the spreadsheet attributes used to develop the taxonomy was framed by the taxonomy's proposed use for suggesting spreadsheet design and control measures. The cases selected for input to mathematical clustering procedures were selected on the basis that they showed sufficient variation to contribute to a taxonomy well representative of the population.

The secondary research goals of developing a useful taxonomy with well structured and intuitive clusters framed the criteria for acceptability of clustering solutions as a basis for the A.D.E. taxonomy.

3.3. Outline of the Research Methods

An analytical survey of spreadsheet application development was conducted. Both qualitative and quantitative data were collected through a self administered questionnaire. Exploratory data analysis using multivariate statistical methods, primarily cluster analysis found groups within the data. These groups were analysed to find which spreadsheet attributes contributed most to the between group variability and within group cohesiveness. From this analysis, the A.D.E. (Application, Developer, Environment) taxonomy of spreadsheet application development was evolved. Validation of the taxonomy will be described in chapter 5.

3.4. Survey of Spreadsheet Application Development

3.4.1. Population

The population of interest to this study consisted of all incidences of spreadsheet application development in Australia. The size and variability of this population were unknown, however continuation of this study was justified as the research was largely exploratory in nature and its successful outcome would assist in the definition of the population variability.

3.4.2. Sample

Sampling Unit

The sampling unit consisted of one incidence of a spreadsheet developer developing a single spreadsheet application; i.e. a single spreadsheet development project.

Sampling Frame

A sampling frame can be defined as "A basic list or reference that unambiguously defines every element or unit in the population from which the sample is to be taken." (Stopher and Meyburg, 1979, p. 12) The lack of availability of a complete sampling frame posed this study's major difficulty. Unsuccessful approaches to identify such a frame were made to: a) Edith Cowan University Libraries, b) Australian Bureau of Statistics, c) Spreadsheet Vendors, d) Australian Consumers Association, e) the Australian Computer Society and f) the national computer press including the Computer Section of 'The Australian' newspaper.

If a suitable frame had been available, its currency could have been suspect and it would probably have suffered from defects of inaccuracy, inadequacy and incompleteness. Frames of subsets of the population of spreadsheet developers were constructed and used in the stratified sampling procedures outlined below.

Sampling Plan

As a complete sampling frame was unavailable, commonly used probability based sampling designs, such as those shown below, were unsuitable. (Stopher and Meyburg, 1979, p. 21-22), (Davis and Cosenza, 1985, p. 215-227):

a) Random sampling

b) Stratified Random Sampling with use of a variable sampling fraction

c) Multistage sampling

d) Cluster sampling

The evolution of a useful and representative taxonomy of spreadsheet application development, required a sample which included a wide range of spreadsheet development projects. Inclusion of as much of the variability of the population as possible, even small groups, was mandatory. To ensure this outcome, compromise subjective sampling decisions were taken.

f) Optimisation models - generate optimal solutions within a series of constraints

g) Suggestion models - leave no room for judgement

Fox published his well known software application taxonomy in 1982. He categorised the function of software in two dimensions: (Fox, 1982, p. 35)

a) Types: 'Application', 'Support' (programmer tools) or 'System' software

b) Classes: 'Product' or 'Project' (used to develop a Product).

Macro (1990, p. 71) added a third class of software to b) - the 'Prototype'. Using Fox's taxonomy, spreadsheets (applications, worksheets or templates) are 'Product', 'Application' software while the parent spreadsheet software is 'Support', 'Project' software. Frequently spreadsheet applications are 'Prototypes' that have migrated to become 'Products' without the checks and balances normally associated with software 'Products' developed by DP professionals.

Rockart and Flannery (1983, p. 779) surveyed end-user computing in seven large American and Canadian companies. Their survey covered all types of end-user computing and was not restricted to spreadsheets. 50% of the applications involved complex analysis, and a further 21% simple analysis or inquiry. Other types of systems developed involved report generation, operational systems and miscellaneous systems.

Schneider and Hines (1990) also classified software applications. Their classification was a special purpose taxonomy for medical software, developed to assist in ensuring patient safety. It was of particular interest to this study as it classified software applications from a control perspective. It considered all types of applications and control, and spreadsheets were not mentioned explicitly in their article. Schneider and Hines considered two aspects of medical software requiring control, 'Patient Safety' and 'Patient Vulnerability'. 'Patient Safety' involved protection from harm by a medical device. 'Patient Vulnerability' involved protection from indirect harm due to erroneous data entering a system.

Schneider and Hines' taxonomy was also three dimensional considering 'Function' (data or device driven), 'Mode' (actively change data or report only) and the concept of a 'Controlled or Uncontrolled environment'. They recommended points of control for each classification within their taxonomy. Their concept of environmental control was used in the development of the A.D.E. taxonomy and their suggestion of basing control on the application category within a taxonomy is considered further in chapter 6.

2.9.4. Application Criticality

A further aspect of the use of a software application is how critical it is to the organisation where it is developed. Weber (1986) considered the criticality of end-user developed systems. He gave suggestions on the assessment of criticality including:

a) Effect on the organisation should the system be withdrawn

b) Scope of effect of the system

c) Use of corporate data

Eom and Lee (1990) classified published articles on decision support systems by the level of management involvement: 'Strategic', 'Tactical' or 'Operational'. Their paper did not restrict itself to a discussion about spreadsheets but considered decision support systems in general. However their classification is also useful to categorise spreadsheets and would assist in giving an indication of how critical a spreadsheet is to an organisation.

Karten (1989) looked at spreadsheet applications from a control perspective and the criticality of the application to the organisation. Her classification of spreadsheet applications was restricted to those types she considered worthy of control:

a) Used for making business decisions especially financial that have a permanent and significant effect on the organisation

b) Users or creators of corporate data

c) Complex (logical or content)

d) Rushed development

e) Catastrophic consequences if in error

f) Developed in an organisation with a heightened sensitivity due to past experiences of errors

Eom and Lee (1990) considered task interdependency in their survey of articles on decision support systems. They were concerned about the sharing of data between decision makers and the impact a particular decision support task exerts on other tasks. They classified their surveyed decision support journal articles by task interdependency

a) Personal support only

b) Group support - using corporate data and relating to each other

c) Organisational support - creating corporate data

Rockart and Flannery (1983) also considered how critical end-user computer systems were to an organisation. They categorised the scope of systems as 'Personal', 'Single department' or 'Multi-departmental' and expressed surprise at the percentage of systems which were not confined to personal use (69%). They also categorised the frequency of use of the applications as 'Daily', 'Weekly', 'Monthly', 'As needed' and 'One-shot'. Their classifications were used to help identify suitable spreadsheet attributes for input to the clustering process. A comparison of the results of the survey of spreadsheet applications described in this dissertation with Rockart and Flannery's findings for general end-user computing, can be found in chapter 6.

2.9.5. Data

Data used in an application is a major contributor to its criticality. Rockart and Flannery (1983, p. 778) reported on the source of data used in their survey of end-user computing applications. Approximately one third was transferred electronically, a further third was keyed in and most of the remaining third was generated by the end-user.

Nesbit (1985, p 80) identified categories of data usage that can cause integrity problems:

a) Multiple purposes - same data used again
b) Mixed time frames - currency for one use may be different for another
c) Big categories small analysis - data aggregated so that useful data is no longer explicit
d) Misunderstood definitions
e) Corporate rather than private data

Buckland (1989, p. 196) distinguished between 'Public', 'Corporate' and 'Non-corporate' data (Private data). His categories considered data from the perspective of its source. 'Corporate' data was considered as either data that effected the finances of the company and was kept as part of its records or data on which routine management decisions were based. He considered 'Private' data to be either "transient or short lived" data or "data developed from analytical work without adequate controls" and 'Public' data as data from public sources. These concepts of data categorised by its source are relevant to spreadsheets and were used in the development of the A.D.E. taxonomy.

2.9.6. Program Implementation

Halstead (1977) was concerned with algorithms and their implementation. He was interested in algorithmic properties that could be measured directly or indirectly, statically or dynamically including 'Length', 'Program Level', 'Modularity', 'Purity' (lack of double negatives, aliases etc.), 'Size', 'Intelligence content' and 'Programming effort'. Fox (1982) also considered the three major attribute categories of software: 'Scale', 'Complexity' (subdivided into 'Technical' and 'Logical') and 'Clarity'. These properties have relevance for spreadsheets.

Lehman (1980) cited by Macro (1990, p. 74) classified programs according to their S, P, or E properties:

a) S - Specified formally

b) P - Problem oriented with an inexact formulation

c) E - Embedded in the real world so likely to change formulation

Macro (1990) extended this classification of programs to software and changed E to mean 'Evolvable'. Few spreadsheets belong to Lehman's category S. Most spreadsheets can be categorised as P with a few in category E. The prevalence of spreadsheet error reports in the literature, outlined in chapter 1, and the current extended spreadsheet usage in many organisations, promotes the case for more spreadsheets being developed in category S i.e. with formal specification (and control).

Other classifications according to program size and temporal properties ('Batch', 'On-line', 'Real-time') are given by Macro (1990).

2.9.7. Complexity

Macro (1990, p. 80) pointed out the "many faceted" nature of sofiware complexity. He considered three aspects:

a) Complexity of Intention - software scope and requirements

b) Complexity of Interaction - dynamic software operation

c) Complexity of Implementation - design and programming

The remainder of this discussion is restricted to 'Complexity of implementation' as this has most bearing on spreadsheet development. This facet of software complexity is an attribute of the implementation of software rather than an attribute of its function or operation. Several different authors have defined aspects of software complexity (Fox, 1982), (Halstead, 1977), (Shneiderman, 1980), (Macro, 1990), (Gilb, 1977, p. 88).

However Macro reports that:

> There are no established and generally accepted metrics for measuring the complexity of a software system, although there is much research into this topic. Macro (1990, p. 86)

Shneiderman (1980) postulated three types of software complexity: 'Logical', 'Structural' and 'Psychological'. 'Logical' complexity was involved with measuring the number of possible paths through a program. He suggested measuring this using either the number of logical IF statements or McCabe's (1976) graph theoretic complexity metrics. Gilb (1977, p. 162) also discussed metrics for measuring 'Logical' complexity.

Shneiderman's (1980) 'Structural' complexity involved 'Absolute' and 'Relative' structural complexity. 'Absolute' was concerned with the number of modules and objects while 'Relative' was concerned about the coupling and links between them. 'Psychological' complexity was concerned with software characteristics that are difficult for humans to understand and had much in common with Macro's (1990) concept of 'Complexity of interaction'.

Meyer and Curley (1989) considered the complexity of computer applications with particular relevance to expert systems. They considered complexity in two parts: 'Knowledge' and 'Technology' complexity. 'Knowledge' complexity was concerned with measuring the domain and information characteristics of the expert system, i.e. the complexity of content. 'Technology' complexity was concerned with the implementation of the system i.e. hardware platforms, programming effort, database and networking.

Miller (1989) discussed the complexity afforded by linking worksheets. He discussed modularisation and linkage within a worksheet, one time consolidation of worksheets, multiple open worksheets linked e.g. Windows D.D.E., three dimensional spreadsheets and multi-dimensional databases.

Based on these ideas about the complexity of general software applications, spreadsheet complexity will be considered in terms of:

a) Design complexity - worksheet layout
b) Formula complexity - functions and formulas used
c) Link complexity - structural links to other entities
d) Logical complexity - number of options in the spreadsheet, controlled by logical IF and LOOKUP functions.

2.9.8. Software Development Environments

Macro (1990, p. 64) defined four paradigms of application development: 'Computation', 'Data-processing', 'Process-oriented' and 'Rule-based'. The 'Computational' paradigm involves complex calculations and differs from the 'Data-processing' paradigm which involves heavy volume simple transaction processing. 'Process oriented' involves calculation in real-time and 'Rule-based' incorporates the artificial intelligence principles of heuristic adaption and the ability to learn.

Sommerville (1985, p. 381) categorised software development environments as:

a) Programming language independent, best used for small systems
b) Programming language specific, used for exploratory programming and prototyping
c) Software Engineering - IPSEs (integrated project support environments)

When considering spreadsheet security, integrity and quality assurance, it is insufficient to consider development environments solely in terms of the software used. Account needs to be taken of the people and procedures involved (as in Sommerville's IPSE), i.e. not just the programming but also the whole software development project.

Dart, Ellison, Feiler and Haberman (1987) of Carnegie Mellon University considered this when they produced a taxonomy of software development environments. They differentiated between 'programming' and 'software development' environments. The former consisting of 'programming in the small' i.e. coding, compilation etc. and the latter a combination of 'programming in the large' and 'programming in the many' i.e. extending into areas such as configuration and project management. Their taxonomy considered basic operating facilities such as memory and data, and state of the art enhanced functionality, such as browsers, windowing and multi-tasking.

Their taxonomy had four categories:

a) Language centred environments - one language only, highly interactive with poor support for programming in the large
b) Structure oriented environments - tools for direct manipulation of structures, language independent generators
c) Toolkit environments - including support for programming in the large activities. No environmental controls
d) Method based environments - support programming in the large and programming in the many, design methodologies etc.

Spreadsheets were not referred to explicitly in this paper, but have aspects of language centred and structured oriented environments.

Perry and Kaiser (1991) produced a general three dimensional model of software development environments looking at 'Structures', 'Mechanisms' and 'Policies'. They placed this in a sociological metaphor of 'State', 'City', 'Family' and 'Individual'. 'Structures' are objects that represent the software under development. 'Mechanisms' are the languages and tools involved. 'Policies' are user requirements that are imposed during the development process. They compared their taxonomy to that of Dart et al. Their concept of policies is pertinent to the control of spreadsheet development.

Schmitt (1988) developed a partial taxonomy of end-user development environments which is also relevant to spreadsheets.

a) Basic, used for decision making within a department. No DP data provided. Application within the scope of the normal functional job of the developer.

b) Sophisticated end-user. Corporate data downloaded from the main-frame and used locally.

c) Distributed programming. Developed for others to run.

2.9.9. Spreadsheet Categorisations

Several partial categorisations of aspects of the spreadsheet application development process have been published.

Moskowitz (1987b. p. 51) categorised spreadsheet templates in the popular computer press primarily by whether the developer was a computer professional:

a) Large templates prepared by programmers usually debugged and validated with care.

b) End user error-prone templates, often adapted by others with no real understanding of the underlying constraints.

Anderson and Bernard (1988) and Ronen, Palley and Lucas (1989) examined types of spreadsheet application. Creeth (1985, p. 92) looked at the type of models he considered were suitable for spreadsheet implementation concluding that accounting packages or financial modelling packages were often the more appropriate tool. Creeth felt that spreadsheets should only be used for very simple models:

a) Models that are solely used by their developer

b) Models that may be used by others but are unlikely ever to require formula changes

c) Models that will seldom be updated

Hassinen, Sajaniemi and Väisänen (1988) reviewed more than one hundred spreadsheets in use in Finnish government and industry and produced a taxonomy of spreadsheet physical and logical data structures.

Anderson and Bernard (1988, p. 42) categorised spreadsheets from an accountant's perspective with the required documentation and controls in mind.

a) Simple spreadsheets developed for and by the same person.

b) Complex spreadsheets developed for and by the same person.

c) Spreadsheet created for another user.

Ronen, Palley and Lucas (1989, p. 87) categorised spreadsheet models in a similar way, but focused on the model reusability as well as whether the developer was also the user of the model.

a) Developer is the user too. One shot throwaway model.

b) Developer is the user too but frequent model runs.

c) Developer not the formal user.

They also categorised spreadsheet applications in terms of information systems as:

a) Transaction processing.

b) Management Information Systems.

c) Decision Support Systems - personal use only.

d) Decision Support Systems designed for others.

Their class d) further considered models designed for few or many users, the expertise of the user and the number of times the model was run.

This review of the literature did not identify a complete taxonomy of all aspects of the spreadsheet development process. The most suitable categorisation pertinent to spreadsheets, was provided by Rockart and Flannery's (1983) extensive taxonomy of end-user computing. A comparison of the A.D.E. taxonomy of spreadsheet application development with Rockart and Flannery's taxonomy was used to validate the former and can be found in section 5.5.5 and chapter 6.

Rockart and Flannery classified end-user applications in several dimensions:

a) By primary purpose e.g. reports, operational systems
b) By systems scope - multi or single department, personal
c) By primary source of data
d) By who developed them
e) By who uses them
f) By frequency of use
g) By inclusion of graphics

Ronen, Palley and Lucas (1989) and Anderson and Bernard (1988) went one step further, suggesting appropriate design and control criteria could be developed for different spreadsheet categories.

There is a need for a more extensive yet generalised spreadsheet application taxonomy to enable comparisons of the design and control recommendations proposed by different authors. Cotterman and Kumar (1989), the developers of an end-user taxonomy, justify its use by pointing out the dangers of comparing research results where groups have not been fitted into such a taxonomy. They used their taxonomy to assess risk caused by end-users. The same point can be made to support the development of a taxonomy of spreadsheet applications. Chapter 6 includes a discussion on how such a taxonomy, with a checklist of matching design and control criteria, could assist a spreadsheet application developer in building worksheets with the appropriate security and integrity controls.

2.10. Summary of this Chapter

This chapter discussed some reports in the literature of relevance to developing a special purpose taxonomy of spreadsheet application development. The concepts of the representation of reality with different models, and the criteria for choosing the 'best' model were considered. A brief history of classification and numerical taxonomy was developed. Finally the literature was reviewed for categorisations and taxonomies of the spreadsheet development process and allied activities.

CHAPTER 3: STUDY METHODOLOGY AND DESIGN

3.1. Outline of this Chapter

This chapter sets out the rationale behind this study and its design in sufficient detail to allow its replication by others. Initially, the study is framed by the goals of the research. A survey of spreadsheet application development and the subsequent exploratory data analyses are described, leading to the construction of a taxonomy of spreadsheet applications development and its diagnostic key. The chapter concludes with a discussion of ethical considerations.

3.2. Framing of the Study

This study was framed by the primary research goal of the development and validation of a special purpose taxonomy of spreadsheet application development. The A.D.E. (Application, Developer, Environment) taxonomy was evolved for use in categorising spreadsheet application development projects.

In a future study, a 'Spreadsheet Control Model' will be developed. A spreadsheet development project's category within the A.D.E. taxonomy could then be input into the control model to ascertain appropriate spreadsheet design and control measures. Thus the long-term research goal of providing assistance for the planning and management of spreadsheet application development, also contributed to the framing of this current study.

The selection of the spreadsheet attributes used to develop the taxonomy was framed by the taxonomy's proposed use for suggesting spreadsheet design and control measures. The cases selected for input to mathematical clustering procedures were selected on the basis that they showed sufficient variation to contribute to a taxonomy well representative of the population.

The secondary research goals of developing a useful taxonomy with well structured and intuitive clusters framed the criteria for acceptability of clustering solutions as a basis for the A.D.E. taxonomy.

3.3. Outline of the Research Methods

An analytical survey of spreadsheet application development was conducted. Both qualitative and quantitative data were collected through a self administered questionnaire. Exploratory data analysis using multivariate statistical methods, primarily cluster analysis found groups within the data. These groups were analysed to find which spreadsheet attributes contributed most to the between group variability and within group cohesiveness. From this analysis, the A.D.E. (Application, Developer, Environment) taxonomy of spreadsheet application development was evolved. Validation of the taxonomy will be described in chapter 5.

3.4. Survey of Spreadsheet Application Development

3.4.1. Population

The population of interest to this study consisted of all incidences of spreadsheet application development in Australia. The size and variability of this population were unknown, however continuation of this study was justified as the research was largely exploratory in nature and its successful outcome would assist in the definition of the population variability.

3.4.2. Sample

Sampling Unit

The sampling unit consisted of one incidence of a spreadsheet developer developing a single spreadsheet application; i.e. a single spreadsheet development project.

Sampling Frame

A sampling frame can be defined as "A basic list or reference that unambiguously defines every element or unit in the population from which the sample is to be taken." (Stopher and Meyburg, 1979, p. 12) The lack of availability of a complete sampling frame posed this study's major difficulty. Unsuccessful approaches to identify such a frame were made to: a) Edith Cowan University Libraries, b) Australian Bureau of Statistics, c) Spreadsheet Vendors, d) Australian Consumers Association, e) the Australian Computer Society and f) the national computer press including the Computer Section of 'The Australian' newspaper.

If a suitable frame had been available, its currency could have been suspect and it would probably have suffered from defects of inaccuracy, inadequacy and incompleteness. Frames of subsets of the population of spreadsheet developers were constructed and used in the stratified sampling procedures outlined below.

Sampling Plan

As a complete sampling frame was unavailable, commonly used probability based sampling designs, such as those shown below, were unsuitable. (Stopher and Meyburg, 1979, p. 21-22), (Davis and Cosenza, 1985, p. 215-227):

a) Random sampling

b) Stratified Random Sampling with use of a variable sampling fraction

c) Multistage sampling

d) Cluster sampling

The evolution of a useful and representative taxonomy of spreadsheet application development, required a sample which included a wide range of spreadsheet development projects. Inclusion of as much of the variability of the population as possible, even small groups, was mandatory. To ensure this outcome, compromise subjective sampling decisions were taken.

A sample was drawn in three unequal parts, initially involving 250 incidences of spreadsheet application development. The sampling procedures used both probability and non-probability based sampling methods. Non-probability based aspects of the method as described by Davis and Cosenza (1985, p. 227) were used:

a) Judgement with quota samples - Quotas of groups of interest were subjectively set by the researcher.

b) Convenience - Chosen in a convenient way by the researcher.

c) Snowball - Used where the cases for analysis were hard to find and one sampled case suggested the names of other possibilities.

The non-random nature of this sample was justified in terms of feasibility. The lack of a sampling frame made random sampling impossible. Acknowledging the non-random nature of the sample, no attempt was made to generalise the findings. The research goal of developing a special purpose taxonomy of spreadsheet application development required the inclusion of representatives from all likely categories. This might not have been achieved with a random sample. The research was exploratory in nature, seeking to generate rather than confirm hypotheses. To generalise to the whole population, the findings would have to be confirmed by inferential statistical methods using a random probability based sample.

The target population was stratified into three unequal strata based on the geographical location of the spreadsheet developers, using the statistical subdivisions of the Australian Bureau of Statistics 1991 Census:

a) Preston Statistical Subdivision of the South West Statistical Division of Western Australia. - Aimed for high (80% +) coverage

b) Perth Statistical Division of Western Australia - Multistage stratified sampling.

c) South Australia, Victoria, New South Wales and Queensland - Selective sampling

Spreadsheet developers were drawn from each stratum, randomly where this was possible. Each developer was asked to provide a sampling unit by assessing a random example of their recent spreadsheet development activity.

Developers were asked to answer the questionnaire with respect to any recent sample of their work. This introduced some element of probability based selection within the strata. It was explicitly stated that there was no requirement as to size, complexity or importance of the spreadsheet development assessed. This still did not permit inference from the site to the target population, but did assist in fulfilling a need for objectivity as suggested by Kish (1987, p 51).

Preston Statistical Subdivision

This stratum was defined as spreadsheets developed in the Local Government Shires of Bunbury, Capel, Collie, Dardanup, Donnybrook-Balingup and Harvey. These shires had a combined population of 60,926 in the 1991 census.

The sampling design within this stratum required assessment of one spreadsheet from at least 80% of the developers in this restricted site. i.e. aim towards complete coverage of developers, with a random selection of spreadsheet from each. Kish (1987, p. 50) justifies the sampling of restricted research sites on the grounds of economics and feasibility. Stopher and Meyburg (1979, p. 109) state that "If no frame exists, the entire survey becomes a non-sample survey, designed both to collect the information for which the survey was originally intended and to set up a sampling frame". A sampling frame for the Preston stratum was constructed by seeking contact details of spreadsheet developers from all identifiable representatives in the site of:

a) Computer vendors and repair persons

b) Local, State and Commonwealth Government Departments

c) Mining companies

d) Staff and students of Edith Cowan University Bunbury Campus.

e) Staff of the South West College of TAFE, Collie and Harvey TAFE.

f) Staff of High Schools.

g) Accountancy, Finance, Law, Medicine and Engineering professional practices.

h) The Research Establishments of C.A.L.M. (Conservation and Land Management) and the Department of Agriculture.

i) Computer Hobbyist user groups.

j) Data Processing Professionals.

k) Bunbury, Collie and Harvey Chambers of Commerce.

Spreadsheet developers were sent a survey questionnaire, a letter of transmittal and a reply paid envelope. They were asked to respond within two weeks of receipt. In addition, selected respondents to the survey were asked to identify spreadsheet developer friends and colleagues who might not yet have been included. Reliance for a high coverage of developers was based on this 'snowball' effect, the initial extensive enquiries to set up the sampling frame, and the loyalty and interest of the local spreadsheet development community towards a research project initiated on their regional University Campus.

Non-response follow-up involved up to three telephone interviews at two weekly intervals until either the form was returned or the respondent gave notification of intention of non-response. It was originally intended to survey non-respondents for reasons for non-compliance in case this had introduced bias to the sample, but the high response rate made this unnecessary.

Justification of Choice of the Preston Stratum

The choice of this restricted site was justified on the grounds of convenience, economic necessity, the feasibility of developing a sampling frame (Kish, 1987, p. 50) and the view that the Preston Statistical Subdivision represented a microcosm of general Australian spreadsheet development practice. Due to the lack of a sampling frame, no attempt could be made to compare the spreadsheet development characteristics of Preston to those of Australia as a whole, however a comparison of the general characteristics of the populations of Preston and Australia was made using the 1986 Australian census statistics.

The graphs shown in Figures 3.1 to 3.4 below are based on these statistics and contrast Preston with all of Australia.

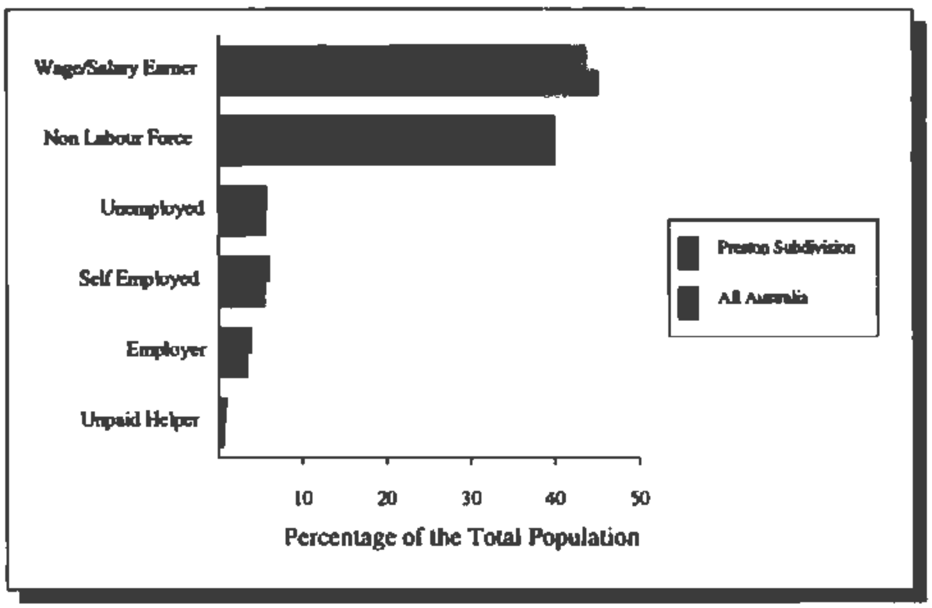


Figure 3.1 **Preston and Australia as a whole: Comparison of the Percentage of the Total Population by Employment Category. Adapted from the Australian Bureau of Statistics 1986 Census figures.**

The plot in Figure 3.1 is based on Table 46 of Appendix F. It shows a comparison between the employment categories of the whole population of Preston and of Australia as a whole. To the eye, they appear similar, however this similarity is not statistically significant as:

χ^2 calculated = 34. (critical χ^2 = 3.18842, $\alpha = 0.05$, 1 d.f.) and H_0 is rejected.

> H_0: There is no significant difference in the employment category distribution of the population of Preston and that of all of Australia.

i.e. when considering employment categories, Preston is significantly different from all of Australia.

The census figures were examined further to establish where Preston differed from all of Australia, so that the sampling procedures could take account of these differences.

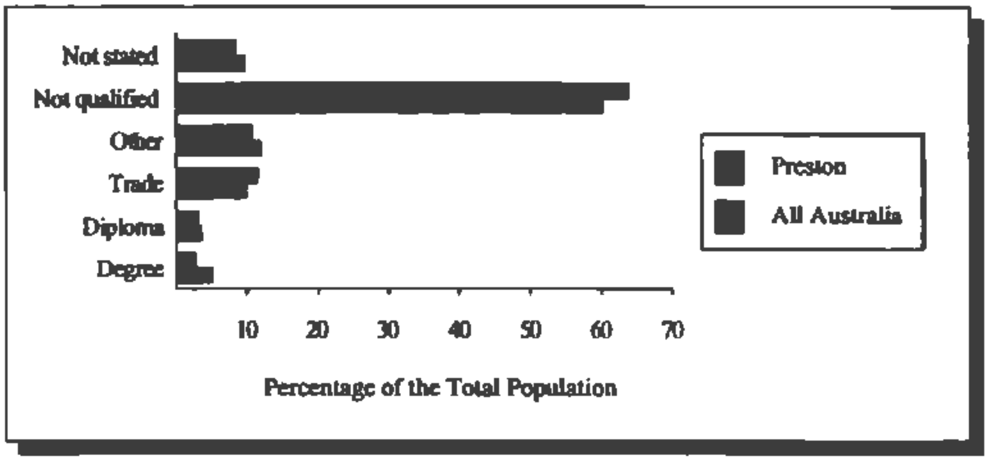


Figure 3.2 Preston and Australia as a whole: Comparison of the Percentage of the Total Workforce by Educational Qualification. Adapted from the Australian Bureau of Statistics 1986 Census figures

Figure 3.2 is based on Table 47 of Appendix F. It shows a comparison between the qualification distribution of the workforce in Preston and all of Australia. Again the similarity is not statistically significant with:

χ^2 calculated = 446. (critical χ^2 = 3.18842, α = 0.05, 1 d.f.) and H_0 is rejected.

> H_0: There is no significant difference in the educational qualifications of the workforce of Preston and that of all of Australia.

i.e. the educational qualifications of the Preston work-force are different from those of Australia as a whole. Preston has more people without qualifications and a smaller percentage of people with degrees or diplomas.

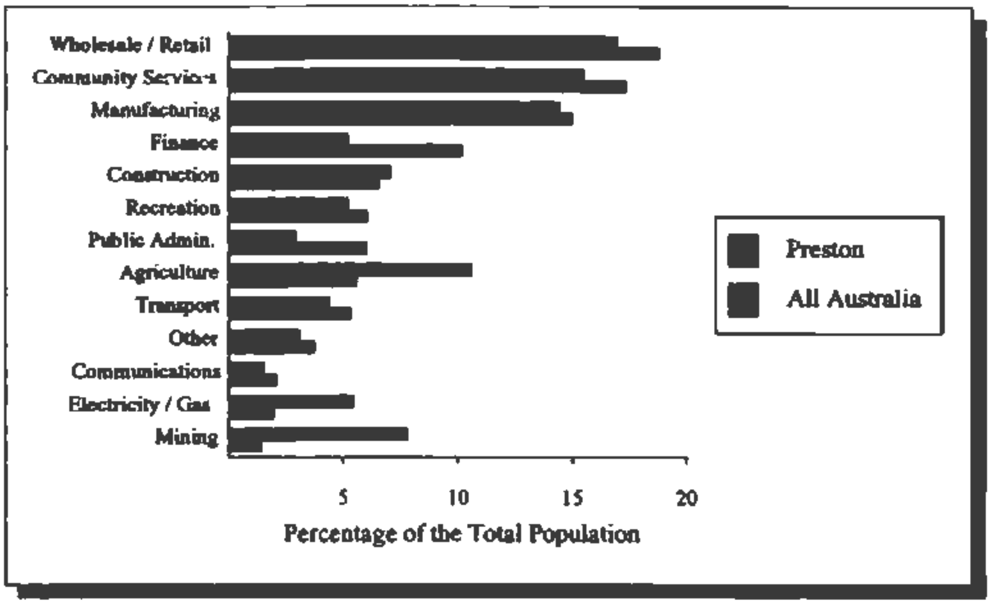


Figure 3.3 Preston and Australia as a whole: Comparison of the Percentage of the Total Workforce by Industry. Adapted from the Australian Bureau of Statistics 1986 Census figures.

Figure 3.3 compares the industry distribution of the Preston workforce with that of all of Australia. Preston has higher percentages employed in the agricultural, mining and gas and electricity industries, while it is low in those employed in public administration and finance.

These differences were considered to be important and were compensated for by selective sampling in the Perth Stratum, with the targeting of Finance and public administration workers.

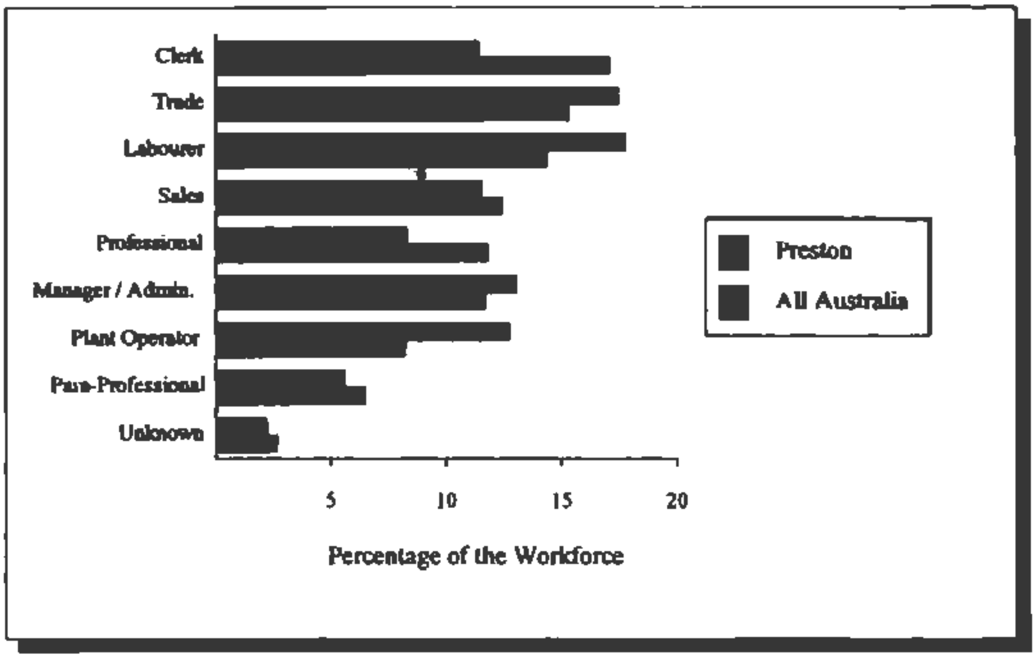


Figure 3.4 Preston and Australia as a whole: Comparison of the Percentage of the Total Workforce by Employment. Adapted from the Australian Bureau of Statistics 1986 Census figures.

Figure 3.4 compares the employment of the Preston workforce with that of all of Australia. Preston has more labourers and plant operators, reflecting the agricultural and mining industries, and is shc n clerical workers and professionals, reflecting its regional and rural character.

Preston was broadly similar to Australia as a whole, however the similarity was not statistically significant, with the major differences being the percentages of administration, finance, clerical, mining workers and labourers discussed above. Preston was considered suitable for use as a stratum for high density sampling in this survey, particularly considering economic and feasibility criteria. The lack of financial and public administration workers was noted, and an attempt was made to target these groups in the multistage sampling applied to the Perth stratum.

Perth Statistical Division

A Multistage sampling technique was used. This stratum was further subdivided based on employment and membership of computer interest groups. An effort was made to target accounting, finance, government and clerical workers, as these employment categories had a coverage in the Preston stratum below the Australian average. Each sub-stratum was sampled separately, either by sending a key person four or six questionnaires for random distribution, or by some other random selection means.

The following sub-strata were sampled:

Academics:

Academics from Edith Cowan University Perth Campuses in the Departments of Accounting, Research and Computer Studies were selected by listing their names, throwing a dice and selecting that person in the list whose position corresponded to the value of the dice. The selected person became the starting point for the next selection. The selection was repeated until sufficient cases were obtained.

Accountants and Finance Workers

Accountants and finance workers were selected for inclusion in the sample due to the less than average coverage this employment category had received in the Preston stratum, see Figure 3.3. Three accountants, based at the Edith Cowan University, The Perth Stock Exchange and a large Perth Accountancy practice, each distributed six questionnaires randomly at Accounting conferences.

A.C.S. S.I.G. Members (Australian Computer Society Special Interest Group)

Each of the twelve members of the Software Quality Assurance S.I.G. was sent a questionnaire and was asked to distribute it randomly at their place of work, largely major government departments. This area of employment and clerical workers in general, had a lower than the Australian average coverage in the Preston stratum.

P.C. Micro User, End User and Medical Informatics S.I.G. secretaries were each asked to distribute four questionnaires at random.

Other

The Secretaries of the West Australian Lotus Users Group and of Women in Computing were also asked to distribute six questionnaires randomly. Questionnaires were sent for further onward distribution to four scientists and engineers, suggested by respondents in the Preston stratum. Six staff members of the Department of Computer Studies distributed questionnaires to acquaintances who did not fall into any other sampled sub-strata.

Transmittal and Follow-up

Each questionnaire was accompanied by a letter of transmittal and a prepaid return envelope. Non-response follow-up was impossible in most of the case in this stratum. In the few cases where non-respondents could be identified, follow-up was by telephone and the reasons for non-response were solicited in an effort to detect bias.

South Australia, Victoria, New South Wales and Queensland

Selective sampling of certain sub-strata was undertaken to give a greater representation of expert spreadsheet developers in the final sample. This was justified by the need to ensure sufficient numbers of expert developers to form a category in the proposed taxonomy. The secretaries of Lotus User Groups in Sydney, Melbourne, Adelaide and Brisbane and the Sydney and Melbourne P.C. User Groups were sent six questionnaires for redistribution. Follow-up of non-respondents was infeasible.

Sample Size

An objective calculation of the required sample size was inappropriate due to the non-probabilistic nature of part of the sample design. A sample size of one hundred was subjectively selected as:

a) This was felt to be large enough to give sufficient variation to develop a taxonomy.

b) This sample size was economically feasible.

c) This was the largest number of cases suitable for input to some statistical procedures for multivariate and cluster analyses using the SYSTAT statistical software.

Two hundred and fifty questionnaires were dispatched to get at least one hundred useable replies.

3.4.3. Bias in the Sampling Procedures

Ideally, if probability based random selection had been used, this sample would have represented the populat. ,n under study with a clearly defined probability of random sample error. Every member of the population would hav. had an equal chance of being included in the sample and results could have been generalised to the population as a whole. The availability of a complete sampling frame of the population would have made this feasible, though extensive economic and time resources would also have been required. These were all unavailable. Sample bias may have been introduced due to the partial non-probabilistic sample design.

If random probability selection had been possible, small, rare, but nevertheless important groups might not have been represented in this sample. Anderburg's suggestion (1973, p. 11) of explicitly including such cases in the sample, provided the rationale for sampling 'experts' in the Eastern States stratum and 'hobby' developers in the Preston stratum.

Independence of units sampled, i.e. the selection of one unit not making the selection of another more likely, was also profitably violated in this study. Stratification, use of volunteers and the 'snowball' effect in the Preston stratum, were relied upon to get a high coverage of developers. These methods were necessary for feasibility and economic reasons but possibly introduced bias. Anderburg justifies this course of action as a virtue rather than a necessity:

> If selection of some data units promotes the candidacy of others, the effect should be exploited for the evidence of association rather than neutralised in deference to independence. (1973, p. 11)

This is what cluster analysis or finding groups in data is all about.

Further bias could have been introduced with the developer's self-selection of which spreadsheet development project to analyse. However developers were explicitly instructed to choose any sample of their work, and were assured that size, complexity and importance of the spreadsheet were immaterial to the current purpose.

The attitudes of the developers to taking part in the study may have introduced bias. Volunteers presumably had high interest, as had many developers within the Preston Stratum due to their loyalty and interest in one of the first projects initiated by their new regional University campus. University status, with its attendant media publicity, was achieved during the data collection phase of the study. Some respondents in the Perth and Eastern States strata were possibly less interested, particularly if they had been instructed to complete the survey questionnaire by superiors or quality control personnel. In spite of assurances of anonymity, further bias could have been introduced by developers not wishing to admit to less than perfect development practices.

Davis and Cosenza (1985, p 229) state that non-probabilistic samples have "basic shortcomings of high variability error and lack the characteristics to estimate this error". This sample bias of this study was due to that part of the sample design that was non-probabilistic in nature. However this was justified in view of the feasibility of attaining the goal of developing a special purpose taxonomy of spreadsheet application development.

The nature of this study was exploratory data analysis in the absence of both a known sampling frame and a population of known parameters. The aims were both to develop a special purpose taxonomy and to suggest hypotheses to guide future research. These hypotheses could be accepted or rejected using probability based confirmatory statistics on new data, i.e. hypotheses generation not hypotheses acceptance/rejection was the purpose of this study.

It is not claimed that the results of this study are directly extendable to the population at large. Hopefully they will be but this will require a confirmatory study with new data. Godehardt supports this view:

> Methods of exploratory data analysis are designed to support researchers in uncovering new phenomena. The essential problem in the interpretation of the results of such exploratory analysis lies in the fact that we are tempted to generalise these models or hypothesis which have been derived from one specific sample to a whole population. This however, is admissible only if models from exploratory studies have been validated with methods of confirmatory statistics and with new data. Model validation on the basis of exploratory methods alone is impossible. The purpose of confirmatory statistics (with careful experimental design) on the other hand, is to validate phenomena and hypothesis from investigations that have previously been performed. . . . This confirmation is necessary . . . . Pure confirmation alone is not sufficient for progress . . . Exploratory methods are indispensable for the advance of scientific research. (1990, p. 16)

3.4.4. Instrumentation

The survey was conducted using active primary data collection by means of a self-administered questionnaire. A copy of this questionnaire with letters of transmittal can be found in Appendix A.

Rationale for choosing mail interview

A mail interview was selected for several reasons, as suggested by Davis and Cosenza (1985, p. 282).

a) Control of bias effects that might have been introduced by an interviewer.

b) Flexibility in allowing busy respondents to schedule the completion of the questionnaire at a time that suited them.

c) Accuracy on sensitive data. The respondent had time available to look up data required from within a spreadsheet rather than making an educated guess during a personal or telephone interview.

d) Economic considerations. Submission costs were low when compared to personal interview.

e) Feasibility of mail interviews, from the geographical location of the researcher in Bunbury, 200km from the nearest metropolitan area.

f) Response confidentiality.

In making the choice of a mail questionnaire, the researcher sacrificed any flexibility in response by respondents, and any useful answers to open-ended questions that might have arisen in discussion with an interviewer. In addition there was a risk of a poor response rate. However the advantages of the mail questionnaire outweighed these disadvantages.

Definition of a Spreadsheet Development Attribute

A spreadsheet attribute or variable was equivalent in this study to the operational taxonomic character of mathematical taxonomy:

> A *character* in this context may be defined to be any property that can vary between taxonomic units, and the possible values that it can be given are called the *states* of that character. (Dunn and Everitt, 1982, p. 11)

The states of the attributes identified the spreadsheet development activity. These states were variant over the cases included in the sample. Examples of such attributes could be a) date of completion of spreadsheet, b) age of spreadsheet developer or c) annual turnover of company where the spreadsheet was developed.

Number of Attributes required

How many attributes should have been included? Obviously more would have been better than less, but this would have caused problems with the data processing due to software limitations. Sneath and Sokal (1973, p. 106) suggest that at least sixty variables (attributes) should be used. In general, mathematical taxonomy articles do not give directions for calculating the optimum number of attributes required. It is frequently suggested that, the number of attributes should not be greater than twenty percent of the cases analysed. It was not known in advance which attributes would have the best discriminatory power between cases and which would prove to be redundant in this endeavour. Neither was it known in advance, whether some attributes would be highly correlated. A decision was made to collect more attributes than would be finally used to develop the taxonomy, and select posteriori those best suited to show variation between the cases.

Criteria for Attribute inclusion

Many different classifications would have been possible from the same set of cases. The choice of attributes determined which of many possible taxonomies was developed. The following criteria were used to determine attribute inclusion:

a) **Relevance** - The attributes chosen reflected the purpose of the classification as a tool to assist in the integrity and control of spreadsheet development.

b) **Variability** or discriminatory power - The attributes chosen were variable over the cases surveyed and had the power to discriminate between cases.

c) **Restrictiveness** - The attribute choice was not restricted to those that had been used for other classifications reported in the literature. The researcher also included attributes chosen on a subjective basis.

d) **Importance** - Consideration was given to the attribute's relative importance and care was taken to include all important and identifiable attributes of relevance (see a).

e) **Redundancy** - Attributes with a high statistical correlation with other attributes, and concordance, were excluded as they were redundant for the purposes of identifying a taxonomy. Statistical correlation alone was not enough to exclude a variable, as such correlation could have arisen just because the two variables belonged to the same taxon (taxonomic group). This was discussed by Jardine and Sibson (1971, p. 171).

f) **Availability** - Attributes which were readily available and easily measured were chosen rather than attributes that the survey respondents could have had difficulty in determining. e.g. It was decided to exclude 'annual turnover of the company' in favour of other more easily determined measures of size and importance such as 'the number of departments or sites on which an organisation was represented'.

Criteria for Attribute exclusion

Sokal and Sneath's discussion on characters (attributes) inadmissible for the purposes of creating a taxonomy was used as a basis to develop exclusion criteria. (Sokal and Sneath, 1963, p. 103)

a) **Meaningless characters** - Attributes that were not a reflection of the inherent nature of spreadsheets under development, were excluded. e.g. names or numbers given to spreadsheets.

b) **Non-orthogonal hence logically correlated** - Attributes that were a logical consequence of another attribute were treated with care e.g. 'the file storage size of a spreadsheet' and 'the number of rows and columns in the spreadsheet'. Their inclusion added nothing except a check on accuracy, as they both measured the same underlying variable.

c) **Invariant** - Attributes that were likely to be invariant over the sample were excluded as these would not have assisted in taxonomy development.

Categories of Attributes

Attributes for inclusion in the questionnaire were chosen in three ways:

a) Using the above criteria for attribute inclusion and exclusion.

b) By an extension to a scheme devised for biological micro-organisms by Dunn and Everitt.

c) By a scheme devised by the researcher, based on whether the attribute value was known prior to the development of the spreadsheet application.

Dunn and Everitt's biological "Characters for classifying micro-organisms" (Dunn and Everitt, 1982, p. 11) was adapted to describe the non-biological environment of spreadsheet application development. Dunn and Everitt's work drew on a previous classification of attributes reported by Sneath and Sokal. (1973, p. 90)

a) **Morphological** - spreadsheet shape. The numbers of rows, columns and dimensionality, spreadsheet size.

b) **Physiological** - spreadsheet output, range of distribution, life-span.

c) **Biochemical** - spreadsheet use, graphics.

d) **Chemical constituents** - spreadsheet building blocks, logic, functions.

e) **Cultural** - development environment, developer demographics.

f) **Nutritional** - spreadsheet input, links to other spreadsheets and databases.

g) **Drug sensitivity** - environmental security risks and controls.

h) **Genetic** - inheritance, model type, importance of attributes.

The questionnaire collected both qualitative and quantitative attributes. Attributes were divided into three broad categories, reflecting the proposed use of the taxonomy as an aid to spreadsheet applications development. 'A priori,' 'posteriori' and 'identifier' attributes were identified. These differed on the stage of the spreadsheet life cycle, when their status could be determined.

'A priori' attributes were those known before the spreadsheet was developed. They measured details of the proposed spreadsheet application, the developer and the environment in which the application was to be developed.

'Posteriori' attributes were those attributes whose value was only available after the spreadsheet had been developed. They were of no direct assistance in supporting the use of the taxonomy to suggest spreadsheet design and control measures. However the questionnaire included a section on 'posteriori' attributes, both to provide some data for validation of the taxonomy according to usefulness, and also to provide some of the data required for future studies, which will develop a spreadsheet development control model.

'Identifier' attributes were used to identify the spreadsheet application and the developer and were only used for follow-up contact. To preserve anonymity, these were not held electronically.

Attributes Included

Attributes selected described the:

a) Purpose of the Spreadsheet.

b) Sector, Industry and Organisation where used.

c) Importance of the spreadsheet to the organisation.

d) Time available for the development task.

e) Organisational spreadsheet development policy.

f) Spreadsheet Application and Developer identifiers and demographic details.

g) Developer's spreadsheet interest, training and development experience.

h) Spreadsheet application size and composition.

i) Inclusion of macros, graphics, borders, absolute and relative referencing, formula complexity and modular design.

j) Usage of corporate and private data.

k) Data entry methods.

l) Spreadsheet output distribution and life-span.

m) Inclusion of control measures for design, formulas, input and output, testing, documentation and security. The developer's opinion was also canvassed as to the efficacy of these control measures for their particular development situation.

Scales to measure attributes

Mixed scales were used to code the questionnaire answers. Itemised rating scales were used for qualitative attributes. Some of these were coded as binary dichotomous (yes/no) if they consisted simply of the two-state presence or absence of a feature e.g. macros, graphics. Qualitative attributes were coded on ordinal scales if they had more than two categories that could be appropriately ranked. A few variables with a choice of categories with no ranking order, required the use of nominal (category) scales.

The quantitative attributes were coded on interval scales e.g. questions in relation to the size of the spreadsheet application.

Some clustering runs used only binary dichotomous data. For these runs, *n* nominal variables were converted to *n-1* binary dichotomous variables where *n* was the number of categories in the original nominal variable. Ordinal variables could be converted to binary dichotomous variables in the same manner, losing the effect of category ranking. Interval variables were converted to ordinal variables using ranges mapped to category values and from thence to binary dichotomous variables.

Most of the clustering runs followed Romesburg's suggestion that when mixed qualitative and quantitative variables are present, they should be treated as if they are quantitative. i.e. all ordinal variables were treated as if they were interval scaled. (Romesburg, 1984, p. 171).

Questionnaire design

A sample questionnaire can be found in Appendix A. The questionnaire was designed in three sections. The first section of twenty questions asked about the spreadsheet developer and the organisation where they were employed. The second section contained forty questions about the spreadsheet application. The third and final section included fifty five questions relating to spreadsheet design and control issues and the developer's opinion as to their efficacy for their particular spreadsheet application. The data collected in the third section was put aside for use in the follow-up studies foreshadowed in the final chapter. This data was collected at the time of the initial survey, to avoid a follow-up study of the same developers and spreadsheets, some time after the initial study when developers or spreadsheet projects might have become inaccessible.

Rationale for design

Guide-lines on the design of questionnaires by Davis and Cosenza (1985, p. 16-18) and Bailey (1982, p. 516) were followed. The necessity for inclusion of each question was carefully considered, in an attempt to keep the questionnaire to a reasonable length.

Questions were asked in simple, clear English. Loaded and emotional terms, and spreadsheet jargon were avoided, where possible. Care was taken not to use words that suggested a preferred response. Each question was precise and dealt with only one subject. There were no 'double-barelled' questions requiring two answers.

The questionnaire layout was simple and easy to follow. The layout was designed both to simplify response, and for ease of coding and data entry. Questions on like subjects were blocked together for ease of response and to avoid placing too great a burden on the respondent's memory. All questions requiring access to a computer were placed in section two, where they would be answered after the respondent already had made some investment in completing the questionnaire.

To make the questionnaire quick and easy to complete, most questions were prepared using itemised rating scales. To simplify response, split ballot techniques were deliberately not used and questions usually had 'yes'/'no' in the same sequence. Where appropriate, provision was made for neutral or 'do not know' answers. At other times, respondents were forced to choose one of the available answers. Closed questions were used to limit responses and simplify the tallying.

Clear and easy instructions and a completed example were provided for each section. The questionnaire started with simple and easy questions and lead on to more complex questions later. The more sensitive questions relating to security controls were asked only in the third section; by that time the developer would have some commitment to finishing the questionnaire.

Questions were worded not to embarrass the respondents. The questions were asked in a non-threatening manner and participants were assured of anonymity. Requests for the respondents' names and telephone numbers (to be used for contact only) were buried deep within the questionnaire and not readily visible at a cursory glance. It was hoped that this would reassure respondents.

The respondents were treated with courtesy at all times and never 'talked down to'. They were thanked for participating in the survey.

Identification of response bias

Participants were asked to give their opinion as to the importance of their spreadsheet application. The possibility of some response bias was accepted and they were given guide-lines to gauge this importance in an effort to control bias.

Unintentional response bias was possibly introduced when participants were asked to gauge their own spreadsheet development expertise. Categories available were 'Novice', 'Knowledgeable' and 'Power User'. The results of the survey suggest the possibility of response bias to this question on a gender basis. This is discussed further in the final chapter.

3.4.5. Pretest / Pilot Study

Initial 'one on one' test and discussion with subject

The questionnaire was tested on a sample of four persons from different backgrounds.

Participants completed the questionnaire and were then interviewed in person or by telephone. Problems with the questionnaire presentation and content were identified and corrected.

Pilot test

A pilot study was undertaken with the submission of the questionnaire to twelve respondents drawn from diverse backgrounds. Respondents were also asked to note the time taken for the filling in of the questionnaire and to choose between high quality green paper and grey/white recycled paper for the final questionnaire. Respondents' opinions on questionnaire content and presentation were solicited.

The analysis of this pilot test highlighted the need for the fine tuning of some questions and the movement of all questions requiring computer access, to the end of section two.

The pilot test also provided data for use in coding and developing the database and spreadsheets required for the analysis phase of this survey.

Rationale for the Pilot test

The pilot test allowed the testing of the questionnaire. Was it easy to understand? Were there sufficient instructions? Did it provide the required answers? Was every question used? Were more questions required?

The pilot test helped with the management of the survey. It determined whether the desired image was projected. It guided the choice of paper. It determined a reasonable estimate as to the time taken to complete a questionnaire. It determined the feasibility of the postal delivery and telephonic follow-up procedures. It gave an initial estimate of levels of non-response and some of the reasons for this.

The pilot test determined the feasibility of the proposed data storage and data import/export between computer programs. It provided test data for use in validating the statistical methods used and gave the researcher an opportunity to gain experience in this area with real data (Stopher and Meyburg, 1979, p. 101-120).

3.4.6. Questionnaire Validity and Reliability

The rationale behind establishing instrument validity will be discussed in detail in chapter 5 and so will not be duplicated at this stage of the dissertation. The questionnaire would be considered valid if it measured what it purported to measure. Content, criterion referenced and construct validity were considered. Questionnaire reliability was established by examining the responses of the original four 'one on one' respondents with their subsequent responses to the pilot study.

3.4.7. Submission to Participants

The questionnaire was submitted to participants with a reply paid envelope and a letter of transmittal. The method by which the participants received the questionnaire differed in each of the three strata and was outlined earlier in this chapter when the methods of drawing a sample from each of these strata were discussed.

Letter of transmittal

A letter of transmittal was included with the questionnaire. Its purpose was to elicit maximum number of returned questionnaires. Slightly different letters of transmittal were used in each stratum and a sample is included in Appendix A.

This letter identified the subject of the research, the University and the researcher. It was printed on official University headed notepaper and personally signed by the researcher. Where possible, the recipient was identified by name. Davis and Cosenza (1985) have identified that the specification of a firm deadline has no effect on increasing the number of responses, whereas prepaid postage, an appeal and follow-up all resulted in an increase response rate. No firm reply date was set but the letter suggested several good reasons why the subject should respond within a reasonable time of two weeks.

3.4.8. Survey Follow-Up Procedures

It was necessary to follow-up some of the developers in the sample.

Non-response follow-up

Follow-up of those developers who did not return their questionnaire was attempted where possible. Follow-up of non-respondents was impossible in the Eastern States stratum as developers who had received a questionnaire were unidentifiable prior to their response. Non-respondents in the Preston stratum were followed up by telephone up to three times at two week intervals, starting three weeks after they had received a questionnaire. Developers in the Perth metropolitan stratum were treated either as those in the Preston or Eastern states strata according to whether they were identifiable.

Preston developers declining to participate

The original intention was to check a sample of non-respondents for possible bias. However there were very few developers contacted in Preston who did not wish to contribute. Some initially felt they were too inexperienced or their spreadsheets too simple, but after telephonic follow-up they realised the importance of their contribution.

Response error follow-up

Some returned questionnaires had probable response errors, i.e. discrepancies between reported and real data. These were detected by the methods outlined in Section 3.5 below. Where such errors appeared to be unintentional, the developer was contacted by telephone and thanked for their interest and contribution to the survey. They were then asked for the amended information and an appointment was made for a convenient time to phone and get the required data. Where such errors were suspected of being deliberate, consideration was given to removing that case from the sample.

3.5. Pre-Analytical Processing of Data

3.5.1. Initial Data Edit

The returned questionnaires were scanned by eye to identify anomalies due to poor handwriting and ambiguous or incomplete answers. Problem questionnaires were submitted to the follow-up procedures outlined above

3.5.2. Data Coding and Verification

Initial Coding

Questionnaires were coded according to the codebooks shown in Tables 22 and of Appendix B. Missing values were given a value of 9.

A review was made of each question where 'other' was the selected answer. Subsequent to review this was either a) accepted, b) recoded to one of the other options or c) referred for respondent follow-up.

Each case was numbered in sequence with an identifier starting with 1. This identifier was written on the front of the questionnaire and a separate list was kept of the name and contact details of the respondent and their case number. To ensure anonymity, this list was kept locked up and the original contact details were defaced from the questionnaire.

Verification

The coding of the questionnaires was checked by another person who signed the correctly coded questionnaires and returned the discrepancies to the researcher for action. After correction, they were resubmitted to the data coding verification process.

3.5.3. SURVEY Database

Database Design

The SURVEY.DBF database was implemented in ENABLE OA software. (see Appendix B Table 24 for field names). Fields were either defined as numeric integers or alphanumeric. Numeric fields had range constraints activated. All numeric fields also accepted the number 9 (used to code missing data except in question 3).

The primary key of this file was LABEL$, the unique identifier of each case and the number written on the front of the questionnaire during the coding process.

An on-line data input/verification form was designed to enter all fields and apply range checks and produce an error message if database constraints were violated. Invalid data was not permitted to enter the database. This form was also designed to be used for verification. When the key of a case (record) was entered, a blank form appeared. The remaining fields were retyped and the form compared them to the data stored in the SURVEY database, alerting with an error message if any discrepancies were found.

Data Entry

One hundred and seven cases were entered to the SURVEY database using the specially prepared on-line data-entry form. Any errors notified by the entry form were corrected. The cases were entered to the database in the sequence of the value of the key LABEL$.

Data entry verification

When the initial data entry was completed, the form was re-used in data verification mode. All data was re-entered and compared to the stored database. Any errors were corrected and resubmitted to the verification process. The form was signed on completion of the verification data entry. Only when all questionnaires had two signatures a) for verification of data coding and b) for verification of data entry was the database passed on to the next stage for the development of new variables, see section 3.5.5.

3.5.4. CONTROLS Database

This ENABLE OA database, CONTROLS.DBF and its accompanying on-line data entry/verification form were similar in design to the SURVEY database. The database was used to store the answers to part three of the questionnaire dealing with design and security control implementation. Data entry and verification were completed as above and the resulting database was set aside for use in follow-up studies foreshadowed in the final chapter of this thesis. The responses to question 61 were required for the validation of the taxonomy under the 'usefulness' criterion as described in section 5.4.8.

3.5.5. Variable Transformations

A few variables were transformed prior to submitting the data-set to the multivariate cluster analysis procedures. Some variables were combined to form super-variables while others had their number of possible values reduced. Others were calculated e.g. the XSIZE variable. Some variables required scale type changes before submission to cluster analysis statistical procedures requiring ordinal or binary dichotomous input. Table 24 (relegated to Appendix B as it occupies nine pages) sets out for each of the 201 variables used in the statistical analyses:

a) Variable name
b) Scale type: nominal, ordinal, binary dichotomous, interval, ratio or alphanumeric label.
c) Source (parent) of any transformation: Either the question number from the survey questionnaire or the variables from which they were transformed.
d) Content description
e) Range of values and meanings
f) Presence or absence in raw, binary dichotomous and ordinal data-sets for use as input to the clustering procedures.

3.5.6. Super-Variables

Spreadsheet Size

The file storage size of a spreadsheet worksheet was considered an imperfect basis for comparing the size of spreadsheets as different spreadsheet software stored spreadsheet templates in different ways e.g. the treatment of unoccupied cells. The size of the matrix i.e. rows by columns by number of worksheets also was unsuitable as a basis for comparison, as some spreadsheets had a modular diagonal design with many unoccupied cells, while others had some cells filled with labels and descriptive matter, not used for calculation.

A super-variable (composite variable) XSIZE was developed in an attempt to minimise these problems. XSIZE contained the ordinal ranks of the 'useful' portion of the spreadsheet sizes and was calculated using an ENABLE spreadsheet template SIZE.SSF. Only that portion of the spreadsheet size devoted to data and formulas was considered, ignoring cells that were unfilled, contained labels, lookup tables, constants etc.

A 'useful' cell proportion was estimated as the smaller of, 1 or the proportion of cells containing data and formulas. This ratio varying in size between .4 and 1 was then multiplied by the size of the spreadsheet in bytes to give an estimate of the size of the 'useful' part of the spreadsheet.

$$useful\_size = @\min(1, .2 \times (CELLFORM + CELLDATA)) \times SIZE$$

This useful-size was then transformed to XSIZE, an ordinal ranking variable, by means of a lookup table within the template that divided the whole range of sizes into six unequal categories.

The spreadsheet template SIZE.SSF also calculated a cell-storage ratio giving the storage size in bytes for a spreadsheet cell:

$$CELL\_STORAGE = \frac{SIZE}{ROWS \times COLUMNS \times WSHEETS}$$

This ratio was then compared with the means of all spreadsheets in the sample and all spreadsheets developed using the same software (PROGRAM$ and VERSION$) to highlight possible anomalies requiring response error follow-up.

Composite variables

Certain super-variables were defined to change nominal scales to ordinal scales, thus permitting the use of distance measures required in the cluster analysis algorithms. These super-variables also reduced the number of variables input to the clustering procedures:

a) XSDENVRN: This variable rated the control of the development environment. It rated having a spreadsheet development policy twice as highly as having it documented or having a library of spreadsheets It did not distinguish how this policy was enforced, provided it was enforced.

$$XSDENVRN = LIBRARY + 2 \times SDPOLICY + SDDOCO + @IF\left(SDENFORC \neq 0, 1, 0\right)$$

XPROF: This variable rated the combined professional and qualification attributes of a spreadsheet developer. It rated a developer with a professional membership, whose highest qualification was school, trade or diploma as having the same status as a developer rated one ordinal group higher on qualification alone.

$$XPROF = QUALIFY + @IF\left(\left(QUALIFY < 4 \text{ and } PROFMEMB = 1\right), 1, 0\right)$$

b) LINKED: This variable rated the degree of linkage of the spreadsheet to other objects. (spreadsheets, databases or WINDOWS objects).

$$LINKED = LINKSS + LINKDB + LINKDDE$$

c) XCOMPLEX: This variable rated the complexity of the physical design of the spreadsheet template.

$$XCOMPLEX = ABSREL + SPLITSCRN + 2 \times LINKED$$

d) XGRAPH: This variable rated the sophistication of the graphics used within a template.

$$XGRAPH = GRAPHICS + @IF\left(GRAPHICS = 1, GRAPHSOP, 0\right)$$

e) XMACRO: This variable rated the sophistication of the macros used within a template.

$$XMACRO = MACROS + @IF\left(MACROS = 1, MACROCOM, 0\right)$$

f) XLOGIC: This variable rated the sophistication of the logic functions used within the spreadsheet based on the concept of 'logic' complexity discussed by McCabe. (1976, p. 308)

$$XLOGIC = IFS + NESTEDIF + 2 \times LOOKUPS$$

g) XFORMULA: This variable rated the complexity of the formulas used within the template.

$$XFORMULA = FORMCOMP + XLOGIC$$

h) ENTKNOW: This variable rated the data entry person's knowledge of spreadsheet data entry procedures. Non-developer users had the lowest rating followed by professional data enterers and finally the designer.

$$ENTKNOW = 4 - ENTERER$$

Transformation from nominal to ordinal variables

Certain variables were transformed from nominal scales to ordinal scales by the reduction in the number of possible values the variable could take. A small amount of information was lost by this process though the judgement was made that this was the best way to proceed as it would permit the use of algorithms designed for ordinal variables as well as the very few algorithms designed to be used primarily with categorical (nominal) variables.

a) XORDFREQ: This variable rated the frequency with which a spreadsheet was run. The values of the nominal variable HOWOFTEN were transformed. Values ranged from 1 to 4 representing a) once, b) few times or occasional with a long gap, c) monthly, and d) daily, weekly and frequently.

b) XSTATUS: This variable rated the employment status of the developer. It was transformed from the STATUS variable. Unpaid helpers had the lowest and executives the highest employment status. Consultants and Self Employed had an XSTATUS of 0 and their status was introduced to the clustering procedures via the binary dichotomous variables STCONS and STSELFEM.

c) THREED: This variable rated the degree of dimensionality of the spreadsheet template. Two dimensional spreadsheets had a value of 0.

Spreadsheets with two to three worksheets had a value of 1, with four to ten worksheets a value of 2 and the remainder a value of 3.

Binary dichotomous variables

Binary dichotomous variables used in this study have only two possible values 0 and 1. Consistently, 1 was taken to mean the presence of a rare attribute and 0 its absence. Some of the clustering procedures used required input in this form. Nominal variables were converted to binary dichotomous scales by coding the presence or absences of a characteristic. When converting an ordinal variable to a binary dichotomous scale, one of two means was used:

a) A value in the existing ordinal scale was selected. Those cases with attribute values above this were coded as '1' and below coded '0'. The selected value was not necessarily the mean. This method reduced an ordinal scale to just two possible values losing considerable information in the process. e.g. in a scale of values ranging from 1 to 6; 5 and 6 could be coded '1' and 1, 2, 3 and 4 coded as '0'. As the cut-off value was subjectively selected, and information was lost, the use of this method was restricted to the few situations where method b) was inappropriate.

b) For each possible value of an ordinal variable, a new variable was introduced coded 1 if the attribute for that case had a value represented by that ordinal value otherwise coded 0. This retained representation of the range of values of the original attributes, but lost their ordinal relationship to each other. For most attributes, this method was judged to be superior. This method was also suitable for the conversion of nominal variables.

The following binary dichotomous variables are defined in Table 24 in Appendix B. They were transformed using method b) unless otherwise stated:

a) PCOMMS, PREPORT, PCLASS, PWHATIF, POPTIM, PFORCST; developed from nominal variable PURPOSE.

b) PREST developed from PURPOSE by method a) where spreadsheets with a purpose of communications, reporting or classification were coded as one.

c) SPUBLIC, SPRIVT and SPERSN; developed from nominal variable SECTOR.

d) IAG, IMINE, IMANUF, IELECT, ICONST, ISELL, IFINCE, IBUSNS, IPUBAD, IEDUC, ICOMP and IOTHR; developed from nominal variable INDUSTRY.

e) OS1 to OS5 developed from nominal variable ORGSIZE.

f) IMP1 to IMP3 developed from ordinal variable IMPORTAN.

g) SDENF0 to SDENF3 from nominal variable SDENFORC.

h) AGE1 to AGE4 from ordinal variable AGE.

i) EXPERT1 to EXPERT3 from ordinal variable EXPERT.

j) TRAIN1 to TRAIN4 from nominal variable TRAINING.

k) READ1 to READ3 from ordinal variable READ.

l) QUAL1 to QUAL5 from ordinal variable QUALIFY.

m) OSCIENCE, OMANAGR, OTEACH, OACCNT, OIT, OTRADE, OCLERK, OOTHER from nominal variable JOB. OIT was also used as a binary dichotomous variable calculated according to method a) in some clustering runs where a developer either had a job in IT (coded 1) or did not (coded 0).

n) STCONS, STEXEC, STDMAN, STEMP, STSELFEM, STHELP from nominal variable STATUS. STCONS was also used as a variable calculated by method a) in some clustering runs where a developer was either a consultant (coded 1) or was not (coded 0).

o) XSZ1 to XSZ6 from the calculated super-variable XSIZE.

p) XGRAPH0 to XGRAPH3 from super-variable XGRAPH.

q) XMACRO0 to XMACRO3 from super-variable XMACRO.

r) FORMCOMP1 to FORMCOMP3 from ordinal variable FORMCOMP.

s) RUNBY1 to RUNBY3 from ordinal variable RUNBY.

t) ENTSELF, ENTCLRK and ENTUSER from nominal variable ENTERER.

u) OUTSELF, OUT1DEP, OUTMDEP, OUTEXORG from ordinal variable OUTSCOPE.

v) XFREQ1 to XFREQ5 from super-variable XFREQ.

w) CDETRAN, CDRPTS, CDOTHR from nominal variable WHEREFROM.

x) KEPT1 to KEPT3 from ordinal variable KEPT.

3.5.7. Data Structures for Entry to Statistical Analysis

Raw data Spreadsheet

An ENABLE OA spreadsheet RAWDATA.SSF was created transferring data from the SURVEY.DBF database. All values of '9' representing missing data were replaced with the character 'space'. After data screening as outlined in section 3.5.8 this spreadsheet was exported in LOTUS format as RAWDATA.WK2. The spreadsheet was then input to the statistical analysis package SYSTAT and converted to SYSTAT internal data-set format as RAWDATA.SYS. Variable transformations were applied to the spreadsheet file RAWDATA.SSF as outlined in section 3.5.5. Some variables were deleted leaving only an identifier and variables coded on an ordinal scale in spreadsheet ORDDATA.SSF. The following forty five ordinal variables and LABEL$ were included:

OIT	ORGSIZE	CDCHANGE	ENTCLRK	QUALIFY
		CDNEW	ENTKNOW	PROFMEMB
PWHATIF	IMPORTAN		RUNBY	
POPTIM		LINKED		EXPERT
PFORCST	ENUFTIME	LINKSS	PRIVATE	XTRAIN
PREST	SDPOLDC	LINKDB		
	SDENFORC	LINKDDE	OUTSCOPE	READ
SPRIVT			XORDFREQ	USERGRP
SPERSN	LIBRARY	XGRAPH	KEPT	
SPUBLIC		XMACRO		XSTATUS
	XSIZE	XLOGIC	GENDER	STCONS
ICOMP	THREED	FORMCOMP	AGE	STSELFEM

Export from the ENABLE spreadsheet in LOTUS format for import to a SYSTAT data-set ORDDATA.SYS was handled in the same way as for the raw data-set described above.

Binary dichotomous data Spreadsheet

Variable transformations were applied to the spreadsheet file RAWDATA.SSF as outlined in section 3.5.5. Some variables were deleted leaving only an identifier and variables coded on a binary dichotomous scale. The presence of an attribute was coded as 1 and its absence as 0 in all cases. This spreadsheet was named BDDATA.SSF. The following one hundred and twenty six binary dichotomous variables and LABEL$ were included:

AGE 1-4	IAG	OS 1-5	ABSREL	RUNBY 1-3
	IMINE	IMP 1-3	SPLITSCRN	ENSELF
PCOMMS	IMANUF		BORDERS	ENTCLRK
PREPORT	IELECT	ENUFTIME	MODBLOC	ENTUSER
PCLASS	ICONST		MODDIAG	PRIVATE
PWHATIF	ISELL	SDPOLICY		OUTSELF
POPTIM	IFINCE	SDDOCO	LINKDDE	OUT1DEP
PFORCST	IBUSNS	SDENF 0-3	LINKSS	OUTMDEP
	IPUBAD		LINKDB	OUTEXORG
SPUBLIC	IEDUC	LIBRARY		
SPRIVT	ICONST	THREED	XGRAPH 0-3	XFREQ 1-5
SPERSN	IOTHR	XSIZE 1-6	XMACRO 0-3	KEEP 1-3
USERGRP	OMANGER	FORMCOMP 1-3	IFS	CORPDATA
GENDER	OSCIENCE		NESTEDIF	CDETRAN
	OTEACH	STCONS	LOOKUPS	CDRPTS
	OACCNT	STDMAN		CDOTHR
QUAL 1-5	OIT	STEMP	EXPERT 1-3	XCDMOD
PROFMEMB	OCLERK	STSELF	READ 1-3	CDNEW
	OOTHER	STHELP	RAIN 1-4	

Export from the ENABLE spreadsheet in LOTUS format for import to a SYSTAT data-set BDDATA.SYS was as described above for the raw data-set.

3.5.8. Data Screening

Input data screening

The database data entry forms had built-in range checks and only allowed data within a valid range into the database. The validation mode of the same forms involved the retyping of data distanced in time from the original data entry. Differences were highlighted and corrected.

Histograms and tabulations

Histograms and box plots were drawn from the SYSTAT data-sets and checked by eye for outliers, anomalies and signs of possible bias. The data-sets were also checked with the SYSTAT TABLES command. Contingency tables showing percentages and frequencies, maximum, minimum, mean and standard deviations for each variable, were assessed for plausibility.

Reasonableness checks

The SIZE.SSF spreadsheet template also performed a check calculating the number of bytes storage per cell. The SIZE.SSF template was then sorted on the primary key PROGRAM$ (software used) and the secondary key VERSION$. Differences between individual templates and the general range for others developed with the same software were identified by eye.

Checks were also performed using SQL (Structured Query Language) on the SURVEY.DBF database to identify intra-record anomalies (between variables within the same record):

a) any binary dichotomous variable that had a value of 1 on more than one variable derived from the same source nominal or ordinal variable. e.g. KEPT1 and KEPT2 both equal to 1.

b) any cases where the organisation size ORGSIZE was incompatible with the range of distribution of the template output OUTSCOPE. e.g. a developer in an organisation with only one department sending the spreadsheet output to many departments.

c) any cases where there was no identified spreadsheet development policy yet the data showed the availability of a documented copy of this policy and/or its enforcement by other than the developer.

d) any cases where CELLFORM, CELLDATA, CELLBLNK, CELLCONS, CELLLABL and CELLOTHER added up to more than 120%.

e) any case where there were no graphics used yet the sophistication of graphics variable had a value.

f) any case where there were no macros used yet the macro complexity variable had a value.

g) any case that was not modular, yet had a value for type of module.

h) any case that was run by self only yet data was entered by the user. Data entered by a clerk was considered acceptable.

i) any developers of status consultant with a low level of expertise.

Anomalies were checked thoroughly and referred for respondent follow-up if required.

<u>Identification and treatment of missing data</u>

Missing data was identified by a space in the SYSTAT data-set. A check was made to see if this was random or appeared to follow some pattern that might identify bias. Missing data were treated in one of three ways a) respondent follow-up where possible, b) deletion of the case, and c) estimation of the missing data. Other possibilities of treating missing data as data itself or of deleting the variable concerned were not used in this study. The major area where missing data was difficult to obtain or where there was a strong suspicion that the data given was incorrect, was 'spreadsheet size'. Here the data was estimated using the spreadsheet template SIZE.SSF, which gave the average number of bytes per cell for each brand of spreadsheet software. If the respondent had completed the number of rows, columns and worksheets, the number of cells could be calculated. It was then an easy matter to estimate the spreadsheet size using the average for all spreadsheets developed with that particular software. This was felt to be a near enough approximation considering the subsequent transformation to 'useful cell percentage' and the eventual six ordinal categories of size.

Identification and treatment of outliers

Possible outliers in the SYSTAT data-sets were identified by three methods:

a) All variables with a binary dichotomous scale were analysed using the SYSTAT TABLES command to ascertain if one of their values had a frequency of less than 10%. Tabachnick and Fidell (1989, p. 67) described analysis problems when such low occurrences were retained. The variable GENDER was removed from the clustering process for this reason. If left, correlation coefficients using this variable in the clustering process, would have had a higher influence on the similarity scores than was appropriate.

b) The standardised scores of all variables were examined and any having a score of greater than ± 3 were reconsidered.

c) Histograms and box plots were drawn for each variable to ascertain if any outlier values could be spotted by eye.

d) A normal probability plot was drawn for the original SIZE data and scanned by eye for non-linearity and possible outliers.

Several possible outliers were treated by

a) Rechecking the data coding, data entry, and any variable transformations involved and correcting if necessary.

b) Confirming that a code intended to represent missing data had not been taken to represent real data.

c) Checking the data with the respondent.

d) Accepting that the distribution was non-normal and reducing the influence of the outlier by changing the score so that it remained deviant, but less so than previously.

e) Discarding the variable involved particularly if it had a high correlation with another retained variable.

The remaining possible outliers were reconsidered carefully. Discarding them from the data-set could result in the non representation of important but rare groups within the final taxonomy. When a case had possible outliers on more than one variable and there was considerable doubt as to the accuracy of the original data then the whole case was discarded. The remaining possible outliers were marked for further consideration and retained. The opportunity was available later to discard them from the data-set, when the results of the early clustering runs and their influence upon them were known.

3.5.9. Standardisation of Data Matrix

The units chosen for measuring attributes could have had an arbitrary effect on the similarities between cases. Standardisation recast attributes into dimensionless units negating this effect. Standardisation also allowed all attributes to contribute to the similarities between objects in the same way, as it removed the higher weighting given to unstandardised variables with large ranges, or high or low means. The data matrices (data-sets) were standardised across variables using the SYSTAT **STANDARDISE** command. Each Z-score had a mean of zero and a standard deviation of one. They assisted in identifying those variables, which showed the greatest similarity within a particular taxon or accounted for the greatest variability between taxons, leading to the development of a diagnostic key for the taxonomy. The standardising function used was:

$$Z_{ij}=\frac{X_{ij}-\overline{X_i}}{S_i} \text{ where } \overline{x_i} = \text{mean and } S_i = \text{standard deviation}$$

3.5.10. Transposition of Data Matrix

Transposed data matrices were prepared using the SYSTAT **TRANSPOSE** command. The cases became columns and the variables, rows. These transposed matrices were required for input into clustering procedures clustering variables rather than the more frequently clustered cases. Some of the cluster analysis runs using correlation coefficients as distance measures, also required the prior transposition of the data matrix.

3.6. Cluster Analysis

3.6.1. Overview of Clustering Procedures

Cluster Analysis is a multivariate data analysis procedure used by mathematical taxonomists. Both the ordinal and binary dichotomous SYSTAT data-sets underwent many cluster analyses. The objective of each cluster analysis procedure was to divide the available cases into groups, maximising between group variance and minimising within group variability over selected spreadsheet attributes. Two different methods of obtaining clusters, Kmeans and agglomerative hierarchical tree clustering were used and their results were compared. Several cluster analyses runs were performed varying the input variables and other parameters. connected with the clustering algorithms

Three runs were selected as the basis for a special purpose taxonomy of spreadsheet applications development suitable for use in the management and control of spreadsheet development. Using the output of these cluster analysis runs, the cases were divided into clusters and the variables (spreadsheet attributes) that had the most effect on the formation of these clusters were identified. A taxonomy of spreadsheet applications development was produced with a diagnostic key suitable for placing a case within a taxon or category within the classification.

3.6.2. Agglomerative Hierarchical Tree Clustering

Input data structures

The input data structure to all agglomerative clustering runs was a two-mode data matrix where the n rows $Y_j\ j=1,n$ represented the n cases derived from a questionnaire return. The p columns represented the variables (spreadsheet attributes). Each row of the matrix defined a vector in p dimensional space.

$$Y_j = \Sigma x_{ij}\ i=1,p$$

Two separate input data matrices were prepared for ordinal and binary dichotomous scaled variables. The ordinal matrix was standardised across all attributes to a mean of zero and unit standard deviation. This nullified any disproportionate effects due to scale measurement differences, allowing each variable to have the same influence on the final clustering solution. (Wilkinson, 1990, p. 22) The first column was always taken up by the unique identifier LABEL$.

	LABEL$	XSTATUS		variable p
CASE 1	0			
CASE 2	2.5	0		
	....	....	....	
CASE n	5	1.8	5.6	0

Figure 3.5 A Section of the Cluster Analysis data input matrix.

Selection of variables

Spreadsheet attributes or variables measured on either ordinal or binary dichotomous scales were divided into three types describing:

a) the spreadsheet development environment

b) the spreadsheet developer

c) the spreadsheet application.

Each clustering run selected appropriate variables of one only of the above types from the input matrices that contained all available variables.

Weighting of Variables

Historically mathematical taxonomists have been divided about the weighting of attributes with Sneath and Sokal suggesting equal weighting for all attributes. (Sokal and Sneath, 1963, p. 50), (Sneath and Sokal, 1973, p. 109). Others suggest that under certain clearly defined circumstances, weighting may lead to more meaningful results. (Everitt, 1980), (Jardine and Sibson, 1971, p. 22)

A recent development of a new controversial category of clustering algorithms, conceptual clustering, uses artificial intelligence based techniques and differential weighting of attributes according to their importance. (Fisher and Langley, 1986), (Thompson and Thompson, 1991)

Variable weighting could be achieved by:

a) Weighting attribute complexity

b) Giving higher weights to attributes that have good discriminatory power between clusters

c) Conversely giving less weight to highly variable attributes

d) Weighting highly, attributes with good diagnostic power

e) Weighting highly, attributes with high functional importance

f) Giving less weight to redundant or correlated attributes

In this study, the use of the Z-scores of variables provided a form of weighting as it reduced the impact of variables with values in small units over a large range. This equal weighting resulted in an equal contribution of all included variables to the solution thus achieving some objectivity as suggested by Romesburg (1984, p. 78).

In some runs, the weighting of variables, suspected by the researcher to be intrinsically of more significance than others, was ignored. Kaufman and Rousseeuw call this "the dilemma of standardisation" (1990, p. 11). As an alternative, in other runs, variables were given zero weight by leaving them out altogether or more significance by repeating their presence in the matrix with duplicate variables with new names.

The selection of a similarity index for each run and the original choice of variables provided two unavoidable sources of weighting.

Distance measures

The clustering algorithms required the measurement of the distance between two cases mapped in *p* dimensional space, in order to cluster together similar cases. The metrics used to measure this distance were of two types:

a) Association or matching coefficients. The greater the value of these similarity coefficients the more similar the two cases.

b) Distance measures, dissimilarity or resemblance coefficients. The smaller the value of this coefficient, the more similar the two cases.

Similarity Coefficients used for Binary Dichotomous Variables

Various indexes were used for binary dichotomous (sometimes qualitative) variables to measure the agreement between two cases over *p* two valued variables. Figure 3.6 shows the values of the attributes of the cases to be compared, arranged into a contingency table, documenting the number of matches and mismatches.

CASE ONE →

TWO ↓

	1	0	TOTAL
1	a	b	a+b
0	c	d	c+d
TOTAL	a+c	b+d	p=a+b+c+d

Figure 3.6 A contingency table used to compare two cases

a = number of variables where both cases have a value 1, d where both are 0, c and b where one case has a value 1 and the other 0. p variables in all.

The main distinguishing characteristic between coefficients was whether to include or not include negative matches d (0,0), as well as positive matches a (1,1) and whether to give the negative matches the same weight. (Lorr, 1983, p. 40). This study used two such similarity coefficients:

a) Simple matching coefficient (Dunn and Everitt, 1982, p. 26), (Kaufman and Rousseeuw, 1990, p. 24), (Romesburg (1984, p. 144), (Wilkinson, 1990, p. 54). This coefficient, ranging in value from 0 to 1, calculated the ratio of positive and negative matches to the total number of variables.

$$\textit{Simple matching coefficient } S_{ij} = \frac{(a+d)}{(a+b+c+d)}$$

However two cases with variables with a (0,0) match may still have little in common e.g. OIT and OTEACH both valued as 0. The developer may well not be an academic nor I.T. worker but could have one of many other possible occupations. SYSTAT implements this coefficient by the commands **CORR, S4** when preparing a correlation matrix (Wilkinson, 1990, p. 54).

b) Jaccard's similarity coefficient was introduced into taxonomy by Jaccard in 1908 (Dunn and Everitt, 1982, p. 26), (Kaufman and Rousseeuw, 1990, p. 26), (Romesburg, 1984, p. 143). This coefficient, ranging from 0 to 1, was similar to the simple matching coefficient except that it excluded negative matches i.e. (0,0). It calculated the ratio of positive (1,1) matches to the total number of variables minus the negative matches. SYSTAT implements this coefficient by the commands CORR, S3 when preparing a correlation matrix (Wilkinson, 1990, p. 54).

$$Jaccard's\ coefficient = \frac{a}{(a+b+c)}$$

<u>Distance measures used with ordinal variables</u>

These coefficients or dissimilarity measures were designed for use with interval and ratio variables but Romesburg (1984) and Kauffman and Rousseeuw (1990, p. 28) suggest their use with ordinal variables. These are resemblance coefficients i.e. the smaller their value, the closer the cases. Several distance measures were used:

a) Normalised or average Euclidean distance coefficient $\overline{d(i,j)}$ (Kaufman Rousseeuw, 1990, p. 11), (Wilkinson, 1990, p. 30) (Romesburg, 1984, p. 97). This coefficient is based on the Pythagorean sum of squares extended to p dimensions. The Euclidean distance between two objects is the square root of the sum of the distance between their components squared distance:

$$d(i,j) = \sqrt{\Sigma\left(x_{ik} - x_{jk}\right)^2}\ where\ k = 1, p$$

The Euclidean distance increased with the number of variables p, so it was normalised to give the normalised or average Euclidean distance:

$$\overline{d(i,j)} = \sqrt{\left(d(i,j)^2/p\right)}\ where\ p = the\ number\ of\ variables$$

A major benefit of this coefficient was that it could still be used with missing values, whereas the straight Euclidean distance coefficient was unsuitable. (Romesburg, 1984, p. 98) SYSTAT implements this metric via the **DISTANCE = EUCLIDEAN** command.

b) Pearson Correlation Coefficient Q (Lorr, 1983, p. 35) (Kaufman and Rousseeuw, 1990, p. 305), (Romesburg, 1984, p. 101), (Wilkinson, 1990, p. 30). This coefficient works best with continuous or interval scales. It is based on the Pearson product moment correlation coefficient r_{jk} that varies between -1 and +1 and does not depend on the choice of measurement unit:

$$Q = 1 - r_{jk} \quad \textit{where } r_{jk} = \textit{pearson product moment corr-coeff.}$$

This coefficient considers a linear relationship between the two variables. SYSTAT implements this metric via the **DISTANCE = PEARSON** command.

$$r_{jk} = \frac{\sum_{i=1}^{n} x_{ij} x_{jk} - (1/n)\left(\sum_{i=1}^{n} x_{ij}\right)\left(\sum_{i=1}^{n} x_{jk}\right)}{\left\{\left[\sum_{i=1}^{n} x_{ij}^2 - (1/n)\left(\sum_{i=1}^{n} x_{ij}\right)^2\right]\left[\sum_{i=1}^{n} x_{jk}^2 - (1/n)\left(\sum_{i=1}^{n} x_{jk}\right)^2\right]\right\}^{1/2}}$$

a) Gamma Coefficient. Wilkinson (1990, p. 30) recommends this distance measure for rank order or ordinal scaled variables. SYSTAT implements this metric via the **DISTANCE = GAMMA** command.

$$1 - g_{ij} \textit{ where } g_{ij} \textit{ is Goodman Kruskal gamma corr-coeff.}$$

Choice of Distance Measure

The variables used (attributes) were of mixed scales. Interval, ratio, nominal and binary dichotomous scales were all represented. Some effort was made to reduce the variables to the same scale prior to cluster analysis with the preparation of two input data-sets, one binary dichotomous and the other ordinal. The binary dichotomous data-set was clustered using either the simple matching coefficient or

Jaccard's coefficient. The ordinal variables were initially clustered using the gamma coefficient for rank order variables. Subsequent runs used the distance measures designed for interval scaled variables particularly the normalised Euclidean distance as suggested by Romesburg (1984) and Kaufman and Rousseeuw (1990).

Resemblance matrix

The data-set was transformed into a resemblance (proximity) matrix with the rows and columns both representing the cases and the cells holding a value for the resemblance coefficient (similarity or dissimilarity) between two cases calculated using one of the distance measures discussed above. It was only necessary to make this calculation for half the matrix as the other half was just a symmetric reversal of the first i.e. the resemblance/distance between **CASE 1** and **CASE 2** is the same as the resemblance between **CASE 2** and **CASE 1**:

	CASE 1	CASE 2	CASE 3	CASE 4
CASE 1	0			
CASE 2	12.4	0		
CASE 3	17.2	6.7	0	
CASE 4	5.6	11.9	32.9	0

Figure 3.7 Part of a Resemblance Matrix

Linkage - amalgamation Algorithms

The hierarchical clustering methods used began with *t* clusters each containing one object and ended up with one cluster containing *t* objects. An object (case) could be considered as the sole member of a cluster of one. At each step two clusters were merged reducing the total number of clusters by one. $t - 1$ amalgamations were required to achieve total fusion of all clusters into one.

Linkage is the name given to the method used to decide whether two clusters should be merged at a particular step. (Wilkinson, 1990, p. 31). A pair of *spanning objects* is defined as a pair of cases, where one is in one cluster, and the other is in a different cluster. Various Linkage algorithms were used in different clustering runs:

Single linkage clustering - the SLINK method

This method sometimes called the 'min' or 'nearest neighbour' method was described by Romesburg (1984, p. 120) and Everitt (1980, p. 25). It was used for some of the early exploratory cluster analyses.

The distance between two clusters was defined as the distance between the two closest members of the clusters. Two clusters were merged based on the minimum distance between a member of one cluster and the nearest member of the other cluster hence the term 'nearest neighbour'.

When considering the amalgamation of two clusters, the algorithm initially listed all pairs of spanning objects from the two clusters. The most similar pair was chosen and their similarity became the similarity of the two clusters. Each member of a cluster was always more like at least one other member of its cluster, than it was like a member of any other cluster. At each stage of the process, the two most similar clusters were amalgamated and the resemblance matrix recalculated.

SLINK was implemented using the LINKAGE = SINGLE command of the SYSTAT software. This method worked well with clearly separated groups but was limited in finding homogeneous groups. Sometimes it resulted in the phenomena of 'chaining', tending to produce long stringy daisy-chain clusters as shown in Figure 3.8. (Wilkinson, 1990, p. 31)

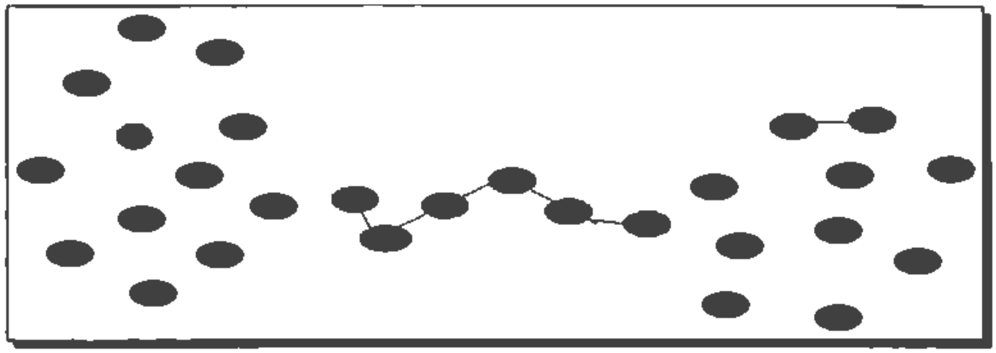


Figure 3.8 An example of chaining showing the first 6 amalgamations adapted from Dunn and Everitt (1982, p. 85)

Due to the daisy-chain effect, SLINK will not find the optimal two clusters that can be easily spotted by eye in Figure 3.8.

Complete Linkage - the CLINK method

This method sometimes called the 'max' or 'furthest neighbour' method and the opposite of SLINK was described by Romesburg (1984, p. 123) and Everitt (1980, p. 28) and was also used for a few of the earlier clustering runs.

The distance between clusters was defined as the distance between the most remote spanning pairs. The algorithm progressed as for SLINK with the preparation of a list of all possible spanning pairs. Clusters were merged based on the maximum distance between spanning pairs. Groups were fused into clusters to maintain the maximum distance between the furthest neighbours of each. Unlike SLINK, each member of a cluster was always more like every other member of its cluster than it was like a member of any other cluster. This method tended to produce clearly defined globular clusters approximately equal in size. It was implemented using the **LINKAGE = COMPLETE** command within the SYSTAT software.

Average linkage - the UPGMA method

The unweighted pair group method using arithmetic averages was described in Romesburg (1984, p. 120) and Everitt (1980, p. 31). This most frequently used method based the merger of two subsets on the middle ground i.e. the average distance between all spanning pairs of objects in the two clusters. It avoided the problems of chaining using SLINK and Romesburg (1984) recommended it over CLINK due to its less stringent requirements. It was implemented using the **LINKAGE = AVERAGE** command of the SYSTAT software.

Centroid Linkage

This method described by Romesburg (1984, p. 136) and Everitt (1980, p. 28) first calculated the centroid of the cluster by determining the average values of all attributes of cases in that cluster. It then based the merger of clusters on the amalgamation of the two clusters with the smallest distances between their centroids. Clusters were replaced on formation by their centroids and the process was repeated till only one cluster was left.

In spite of its intuitive attractiveness, this method was used for only a few runs as it gave problems with producing trees with stray branches that did not connect to others, an outcome also reported by Romesburg (1984, p 136) and Wilkinson, 1990, p. 32) This method was implemented using the **LINKAGE = CENTROID** command of the SYSTAT software.

Ward's minimum variance method

This method described by Romesburg (1984, p. 129) and Everitt (1980, p. 31) was similar to centroid linkage with an adjustment made for covariances. It was used sparingly in this study as Romesburg (1984) reported that it did not guarantee an optimal partitioning of objects into clusters. It was implemented using the **LINKAGE = WARD** command of the SYSTAT software.

__Prepare Dendrogram__

The output of the SYSTAT cluster analysis was produced as a tree or dendrogram. The branches of the tree corresponded to the cases and were labelled with the case number. The tree was ordered so that the most similar cases were next to each other. The length of the branch before it joined another corresponded with the life-time of a particular cluster. When the command PRINT = LONG was used, SYSTAT also printed the amalgamation distances or cluster diameters for each cluster. The order in which the joins were made showed how clusters were formed.

The dendrogram showed the order of the joining of clusters, the lifetime of clusters before fusion into larger groups and the similarity between cases forming a cluster.

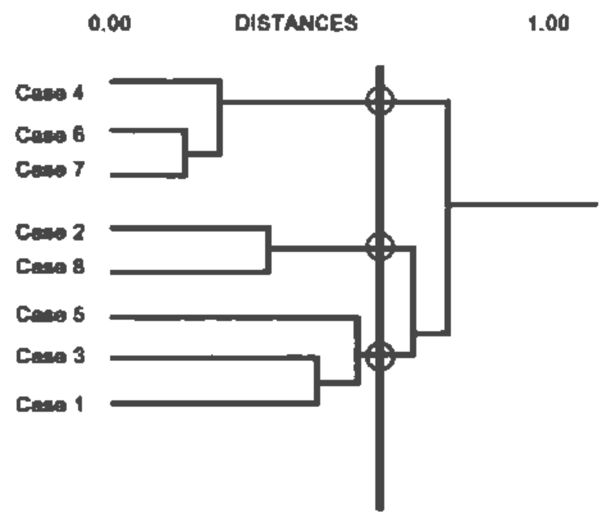


__Figure 3.9__ **An example of a tree dendrogram**

In the above example the tree has been split to give three clusters. Cases 6 and 7 joined first, followed by case 4 to form a cluster, which subsequently had a long life remaining unchanged until the final fusion of all clusters. Then cases 2 and 8 joined to form the second cluster. The remaining cases formed the third cluster. The branches of the tree lead to each separate case. The 'root' of the tree was the final linkage of all clusters into one set.

Transforming the Dendrogram to Clusters

Each dendrogram was transected by a line. The intersects of this line with the branches determined the number of clusters. The line could be moved to another position to give a greater or lesser number of clusters. The line's position was selected both to give a convenient number of clusters and to transect the dendrogram at a position where the number of clusters remained constant over as large a range as possible. This implied that the number of clusters was constant over a wide range of the resemblance coefficient, indicating that they were well separated and therefore least sensitive to error (Romesburg, 1984, p. 213). Romesburg also suggested that the taxonomist could consider cutting the dendrogram at other places if this resulted in producing classes that were related to the research goals. In this study, the first attempts at finding a suitable distance to cut the dendrograms followed Romesburg's first suggestion at cutting where the clusters were most stable, but subsequent attempts looked at cutting at other convenient distances.

Clustering Runs

SYSTAT hierarchical runs were specified using the JOIN ROWS option. Many different clustering runs analysis runs were done varying:

a) The variables used
b) The weighting of the variables
c) The scales on which the variables were measured, binary dichotomous or ordinal
d) Distance measures
e) Linkage methods

These were documented using the run documentation instrument shown in Appendix B. Dendrograms were obtained for each run and possible clusters were assessed see section 3.7 for further details.

Clustering cases and variables

In some runs a simultaneous clustering of rows and columns (cases and variables) was achieved using the SYSTAT **JOIN MATRIX** option. The output display was a shaded display of the original data matrix, differing from the tree dendrogram obtained when clustering the rows or columns separately.

> Rows and columns are permuted according to an algorithm in Gruvaeus and Wainer (1972). Different characters represent the magnitude of each number in the matrix. (Wilkinson, 1990, p. 33)

SYSTAT used an adaptive routine to choose several symbols to display numerical intervals within the matrix. The researcher selected six symbols as an appropriate number for most runs of this type. SYSTAT selected the cut-points between the symbols' ranges to heighten the contrast in the display using techniques derived from computer pattern recognition algorithms.

	OIT	TRAINING	SPUBLIC	AGE	STEMP
CASE-12	0	+++	0	+++	****
CASE-98	o	0	0	0	o
CASE-17	o	o	o	o	o
CASE-66	o	o	:	:	:
CASE-23	:	:	:	:	:

Figure 3.10 An example of SYSTAT matrix clustering output

> Gray-scale histograms for visual displays are modified to heighten contrast and enhance pattern detection. To find these cut-points, we sort the data and look for the largest gaps between adjacent values. (Wilkinson, 1990, p. 33)

The rows of the matrix were arranged in the same sequence as the rows of the tree dendrogram, obtained when the rows were clustered separately. The columns of the matrix were similarly arranged. Each cell within the matrix had one of the six symbols substituted for its numerical value. This display enhanced the visual splitting of the matrix into clusters. Figure 3.10 demonstrates this concept.

3.6.3. Kmeans Clustering Algorithm

The Kmeans algorithm used was an example of partitioned clustering and differed from the hierarchical techniques outlined above. Partitioned clusters contain no other clusters and therefore cannot be represented by a tree dendrogram. The Kmeans algorithm is an example of a 'Hill and Valley' or 'Hill climbing' technique (Dunn and Everitt, 1982. p. 88.), (Jackson, 1983, p. 172). The Kmeans algorithm could be considered as being similar to a multivariate analysis of variance where the groups were not known in advance. It is an iterative procedure assigning cases to a prescribed number of non overlapping clusters as described in Wilkinson (1990, p. 35) based on original work by McQueen (1966). The algorithm was implemented using the SYSTAT **KMEANS** procedure.

Before using this algorithm, the researcher had to decide how many clusters were required. The Kmeans algorithm then selected well distributed 'seed' cases, one for each proposed cluster.

> Seeds for new clusters are chosen by finding the case farthest from the centroid of all cases in Euclidean distance. (Wilkinson, 1990, p. 38).

Each new case in turn was assigned to the cluster represented by its nearest seed. The mean of the cluster was then recalculated to take account of the additional case. This was continued until all cases had been added to a cluster. The algorithm then processed each case separately attempting to re-assign it to another cluster so that the overall within-groups sum of squares calculated using Euclidean distance was minimised. This process was repeated until no more reduction in the within-groups sum of squares could be achieved (Wilkinson, 1990, p. 26).

> It seeks to partition n cases into K groups so that the value of trace W is minimised. W is the p x p matrix obtained from summing the within-cluster sum of squares and product matrices over all k clusters;
>
> $$W = W_1 + W_2 + \ldots + W_k$$
>
> (Dunn and Everitt, 1982, p. 88)

The output of the SYSTAT KMEANS procedure first listed the F-ratios for each variable. Those variables with higher F-ratios were those variables that were the better discriminators between cases.

The output then listed for each cluster; the cases assigned to that cluster, and the statistics of the variables for those cases. Minimum, mean, maximum and standard deviation were calculated. When the run involved standardised data, these statistics gave an easy method of deciding whether higher or lower than average values of variables were responsible for the cases clustering together.

3.7. Exploratory Data Analysis

3.7.1. Clustering Runs

Three separate series of hierarchical clustering runs were carried out using suitable variables to represent the development environment, the spreadsheet developer and the spreadsheet application. Figures 7.1 and 7.2 of Appendix B show forms for recording the following variable parameters:

a) the variables chosen.
b) the weighting of the variables.
c) the initial data matrix, standardised or not.
d) use of ordinal or binary dichotomous scales.
e) the distance measure.
f) the linkage method.
g) inclusion of possible outlier cases.

The resulting tree dendrograms were examined closely and a line was drawn to cut the tree into clusters. If the clusters looked promising for use in developing a taxonomy, a matrix clustering of cases and variables was also executed giving an output of a density plot matrix. Kmeans clustering runs were completed using values of k ranging through the number of hierarchical clusters ± 2.

The outputs from the Kmeans and hierarchical matrix and row clusterings were compared and examined closely, to determine if they could be considered as the basis of the taxonomy, considering the criteria outlined in section 3.7.2 below.

3.7.2. Criteria for Usefulness and Acceptability of Clustering Runs

A priori it was impossible to tell which clustering algorithm would be most suitable. Kaufman and Rousseeuw suggest that:

> It is permissible to try several algorithms on the same data because cluster analysis is mostly used as a descriptive or exploratory tool in contrast with statistical tests that are carried out for inferential or confirmatory purpose. That is we do not wish to prove (or disprove) a preconceived hypothesis: we just want to see what the data are trying to tell us. (1990, p. 37)

Hierarchical clustering algorithms have an inherent defect. They are rigid and can never repair what has been done at a previous step. Once two cases have been joined at a certain level, they can never be separated again. Kmeans avoids this problem. It has as a goal the objective of selecting the 'best' clustering which may or may not be hierarchical. Kaufman and Rousseeuw (1990, p. 45) feel that the two methods are not in competition because their goals are different. If a tree structure is required, as is often the case in the biological sciences, then hierarchical clustering is useful. Alternatively, if a particular number of non-overlapping clusters is required and nesting clusters inside others is unnecessary, then Kmeans is the appropriate choice.

Lorr (1983, p. 101) suggests that at least two different clustering methods should be used to confirm that an underlying structure is indeed being recovered, rather than simply artefacts of the cluster analysis process.

Authors also differ over which linkage to use. Kaufman and Rousseeuw (1990, p. 47) suggest avoiding SLINK because of chaining, unless elongated clusters are suspected and CLINK because of its tendency to produce compact, but not necessarily well separated clusters. They recommend UPGMA. Romesburg (1984) also favours UPGMA and Lorr (1983 p. 101) agrees with this recommendation. Accordingly, this study used UPGMA, where appropriate, for most of the clustering runs.

3.7.3. Interpretation of the Clustering Results

The clusters obtained by analysing the hierarchical dendrograms and Kmeans output still required interpretation. Two hundred and fifty different sets of clusters were obtained, one from each run. A decision had to be made whether to retain or reject each of these clusterings. This could not be achieved based on 'correctness' or 'the right model'. Anderburg (1973, p. 23) suggested that this was not the type of problem where there was an optimal solution as in linear programming. Heuristics and researcher intuition had an important part to play in arriving at a solution:

> The mechanical results derived from submitting a set of data to some cluster analysis are themselves devoid of any inherent validity or claim to truth; such results are always in need of interpretation and are subject to being discarded as spurious or irrelevant . . . .The use of cluster analysis requires the active participation of the analyst to interpret the results and judge their significance. This stage of the process is subjective, intuitive and heuristic. (Anderburg, 1973, p. 176)

The skill, insight, experience and subjective judgement of the taxonomist had an important part to play:

> These methods (cluster analysis) are best seen as tools for data exploration rather than for a production of a formal classification . . . . one cannot replace careful thought by automated computer methods. (Dunn and Everitt, 1982, p. 105)

Many clusterings were produced, all seemingly valid but some more intuitively useful than others. Clifford and Stephenson (1975, p. 125) suggest that it is up to the researcher to choose which cluster :s most suitable. The criteria used for accepting the clustering solutions were those laid out in section 1.4.2 dealing with the secondary research goals of achieving well structured and intuitive clusters which could be used to achieve the primary research goal of producing a special purpose taxonomy of spreadsheet application development.

An additional criterion for acceptability, was the agreement between solutions provided by the Kmeans and hierarchical algorithms. As both methods forced a clustering solution on data, whether it was homogeneous or not, the outcome of 'no clusters present' was never an available option. If two different algorithms gave similar results, there was an indication that clusters were really present and modelled the underlying structure of the data. The clustering was likely to be 'real' rather than an artefact of a particular algorithm (Dubes and Jain, 1979).

3.8. The A.D.E. Taxonomy

This taxonomy was evolved for use in categorising the spreadsheet application development process. It was developed in three parts.

a) A the <u>Application</u>

b) D the <u>Developer</u>

c) E the development <u>Environment</u>

<u>3.8.1. Development of the Taxonomy</u>

Each of the three parts of the taxonomy was designed separately, using the clustering run that was considered the most suitable, considering the criteria outlined above in sections 3.7.2 and 3.7.3.

The tree dendrogram output of the SYSTAT JOIN ROWS procedure was transected by a line chosen to divide the tree into appropriate clusters as described in section

3.6.2 and Figure 3.9. As the graphical shaded density matrix output of the SYSTAT JOIN MATRIX procedure had been sorted so that its rows were in the same sequence as the dendrogram, the allocation of cases into clusters could be copied from the dendrogram.

In the graphical shaded density matrix, dissimilarity/similarity coefficients were replaced with symbols that were shaded to give an impression of their magnitude. A 'profile' of each cluster was then visually apparent. The variables having least variability within the cluster and most variability between this cluster and other clusters could be visually identified.

The cluster profile was finalised by examining both the statistics produced as part of the Kmeans output, and the matrix cluster density plot from the SYSTAT MATRIX clustering. The cluster name was suggested by its profile. After all clusters had been identified and their profiles constructed and named, the A.D.E. taxonomy was packaged:

a) The named clusters were rearranged in a hierarchical manner to form a section of the taxonomy.
b) The three sections representing the Application, Developer and Environment were combined.
c) Codes were provided for each class.

3.8.2. A Diagnostic Key for the A.D.E. Taxonomy

The diagnostic key, for use in assigning a spreadsheet application development project to its three categories within the taxonomy was developed in three separate parts for the three sections covering the Application, Developer and Environment. A decision tree was prepared for each section. A user had only to follow each question through the three decision trees to arrive at the appropriate three A.D.E. codes that categorised their project. The diagnostic keys were designed to minimise the branches of the decision tree i.e. the number of questions required.

3.8.3. Validation of the A.D.E. Taxonomy

The taxonomy was validated with respect to the goals of this research laid out in Chapter 1 and also with respect to criteria established in reports in the literature. The rationale and methods for validation of the taxonomy and its diagnostic key are described in detail in Chapter 5.

3.9. Assumptions and Limitations of this Study

Underlying assumptions

Several assumptions have been made in this study:

a) It was assumed that respondents had the ability to report accurately and had in fact done so!

b) It was assumed that the spreadsheet development environment is not homogeneous but heterogeneous i.e. there are different classes of spreadsheets, developers and development environments. The validation exercises described in Chapter 5 go some way towards confirming this assumption.

c) It was assumed that the attributes chosen were suitable to develop a taxonomy for use in the design and control of spreadsheet projects.

d) Finally it was assumed that in the absence of a sampling frame, the sampling procedures did choose a sample of cases that represented the population of all spreadsheet developers sufficiently adequately to allow for the development of a special purpose taxonomy for use in the control of spreadsheet application development.

Limitations

The primary limitation of this study was the non-generalisability of the results due to the non-probabilistic sampling methods used.

The use of two measurement instruments of unknown validity also limits the generalisability of the results however attempts were made to establish the validity of these data collection instruments.

The A.D.E. has been designed for use in the management and control of spreadsheet development projects. i.e. it is a special-purpose taxonomy rather than a general taxonomy. This limits the general applicability of this taxonomy but makes it much more appropriate for the use for which it is intended.

3.10. Ethical Considerations

The researcher was mindful of ethical considerations when conducting this research. These reflected the rights of society as a whole and of the subjects in particular. Efforts were made to ensure the maintenance of the rights of all involved directly or indirectly in this study, based on the framework of major ethical relationships in business research evolved by Davis and Cosenza (1985, p. 457).

Societal rights

As research exists within society and is nurtured by it, it has certain responsibilities towards society. Society has a right to be informed of any outcome of this research that may effect its health and well being (Davis and Cosenza, 1985, p. 457). In this respect, society could be considered, either as the Australian population as a whole, or spreadsheet developers and those who are responsible for managing them, in particular. Their rights will be supported with the publication of the more significant results of this study.

Society can also expect objective, complete, unbiased and scientifically sound research results. (Davis and Cosenza, 1985, p. 456). This study was neither completely objective nor unbiased. It would not have taken place if these criteria had been immutable, however the bias and lack of objectivity have been clearly identified as has their effect on the generalisability of the results.

Subjects' rights

Subjects had the right to receive adequate information to allow them to make an informed choice whether to participate in the study or not. They had the right to refuse participation without any adverse consequences. The sampling procedures respected these rights.

Subjects had the right to ask for and receive results of the study if requested. Copies of the results were sent to those who requested them.

Subjects had the right to have consideration given to their busy workload and appreciation for the time taken to cooperate in this project. The questionnaire design tried to make response as easy as possible. The follow-up procedures were designed to be polite and unobtrusive as well as effective. Respondents' contributions were always valued by the researcher and they were thanked for their cooperation.

Finally, subjects had the right to expect that assurances of anonymity would be respected and their privacy guaranteed. To achieve this goal, the subjects contact details were not held in the electronic databases and were removed from the original questionnaires and replaced with a number. The corresponding list of names and numbers was kept under lock and key until the end of the study when it was shredded.

Researcher's rights

Given that the researcher was acting ethically, she had the right to expect reciprocal behaviour from the respondents. This primarily involved "the reporting of data as truthfully and unbiased as possible as long as it does not conflict with some other highly held ethical value or principal of the individual" (Davis and Cosenza, 1985, p. 463). This was in part beyond the researcher's control. However procedures were put in place to make it simple for respondents to report accurately and to identify cases where this might not have been the case.

3.11. Summary of this Chapter

This chapter has described in detail the study methodology and design and the rationale for the choices made.

The sampling process, questionnaire design, validation and submission were described. The data coding, screening and data structures for analysis were detailed together with the development of suitable variables for input to the clustering process.

The Kmeans and hierarchical clustering algorithms were described with their variable input parameters. A series of clustering runs was developed leading to the formation of the three part A.D.E. taxonomy of spreadsheet applications development and its diagnostic key.

The chapter ended with attention to some ethical considerations.

CHAPTER 4: RESULTS

4.1. Overview of this Chapter

This chapter documents the results of this study. Supporting material can be found in Appendix C, D and E.

The sample is described, including return statistics, and the identification of possible outliers. Graphs are drawn to illustrate the sample composition, and some interesting results are reported.

A series of computer cluster analysis runs is described, together with their variable input parameters and output clusterings. A taxonomy of spreadsheet application development is developed from these runs, together with a diagnostic key used to place a spreadsheet development project within the taxonomy.

4.2. The Sample

The sample was drawn in three parts using the multi-stage stratification sampling plan outlined in 3.4.2: a) Preston, b) Perth Metropolitan and c) Eastern States.

4.2.1. Sample Responses

Two hundred and sixty eight questionnaires were distributed between September and November 1991. Twenty five identifiable cases were followed up for non-response. By December 1991, one hundred and eight replies were received.

Table 2:

Spreadsheet Survey: Questionnaire distribution and response

	Preston	Perth Metropolitan	Eastern States	Total
Dispatched	85	142	40	267
Responded	65	33	10	108
Response rate	76.5%	23.2%	25.0%	40.5%

As described in sections 3.5.1. and 3.5.2., the sample responses were initially scanned by eye and then coded and entered into the databases. Variables were transformed and data structures generated as outlined in sections 3.5.3. - 3.5.5.

4.2.2. Data Screening

The data screening methods used were discussed in section 3.5.8. Reasonableness checks using SQL were carried out on the database. Bar graphs (see Fig. 4.1) and / or Box Plots (see Fig. 4.2) were drawn for appropriate variables to assess possible outliers, incorrect codes and other anomalies

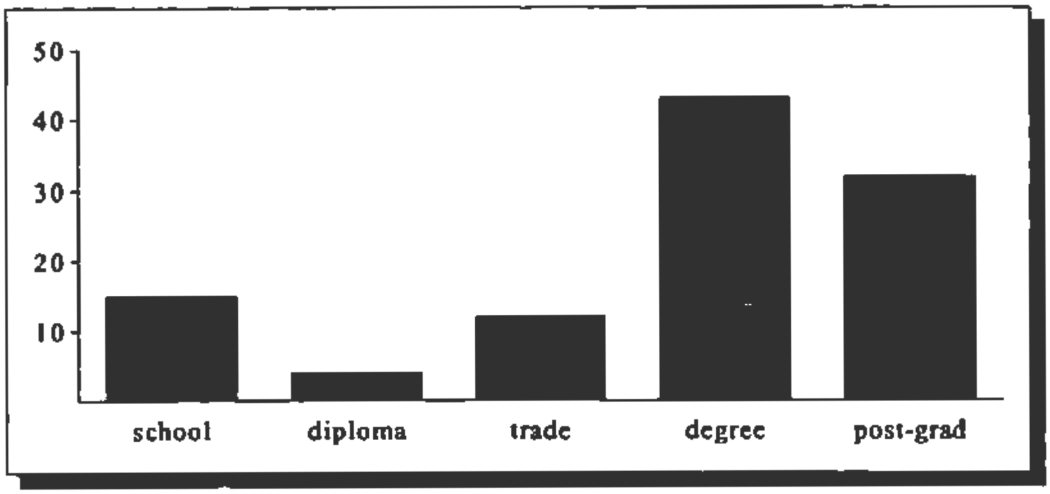


Figure 4.1: Spreadsheet survey: Bar graph showing the distribution of cases by value of the variable QUALIFY.

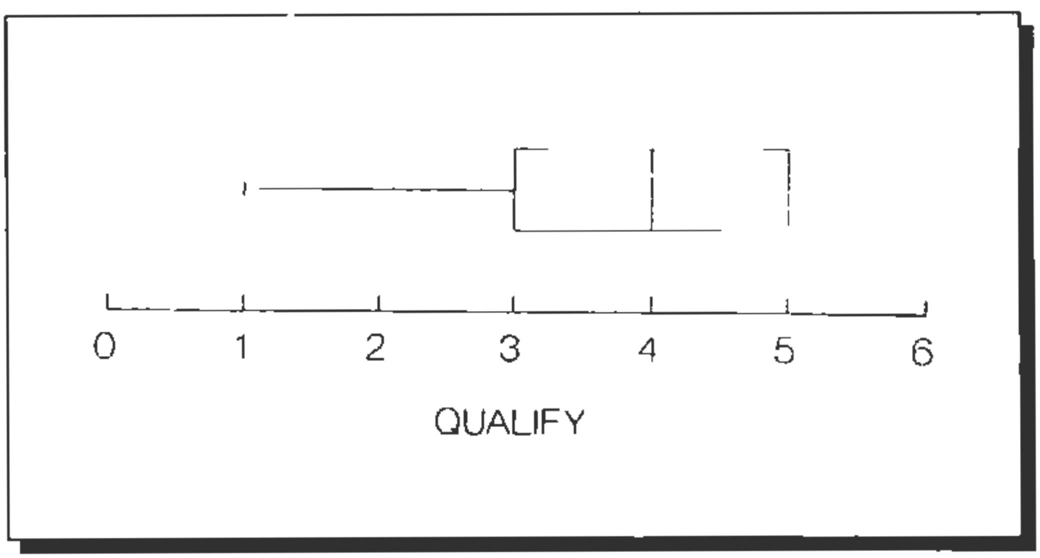


Figure 4.2: Spreadsheet survey: Box plot showing the distribution of values of the variable QUALIFY .

Contingency tables (see Table 3) were calculated for all variables and assessed for plausibility.

Table 3:

Spreadsheet survey: Contingency table showing the distribution of values for the variable QUALIFY, the highest level of qualification attained by survey respondents.

	1	2	3	4	5	Total
Frequency	15	4	12	43	32	106
Percentage	14.15	3.77	11.32	40.57	30.19	100

4.2.3. Missing Value Treatment

Missing values were treated as described in Section 3.5.8. If the respondent could not be contacted these were usually replaced by the character 'space', recognised by SYSTAT as a missing value.

The major question that caused respondents difficulty when completing the questionnaire, was the question on the variable SIZE, used to record the 'raw' spreadsheet size in bytes. This question was either unanswered or dubious in 22% of returns. The assumption was made that respondents were either unwilling to use their computers to determine the answer to this question or did not know how to obtain the answer. This was verified on follow-up discussions with respondents by telephone. Other respondents may have guessed the answer to this question. The spreadsheet SIZE.SSF was used both to check the plausibility of spreadsheet 'raw' size (prior to transformation) and to estimate it, if necessary, when it was impossible to contact the respondent. A listing of part of this spreadsheet can be found in Table 25 in Appendix C.

4.2.4. Outlier Identification and Removal

The variable SIZE recorded the original size in bytes of the spreadsheet prior to any transformation. Both a normal probability plot (Fig. 4.3) and a box plot (Fig 4.4) were drawn for the variable SIZE. These plots showed SIZE was not normally distributed but was skewed to the right.

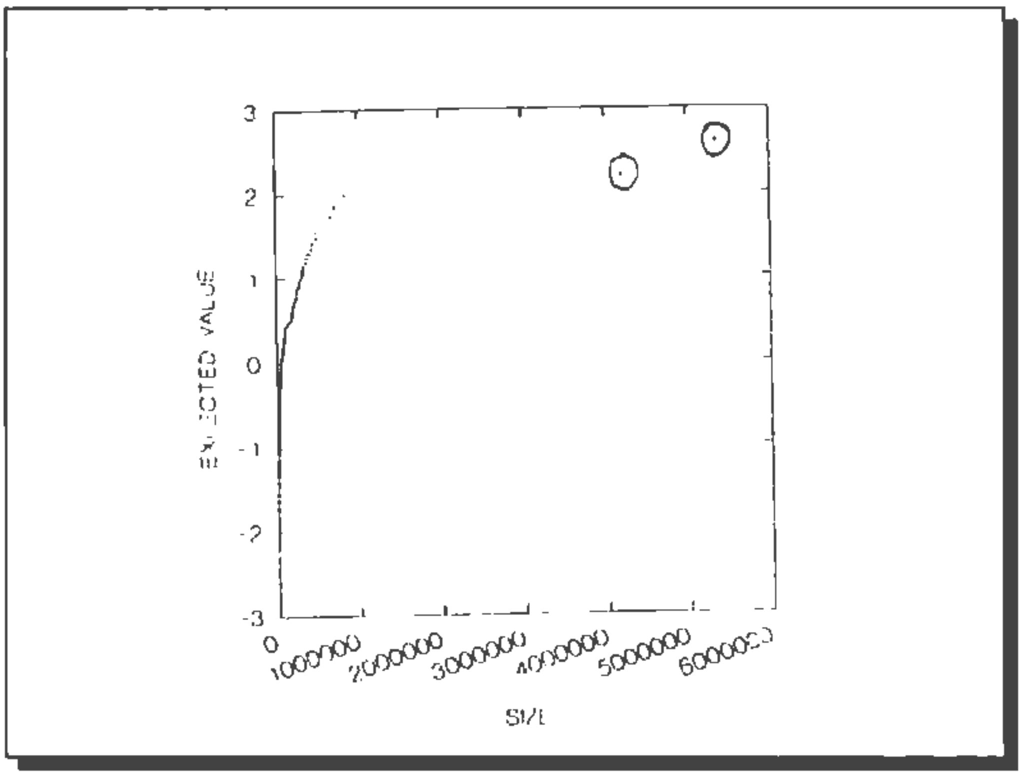


Figure 4.3: Spreadsheet survey: Normal Probability Plot of the Variable SIZE. The plot is not a straight line as SIZE is not normally distributed. Two outliers are clearly visible.

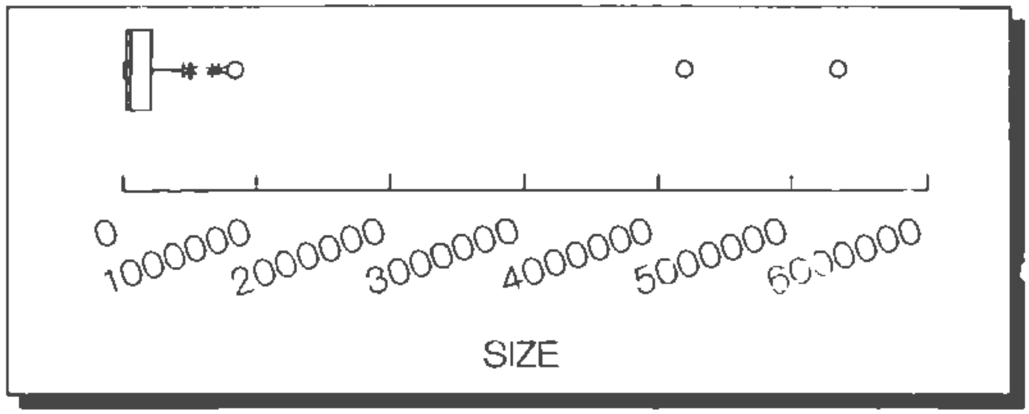


Figure 4.4: Spreadsheet survey: Box plot for the variable SIZE showing skewness to the right and three possible outliers.

Three possible outlier cases were identified. After discussion with one of the respondents and in the unavailability of another, it was decided to remove cases 15 and 108 from the sample. The other possible outlier was retained as it was not so anomalous as the other two, however the value of its SIZE score was reduced by ten percent. The researcher felt that this case could belong to a minor, but plausible, category representing very large, computationally simple, spreadsheets. This category would have been unrepresented if the case had been removed.

Ordinal Variables

The standardised scores of all ordinal variables were examined to identify those with values outside three standard deviations from the mean. Seven variables had occasional cases with values outside this range: STCONS, ICOMP, POPTIM, SPERSN, SDPOLDC, SDENFORC and THREED. It was decided to leave these variables and the anomalous cases in the data-set, as all seven variables were in fact binary dichotomous with only two possible values. The retention of the rarer attributes could well assist in identifying categories in the final taxonomy.

Binary Dichotomous Variables

The scores of binary dichotomous variables are presented in Table 27 in Appendix C. The table was scanned and variables with either score having a frequency of less than 10% were reconsidered. Some cases had frequencies of less than 10% in some of the variables describing occupation. IMANUF, IELECT, ICONST, ISELL, ICOMP, IOTHR had less than 10% of all cases with a value '1'. These variables were removed from the analysis as their presence would have had a high influence on the distance measures inappropriate to their importance as identifiers of clusters.

PCLASS describing spreadsheets with a primary purpose of classification also had less than 10% of cases with a score of '1'. This variable was combined with PCOMMS and PREPORT to form the new variable PREST.

SPERSON and SDDOCO had similar low frequencies but were retained in the data-set as their importance warranted.

4.2.5. Sample Descriptive Statistics

After data scanning and clean up processes, one hundred and six cases were retained in the sample. Ordinal and binary dichotomous data-sets were prepared for these cases and input to the SYSTAT software where they standardised to a mean of zero and a standard deviation of one, in effect making them dimensionless.

Developer Profile

Variables measuring respondents stratum, age, gender, professional memberships and industry were not used in the clustering runs. They served however to show the variation within the sample. Other variables used to describe developers such as organisation size, employment status, educational qualifications, user-group membership, training and reading spreadsheet articles were used for clustering.

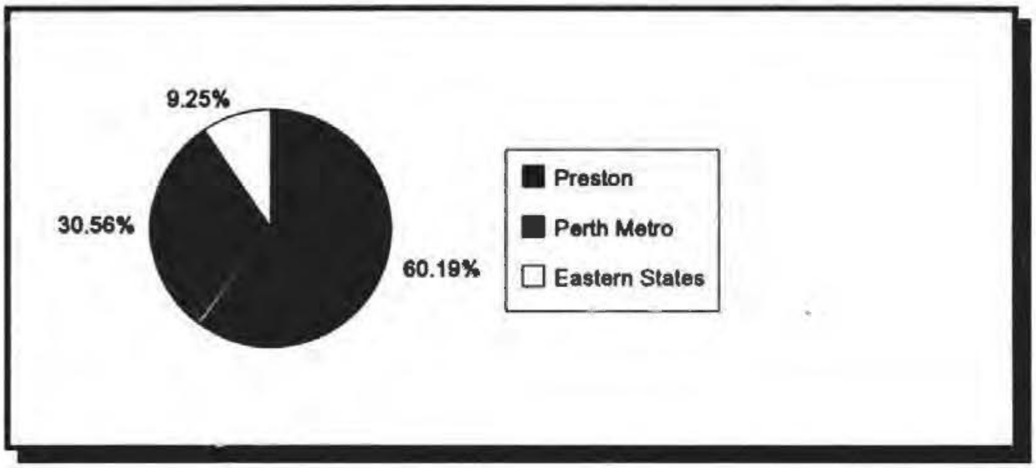


Figure 4.5. Spreadsheet survey: Developers by stratum.

Preston made up the bulk of the sample (60%), 10% were from interstate and the remainder from Perth.

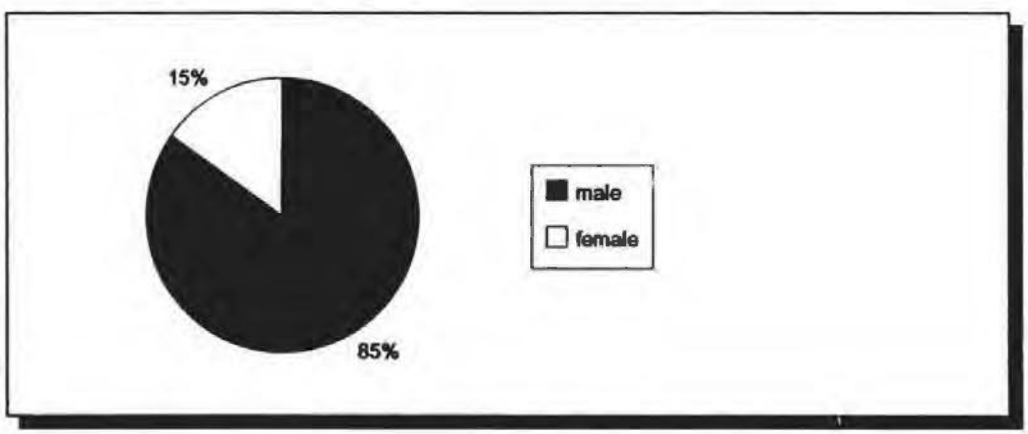


Figure 4.6. Spreadsheet survey: Developers by Gender

Most survey respondents were male. Only 15% were female.

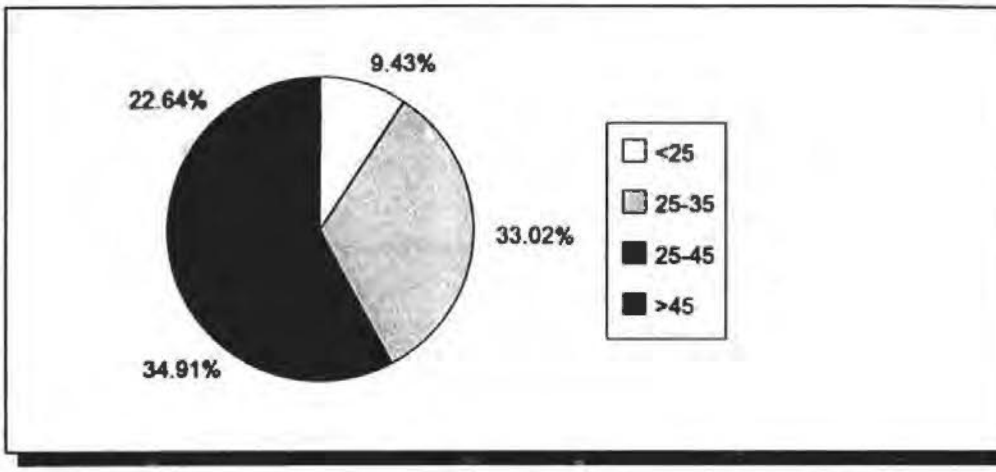


Figure 4.7 Spreadsheet survey: Developers by Age

Less than 10% of the sample respondents were under twenty five years and 58% were older than thirty five.

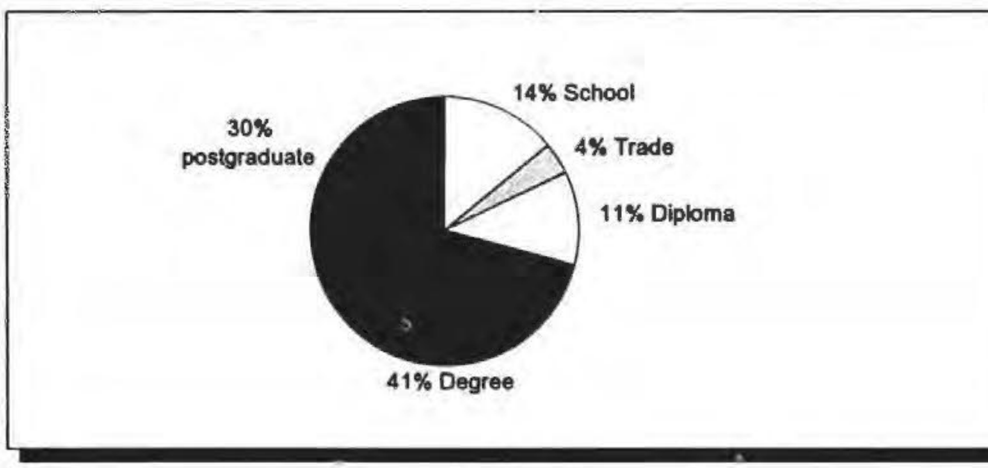


Figure 4.8 Spreadsheet survey: Developers' highest qualifications

The respondents were well qualified with 71% having a degree or post-graduate qualification. 51% had membership status in professional organisations.

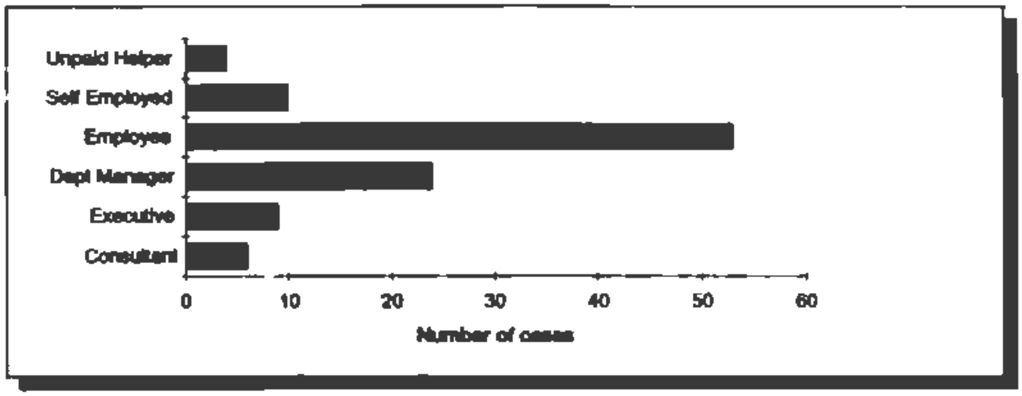


Figure 4.9. **Spreadsheet survey: Developers' employment status**

About half the respondents classified themselves as employees rather than management, yet Figure 4.7 shows 58% were older than 35, and Figure 4.8 shows 71 % had degrees or post-graduate qualifications.

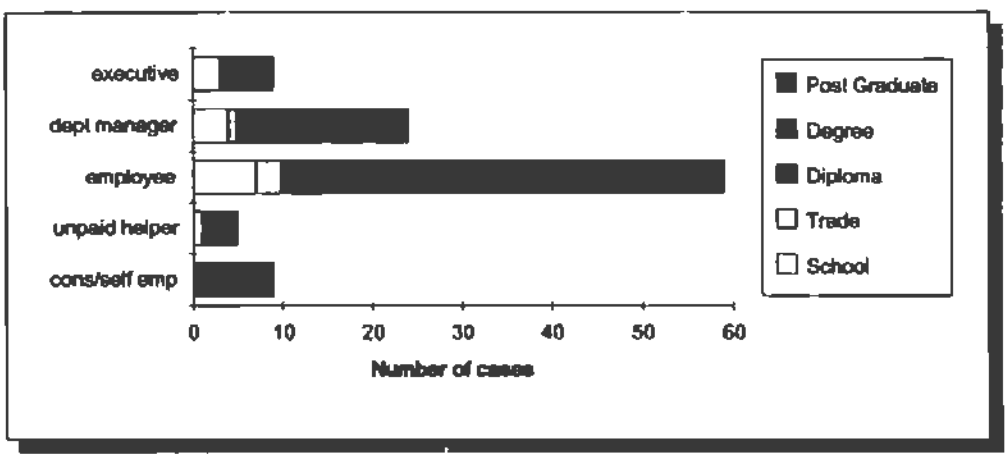


Figure 4.10 **Spreadsheet survey: Developers' employment status and highest educational qualification.**

The respondents who classified themselves as employees had a high rate of degrees and post-graduate qualifications, combined with their non-managerial status. They presumably were well qualified, technically capable, competent people working possibly independently, designing and building spreadsheets in uncontrolled environments without the overall picture of the organisation that someone with managerial status would have had. A situation worthy of some attention, when considering the control of spreadsheet development.

45% of the developers worked for small, single person or one department organisations, 13% for medium sized, multi-department, one site organisations and 42% worked for large organisations with many departments on more than one site.

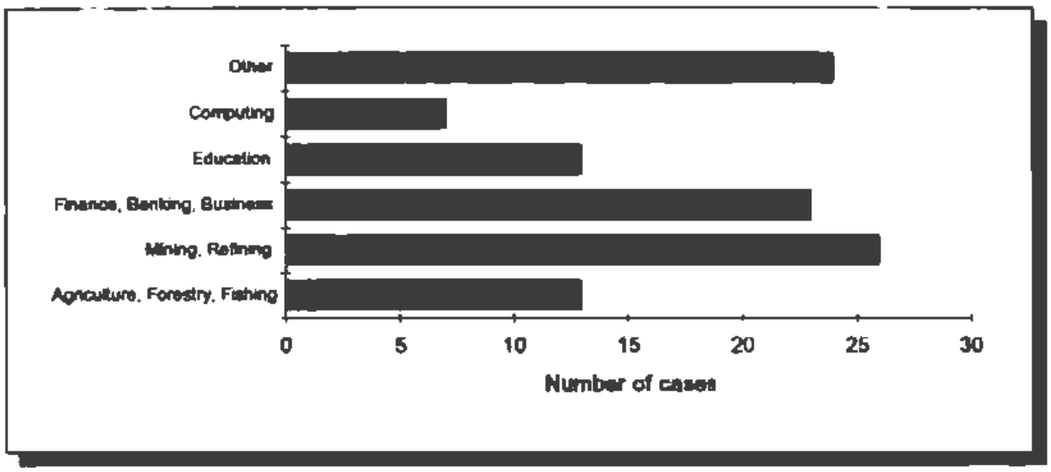


Figure 4.11 **Spreadsheet survey: Developers by Industry**

As might have been expected from the distribution of industries in Preston, the largest stratum (see Figure 3.3), about 25% of the respondents were employed in the mining industry. The farming, forestry and fishing industries also had high representation. Business, finance and banking accounted for another 22%. The computer industry had only a small representation of 7% i.e. 93% of the spreadsheets surveyed were developed outside the computer industry. Most of the developers worked in the private sector with only 5% private or recreational development.

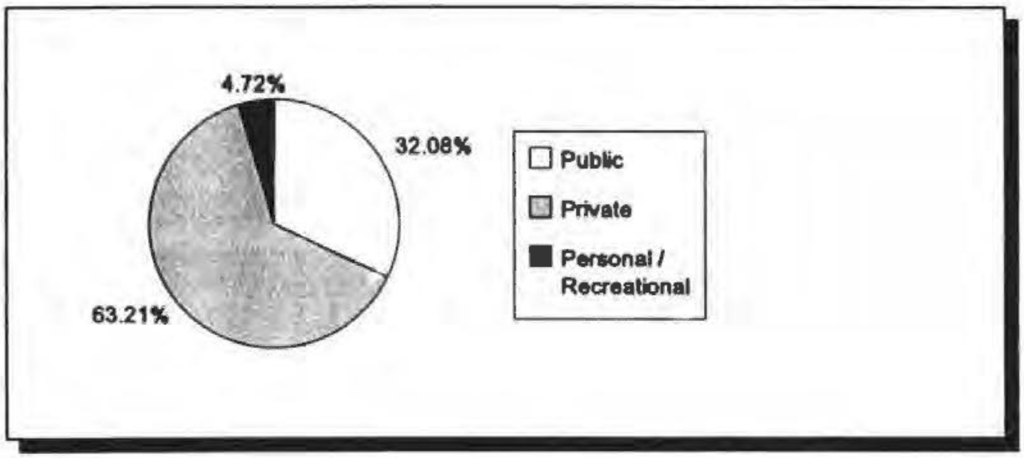


Figure 4.12. Spreadsheet survey. Respondents by sector

Developers had varied interest in spreadsheets, the majority not appearing to have high interest. 11% belonged to a spreadsheet user-group and these developers presumably did have a considerable interest in spreadsheets.

The number of articles read concerning spreadsheets, was considered as another sign of spreadsheet interest. The majority (60%) of developers in the sample read less than three articles about spreadsheets in a year, however 21% read more than eight articles on spreadsheets and could be presumed to have an interest in spreadsheets.

The training received in developing spreadsheet models also varied. A high 52% were self trained and 8% were trained solely by work-mates. The remaining 40% were divided evenly between those who had attended courses and those who considered they had professional data processing training.

Software Profile

The variables describing the brand of software and operating system, were not used for clustering. A broad range of software packages was represented.

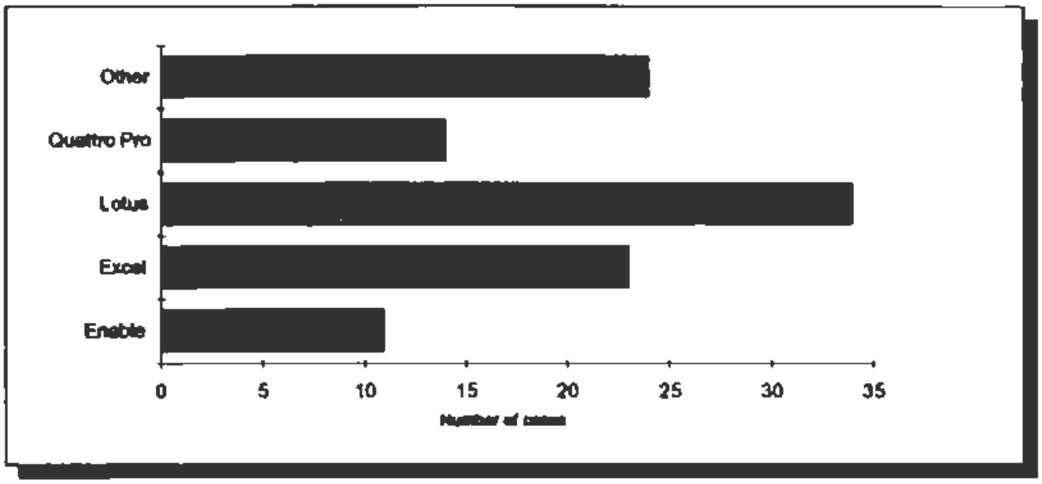


Figure 4.13. Spreadsheet Survey: Software used for development

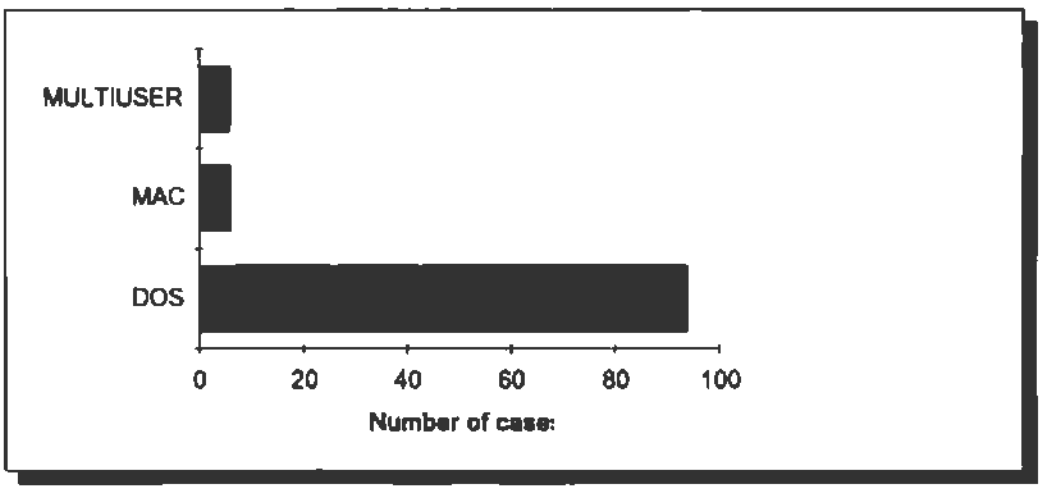


Figure 4.14. Spreadsheet Survey: Operating System used.

DOS and its many variations was the predominant operating system, used in over 90% of cases. A few developers worked with an Apple Macintosh or in a multi-user environment on mainframes, or minis running PICK or UNIX. OS/2 was not represented. The DOS figures included developers who specified that they were using Microsoft Windows 3.0. running as a DOS shell.

Control Profile

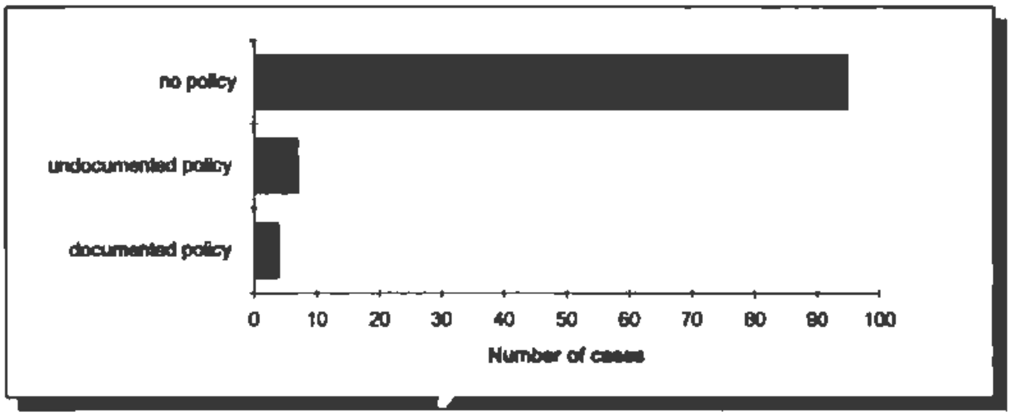


Figure 4.15. Spreadsheet Survey. Awareness of control policy.

There was minimal control of spreadsheet development in the respondents' parent organisations. Only 11% of developers were aware of a spreadsheet control policy within their organisation, with one third of these having a documented copy.

If the policy was enforced, it was self enforced in more than half of these cases, and in only one case in the sample, was there any reported involvement of the I.T. department. No respondent reported auditor enforcement of the policy.

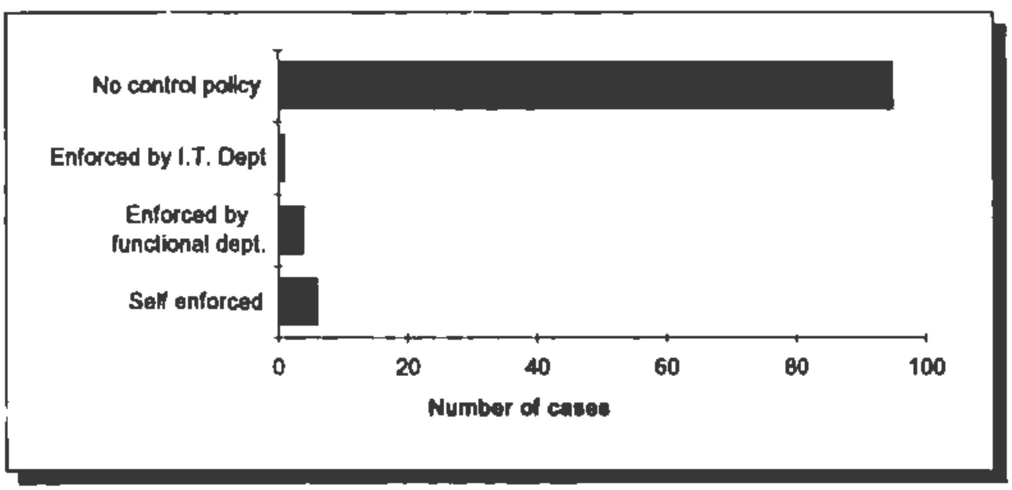


Figure 4.16 Spreadsheet survey. Enforcement of spreadsheet control policy.

6% of the total number of respondents, who were otherwise working in a non controlled development environment, did have access to spreadsheet libraries of supposedly quality templates. These examples, if they were indeed of quality and used wisely, could have impacted on the control of spreadsheet development for these respondents.

Another aspect of control, is the provision of sufficient time for the adequate completion a spreadsheet development project. 18% of the respondents noted that their projects were rushed and they would have preferred to have had more time available.

The overall level of control of spreadsheet development projects was low in this sample.

Spreadsheet Survey: Application Profile

Notwithstanding the lack of developmental control outlined in Section 4.2.5., most of the spreadsheets in the sample had a non-trivial and even important usage.

The spreadsheet applications were used for a variety of purposes, the most common being report generation. Nearly 70% of the applications were involved with some type of reporting. The remaining 30% of the spreadsheets were used to create models to assist decision making. Forecast or prediction models accounted for 18% of the total and there were a few 'what if' and optimiser models.

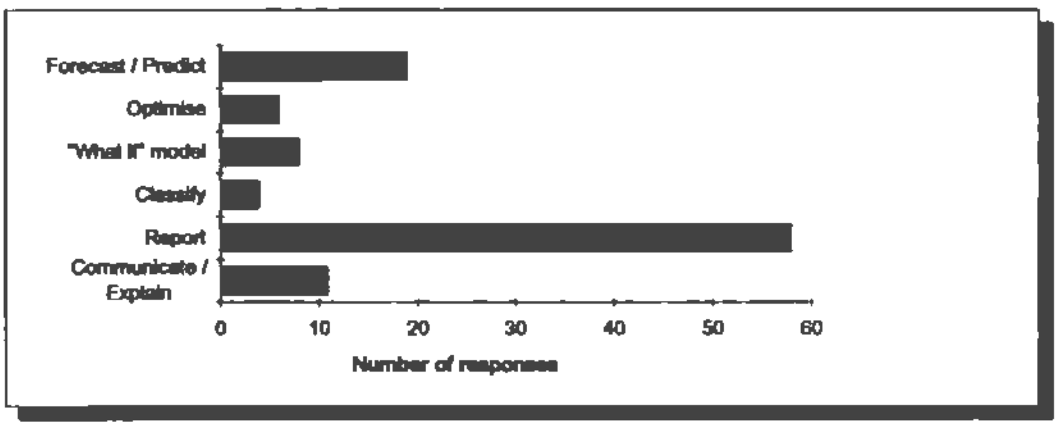


Figure 4.17. Spreadsheet survey. Spreadsheet purpose.

The spreadsheets were used for important objectives, and most respondents (92%), classified their application as being of moderate or major importance. This was confirmed by the proportion of spreadsheets that either modified existing Corporate data (27%) or created new Corporate data (49%). 40% of the spreadsheets in the sample had no involvement with Corporate data.

The importance of the majority of the spreadsheets was also confirmed by the distribution of their output. Only 17% of the spreadsheets were solely for the developer's own use, and the output of the remainder was distributed to others. 29% of the total sample was distributed beyond the developer's organisation.

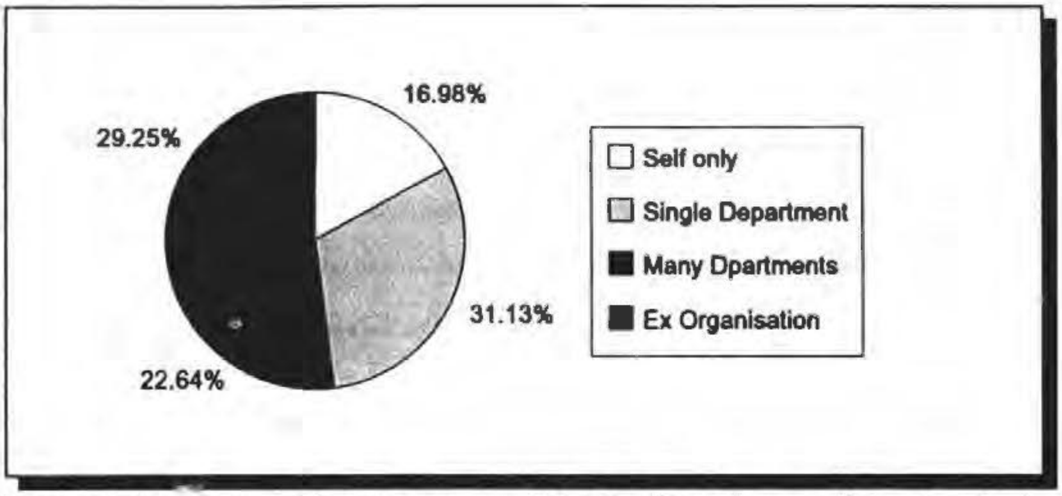


Figure 4.18. Spreadsheet survey. Distribution of spreadsheet output

Most of the spreadsheet output remained in circulation for some time, with more than half (55%) remaining in use for longer than a month.

Most (67%) of the spreadsheets were run on a regular basis (daily, weekly, monthly or frequently), and a smaller proportion (17%), was used once or only a few times. The remaining 16% were run occasionally after long gaps in time. These spreadsheets were of particular interest from a control perspective, as they could have been used as a basis for important decision making, by users unfamiliar with the infrequently run template.

Most of the spreadsheets were intended to be run solely by their developer, but 18% were prepared for other users to run and 10% for data entry by clerical assistant.

Spreadsheet Survey: Template Profile

There was a large variation in the size and complexity of the spreadsheets. Size ranged from 800 bytes to 5.3 megabytes. The mean spreadsheet size was 218 kilobytes. Spreadsheet size was not normally distributed (See Figure 4.3 normal probability plot) and was skewed to the right i.e. showing a predominance of larger spreadsheets.

Complexity was considered in three parts design, logical and link:

a) Design complexity was shown by the use of borders, split screens and modular design.

b) Logical complexity was shown by the use of both absolute and relative referencing, @IF functions, look-up functions and formulas.

c) Link complexity was shown by links to templates and other non spreadsheet software, graphics and macros.

Spreadsheet Design Complexity

The spreadsheets sample did not show as high a design complexity as might have been expected. 25% of spreadsheets used split screen techniques and 49% had fixed borders incorporated within their design.

Exactly half the spreadsheets had a modular design. As defined in Figure 4.19 below, 38% of spreadsheets had a blocked, and 12% a diagonal modular shape. It is interesting to note that half of these predominantly large spreadsheets were not designed in a modular manner.

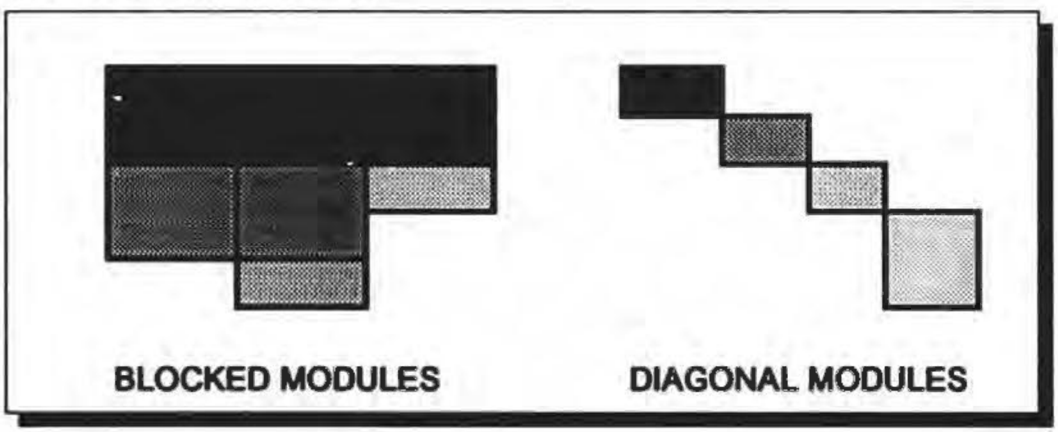


Figure 4.19 Modular Spreadsheet Designs

The comparison of the size of a spreadsheet with modular design shown in Figure 4.20, shows that this tendency to non-modular design was not restricted to smaller spreadsheets.

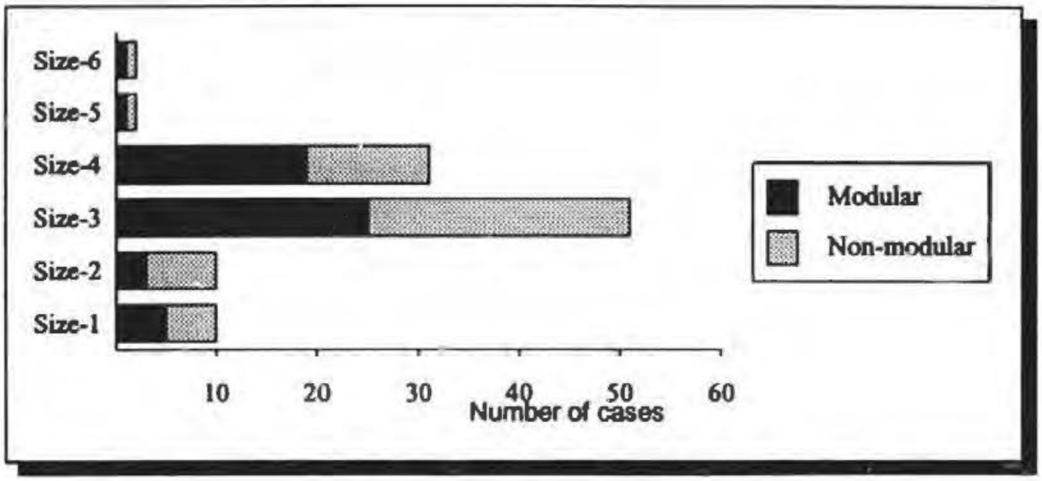


Figure 4.20: Spreadsheet survey. Comparison of modularity of design with spreadsheet size categories ranging from size-1, small to size-6, large.

Spreadsheet Logical Complexity

The logical complexity of the spreadsheets surveyed was non-trivial. 66% of the spreadsheets used both absolute and relative referencing. 47% of the spreadsheets used logical @IF functions and the function was nested in over half of these (27% of the total sample). Look-up functions and tables were used in 27% of the responses.

In over half of the cases (57%), the developer categorised the formulas used as average or complex.

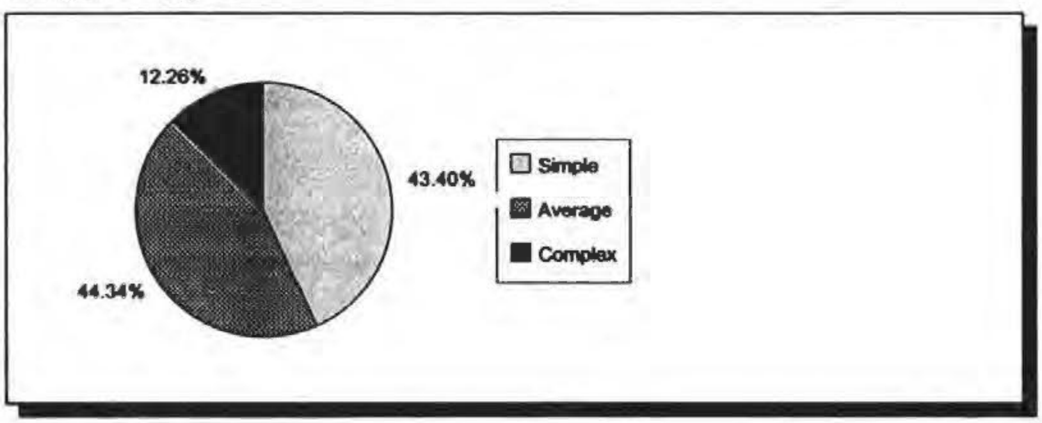


Figure 4.21 Spreadsheet survey: Formula complexity

__Spreadsheet Link Complexity__

The link complexity of the sample was also non-trivial. 36% of the sample had links to other spreadsheets and 21% involved links with a database. 8% involved Windows D.D.E. (Dynamic Data Exchange).

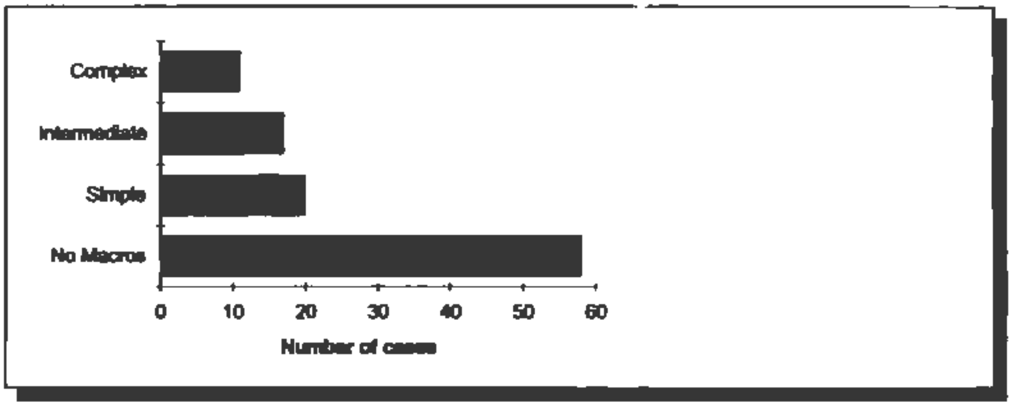


Figure 4.22. Spreadsheet survey: Use of Macros

Macros were used by 45% of the spreadsheets but only 10% of respondents considered their macros complex.

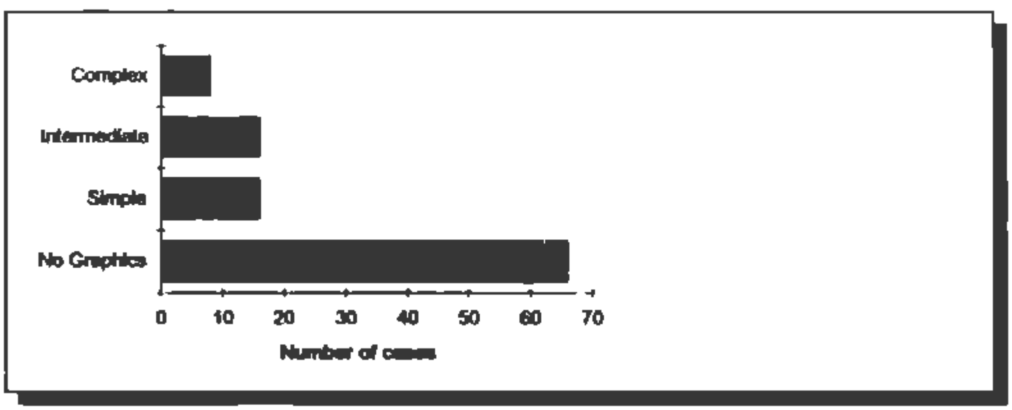


Figure 4.23. Spreadsheet survey: Use of Graphics

Graphics were slightly less common, featuring in 38% of the spreadsheets. 8% of the total sample respondents considered their graphics to be complex

4.3. Clustering Runs

A series of clustering runs was carried out using the SYSTAT software. Data scales were varied (binary dichotomous or ordinal). Data attribute selection and weighting were varied. The clustering algorithm was varied (hierarchical joins or Kmeans partitioning with variable number of clusters). The linkage was varied. (single, complete, centroid, average, median and Ward) The distance measure was varied. (PCT, Gamma, Pearson, Jaccard, Mu-2, Rho, Tau and Euclidean). Runs were grouped, with each new group testing some major change in the clustering input parameters. A summary of the parameter variations for each run can be found in Table 28 of Appendix D.

The rationale for the strategy used is outlined below. One hundred and fifty cluster analyses were performed.

- Eighty four to experiment with parameters usage in the clustering algorithms.
- Twenty six to develop the Spreadsheet Developer categories of the A.D.E. taxonomy
- Thirty one to develop the Spreadsheet Application categories of the A.D.E. taxonomy
- Nine to develop the Environmental categories of the A.D.E. taxonomy.

4.3.1. Experimental Runs To Select Parameters For Production Runs

The objective of these initial 84 runs was largely experimental. The SYSTAT computerised implementation of the algorithms was investigated using the survey data, and clustering parameters were trialed and selected for use in the final analyses to generate the clusters from which the taxonomy was derived. Experimental cluster analyses were carried out using binary dichotomous, ordinal and mixed scales, six different linkage methods and ten different similarity or distance measures. Details of these runs and the rationale behind the selection of their parameters can be found in Appendix D and Table 28.

On the basis of these experimental runs, it was decided that ordinal scaled variables using an Euclidean distance measure and both the Kmeans and hierarchical joining algorithms with average linkage (U.P.G.M.A.), offered the best route to find clusters suitable for building a taxonomy.

4.3.2. Production Runs For The Developer Categories Of The Taxonomy

These runs used the standardised ordinal data-set with average linkage and Euclidean distances for creating hierarchical tree dendrograms joining rows and Kmeans for partitioning. They varied the attributes selected and their weighting.

The nine group 18 clustering runs investigated the weighting of variables EXPERT and XTRAIN describing spreadsheet developers' expertise and training. A easily identifiable clustering solution was obtained with excellent agreement between KMEANS and JOIN algorithms. User-group members and self-employed persons separated out into clearly separated clusters.

The final seventeen runs used to cluster developer attributes investigated the effect of the XSTATUS variable on the clustering. Consultants and self employed persons had an XSTATUS of 0 (less than the XSTATUS of an employee) and it was felt that this did not reflect a true measure of status. Each of the cases where XSTATUS was 0 was re-examined in the light of the respondent's answers to other questions and follow-up telephone interviews where necessary. In 60% of the cases the coding of the XSTATUS variable was upgraded from 0.

TABLE 4:

Spreadsheet survey. Changes to XSTATUS variable for self-employed persons and consultants.

STATUS	CASE	NEW XSTATUS
SELF-EMPLOYED	15	3
SELF-EMPLOYED	46	2
SELF-EMPLOYED	78	2
SELF-EMPLOYED	79	2
SELF-EMPLOYED	101	2
CONSULTANT	100	1
CONSULTANT	25	2
CONSULTANT	76	2

Variables representing self-employed (STSELFEMP) and consultant (STCONS) status were included with the developer variables clustered. These two additional variables compensated for the changes made to the XSTATUS variable. Compact, well separated clusters were obtained, however CASE 15 was identified as a possible outlier as it formed a one-member cluster with a very late joining with the remaining clusters. This case was reinvestigated and a decision was made to drop it from the analysis as the validity of much of its data was in doubt.

The later group 20 runs were the final runs used to identify developer clusters. These runs weighted expertise (EXPERT) three hundred percent but did not weight training. Occupation as a data processing professional (OIT) was included, but not working in the computer industry (ICOMP).

The following variables were used to produce the dendrogram:

- ORGSIZE - Size of the user organisation
- USERGRP - User-group membership
- EXPERT - Developer expertise
- WTEXP1 - Developer expertise
- WTEXP2 - Developer expertise
- XTRAIN - Spreadsheet training
- READ - Reading concerning spreadsheets
- QUALIFY - Academic and other qualifications
- PROFMEMB - Membership of a professional body
- XSTATUS - Status in the work-force
- STSELFEM - Self employed
- STCONS - Working as a consultant
- OIT - Occupation in I.T.

The hierarchical JOIN run 20m (with ten clusters and with the biggest cluster further subdivided into two unequal parts) was compared with KMEANS for 14 clusters in run 20q. An almost perfect match was obtained of clusters derived from the two methods when two groups of small clusters were combined leaving only case 53 assigned to different clusters by the different algorithms. Run 20r analysed a matrix clustering to assist in the identification of the cluster profiles. Copies of these final runs for clustering of the developers' variables can be found in Figures 7.3 and 7.4 and table 29 of Appendix D. The following ten clusters were identified:

- C1 I.T. professional spreadsheet expert consultants (Spreadsheet Gurus)
- C2 Other I.T. professional consultants not spreadsheet experts
- C3 Spreadsheet consultants but not I.T. professionals
- D1 User group members
- D4 Novice developers

- D3 Knowledgeable developers
- D2 Lay experts
- I2 Non consultant I.T. professionals interested in spreadsheets
- I1 Non consultant I.T. professionals disinterested in spreadsheets
- D5 Self-employed developers

4.3.3. Developer Cluster Profiles

A cluster profile described the attributes that lead to within cluster homogeneity and between cluster heterogeneity i.e. the effect the variability of attributes had on the clusters generated. Cluster profiles were developed for each of the ten clusters by an analysis of the row and matrix join clustering outputs and a comparison with the Kmeans output. Copies of all relevant SYSTAT outputs can be found in figures 7.3 and 7.4 and table 29 of Appendix D.

The clusters were identified by transecting the tree dendrogram from the row clustering output at a suitable distance resulting in the identification of ten clusters. The largest cluster was further sub-divided into two clearly separate groups and two of the smaller groups were combined. This division into clusters was then superimposed on the shaded matrix output. Correspondence with the Kmeans clustering output was established. Profiles of cluster membership were developed, considering both the shaded matrix output and the cluster means and standard deviations on each variable from the Kmeans analysis.

C1 - I.T. professional spreadsheet expert consultants

This cluster, identified in the dendrogram, corresponded to cluster two of the Kmeans analysis. It was a small cluster with only one member, case 25. However it was retained as a cluster due to its differences from other clusters, (it was the last to join in the hierarchy), and its importance in identifying a class within the taxonomy. This cluster represented well trained, highly qualified I.T. professionals acting as consultants with a particular interest in spreadsheets. User-group membership and extensive reading about spreadsheets were typical. Members of this group could be considered spreadsheet 'gurus'.

C2 - Other I.T. professional consultants

This small two-member cluster was identified in the dendrogram and corresponded to cluster eleven and case 53 from cluster four of the Kmeans analysis. Members were professional I.T. based consultants, who were not spreadsheet specialists. Qualifications were high but members had lower spreadsheet expertise than C1s or C3s and were self-trained. They did not exhibit high spreadsheet interest as they were not user-group members and read little about spreadsheets.

C3 - Non I.T. professional spreadsheet consultants

This cluster identified in the dendrogram corresponded to the remainder of cluster four in the Kmeans analysis. It had three members all acting as spreadsheet consultants but not primarily employed in an I.T. based occupation. They belonged to small organisations when they were consulting. Some were academics. These developers were well qualified and well trained, They had higher expertise than C2s, however they did not belong to a user-group and read little about spreadsheets.

D1 - User group members

This cluster of seven members, identified in the dendrogram, corresponded to clusters thirteen and ten of the Kmeans analysis. Developers were user-group members with good (cluster ten) to high (cluster thirteen) expertise. They read extensively and surprisingly were predominantly self-trained. More than half were departmental managers or executives and the majority belonged to larger organisations.

D4 - Novice developers

This medium-sized fifteen member cluster, identified in the dendrogram, corresponded to cluster three of the Kmeans analysis. Developers were novices and they were mainly employees rather than managers. Most had degree or post-graduate qualifications but had not received professional spreadsheet training, 70% were either self-trained or helped by work-mates. They tended not to read much about spreadsheets and did not belong to a user-group.

D3 - Knowledgeable developers

This cluster, identified in the dendrogram corresponded to cluster one of the Kmeans analysis. This was the largest cluster with fifty-four members involving 50% of the sample. Cluster members were all knowledgeable about spreadsheets. They were mainly employees with only a few managers represented. They tended to have high qualifications and the majority had professional memberships. A clearly identifiable subset of twelve members had no post-school qualifications though most did have professional memberships and some were managers. Cluster members were not user-group members and tended to have a low rate of reading about spreadsheets. The training they had received varied with some having attended courses or professional I.T. training and some self trained.

D2 - Lay experts

This medium-sized cluster of nine members was identified in the dendrogram and corresponded to cluster eight of the Kmeans analysis. Members did not belong to user-groups but had very high expertise. They also had high status, most being managers or executives with high academic qualifications. They tended not to belong to professional bodies. Their training in spreadsheet methods varied but they all read considerably about spreadsheets.

I1 - Non consultant I.T. professionals interested in spreadsheets

This small three-member cluster was identified in the dendrogram and corresponded to clusters twelve, six and case 45 from cluster five of the Kmeans analysis. Members were professional I.T. employees but not consultants. They were knowledgeable and read considerably about spreadsheets and were well trained.

I2 - Non consultant I.T. professionals disinterested in spreadsheets

This small two-member cluster was identified in the dendrogram and corresponded to cluster fourteen of the Kmeans analysis. These I.T. professionals were spreadsheet novices, self trained and showed little interest in spreadsheets.

D5 - Self-employed developers

The final developer cluster was identified as two separate but adjacent clusters in the dendrogram. corresponding to clusters seven (9 members) and part of cluster five (case 11) in the Kmeans analysis. All developers were self-employed, tending to work in small organisations.. Their academic qualifications were high with 45% having post-graduate degrees. Their expertise varied and they were predominantly self trained. Most read little about spreadsheets though 30% belonged to a user-group, the only developers outside cluster D1 who did.

4.3.4. Production Runs for the Application Categories of the Taxonomy

Subdivision of the remaining attributes into two classes

Group 21 runs investigated non-developer spreadsheet variables. Case 72 was found to be very different from the other cases and on review it was considered to be of doubtful validity so it was removed from the data-set for the runs of this group. The variables describing the industrial sector were also removed. (SPUBLIC, SPRIVT, SPERSN, ORGSIZE) The results of these analyses showed easily discernable clusters which were difficult to interpret. The variables describing environmental control were the biggest discriminators between clusters.

Initially the decision was made to divide the non-developer representing attributes into two classes; a priori and postieri; those known before the spreadsheet was developed and those only known after. The a priori classification would be more pertinent to the proposed use of this taxonomy to assist in developing security controls for spreadsheet development. Many of the a priori attributes dealt with environmental factors e.g. spreadsheet control policy, sufficient development time and personal use of the spreadsheet. Subsequently the decision was made to remove attributes from the data-set that dealt with developer or environmental factors and cluster them separately. The remaining attributes described the spreadsheet application. There were a few a priori attributes (e.g. purpose, corporate data inclusion) but largely postieri attributes (e.g. size, macro and graphic inclusion, links to other applications, complexity). The data-set, with case 72 included, was subdivided into developer, application and environmental variables.

Group 22 runs investigated the inclusion in the clustering of the variable SPERSN describing development for personal or recreational use. Analysis of these runs resulted in the transfer of consideration of this variable to the environmental clustering runs.

Clustering Application variables

The initial runs from group 23 clustered application variables, resulting in a few interpretable clusters and six additional clusters with just one member. The effects of weighting the size and importance variables (XSIZE, IMPORTAN) did not lead to an improved clustering. However, combining the three link variables (LINKDDE, LINKSS, LINKDB) into a composite variable LINKED reduced the number of one-member clusters.

Group 24 runs completed the analysis of the application variables. The variable RUNBY was retained. This measured how many people ran a spreadsheet. ENTKNOW, an ordinal scaled variable, measured the knowledge the data enterer had of the spreadsheet. Did a developer who designed a spreadsheet have more or less knowledge of the data entered than a user who ran the spreadsheet regularly? The sample had not collected data to answer this question so ENTKNOW was replaced by the new binary dichotomous variable ENTCLRK describing data entry by a data-entry clerk. This replacement reduced the number of small clusters. There was no longer any discrimination between spreadsheets prepared for data entry by a user who was not the developer, and one who was. Spreadsheets prepared for clerical entry were still considered separately in view of the final security oriented purposes of the taxonomy. Spreadsheets run by persons other than their developers were still represented by the variable RUNBY.

The inclusion of the variable PFORCAST resulted in a clearly identifiable cluster containing some, but unfortunately not all of the forecasting applications. This variable was discarded from further analyses but variables describing optimisation and "What if" models were retained. POPTIM and PWHATIF measured problem solving exercises which were different from the largely reporting functions of the other purpose variables PCOMMS, PREPORT, PCLASSIFY. (These had already been combined into PREST). Whilst it was recognised that forecasting differed in function from reporting, classification or communicating in that it created data, PFORCST was merged with PREST to reduce the number of clusters. Optimiser and 'What if' models have an iterative solution. Spreadsheets, when used for forecasting, or for reports, have a similar type of non-iterative solution. The 18% of forecasting spreadsheets in the sample were not permitted to exert an influence on the final analysis. The smaller 13% of goal seeking application variables PWHATIF and POPTIM were retained as separate entities as their functions were very different from those largely reporting functions represented by PREST.

Runs 24a and 24j were the final runs used to develop the application section of the of the A.D.E. taxonomy. Copies of their output can be found in Appendix D. Run 24a produced a dendrogram using join average linkage with Euclidean distance. The dendrogram was transected to give ten clusters. Tallying from the left; a) the small one or two member clusters 2 and 3 were combined as were 9 and 10, b) the largest cluster was transected at a lower distance and split into six unequal parts, and c) the first two of these secondary clusters were combined giving a total of twelve clusters for the whole dendrogram. Run 24g used the Kmeans algorithm to split the sample into nine partitioned clusters. Run 24j further subdivided the first of these clusters to give a total of fourteen clusters and was also considered when developing the taxonomy. Agreement between the Kmeans and dendrogram methods was satisfactory with ninety three out of one hundred and six cases being placed in similar clusters. The following attributes were used without weighting:

- PWHATIF - "What if" purpose
- POPTIM - optimiser purpose
- IMPORTAN - spreadsheet importance to the organisation
- THREED - three dimensional
- XSIZE - useful size (ignoring labels and blank cells)
- XGRAPH - graphics usage
- XMACRO - macro usage
- XLOGIC - Logical complexity
- RUNBY - who runs the spreadsheet
- PRIVATE - private data only

- OUTSCOPE - output distribution
- XORDFREQ - frequency of running the spreadsheet
- CDCHNG - changing corporate data
- CDNEW - source of new corporate data
- KEPT - output retention
- ENTCLRK - clerical data entry
- LINKED - links to other entities (spreadsheets, databases, DDE)

From these runs clusters were identified. Cluster profiles were determined by analysing the shaded matrix cluster output and the Kmeans cluster mean and standard deviation statistics from figures 7.5 and 7.6, and table 30 of Appendix D. The application section of the A.D.E. taxonomy was then developed:

- M1 - Models - "What if"
- M2 - Models - Optimiser
- M3 - Models - very complex
- O1 - Data entry by data-entry clerk - Unimportant spreadsheets
- O2 - Data entry by data-entry clerk - Important spreadsheets
- O3 - Data entry by user - Important spreadsheets
- S1 - 3D spreadsheets - Complex.
- S2 - 3D spreadsheets - Simple
- S3 - Non 3D spreadsheets - Complex
- S4 - Non 3D - Corporate data creators
- S5 - Non 3D - General
- S6 - Specialised Graphical spreadsheets

4.3.5. Application Cluster Profiles

M1 "What if" models

This cluster of eight members was identifiable in the dendrogram and corresponded with cluster seven of the Kmeans analysis. Members were all "what if" models. Most were run only once or a few times usually by the developer only. Their output was kept for a short time and not distributed far. They tended to use, rather than create or modify corporate data.

M2 - Optimiser models

This five member cluster was clearly identified in the dendrogram and corresponded to cluster four of the Kmeans analysis. Members were all optimiser models usually run by the developer, kept for only a short time and not distributed beyond departmental level. 40% involved corporate data. These models were simple with low link, formula and logical complexity.

M3 - Very complex models

This cluster had only one member and was clearly identified both on the dendrogram and by the Kmeans analysis, where it corresponded to cluster number two. It was retained in the taxonomy as it was one of the last clusters to join the tree, making its member very different from others in the sample. This model had high logical and formula complexity. It involved graphics, macros and links to other entities. It was run frequently by many users. This optimiser model was of moderate importance and size and used corporate data.

O1 - Data entered by data-entry clerk. Unimportant spreadsheets

This small two member cluster was identifiable on the dendrogram and corresponded to clusters six and eight in the Kmeans analysis. Members were large unimportant spreadsheets run often and regularly with data entry by a data-entry clerk.

O2 - Data entry by data-entry clerk, Important spreadsheets

This cluster of eight members was clearly identifiable on the dendrogram but not from the Kmeans analysis where it was combined with members of classes O2 and S3 to form cluster three. Increasing the number of clusters in the Kmeans analysis to 20, identified this subgroup.

These spreadsheets were of moderate to high importance, run regularly with clerical data entry. They were of moderate size and complexity, and used macros. Corporate data was involved. Their output was distributed within the department and in some cases beyond the organisation.

O3 - Data entry by user, Important spreadsheets

These thirteen spreadsheets were clearly identifiable as a cluster in the dendrogram and were combined with O2s to form the third cluster in the Kmeans analysis. The user was considered as the person who ran the spreadsheet, not necessarily the developer or even the person who entered most of the data.

Members of this cluster were run regularly involving the creation of new corporate data in 85% of cases. They were of high importance with most (75%) distributed beyond the user organisation. They tended to be large, use macros and be of moderate to high formula complexity. Most of these spreadsheets involved data entry by the user rather than the developer but a clearly defined subset of five members in the dendrogram had the developer as the user. This subset was not identifiable in the Kmeans analysis, so it was decided to retain the concept of "run by a user who was not the developer" in the profile for this class in the taxonomy.

S1 - 3D complex spreadsheets.

This small cluster of two members was clearly identifiable both in the tree dendrogram and in the Kmeans analysis where it corresponded to cluster five. Spreadsheets were large, three dimensional, logically complex and involved private not corporate data.

S2 - 3D simple spreadsheets

This small cluster of four members was identified on the dendrogram. It was combined with S4 and S5 to form the first cluster of the Kmeans analysis. These three dimensional spreadsheets were moderately large but not complex. They tended to use but not change or create corporate data and were only of moderate importance.

S3 - Non 3D, complex spreadsheets

This cluster of three members was identified on the dendrogram. It was not identified as a separate group by the Kmeans analysis and formed part of cluster three where it was combined with O2s and O3s.

Members were complex spreadsheets with links to other entities. They were of moderate importance, modified corporate data and their output was distributed at least inter-departmentally and often beyond the organisation

S4 - Non 3D Corporate data creators

This large cluster of twenty one members, was identified from the dendrogram. When the number of clusters was increased to fourteen, it was also identifiable as cluster 14 in the Kmeans output.

Members were not three dimensional. They were of moderate to high importance creating new corporate data which was distributed in 40% of cases beyond the organisation. Many had either links to other entities, graphs or macros but none was of high logical or formula complexity. Most (75%) of these spreadsheets were run by their developer.

S5 - Non 3D - General

This largest cluster had thirty members. It was identifiable on the dendrogram and formed part of cluster one in the Kmeans analysis being separated from the S4s when the number of Kmeans clusters was increased to fourteen.

Spreadsheets tended to be simple rather than complex. There was a low usage of graphics, macros and links. They used mainly private data, with a few (20%) using but not changing or creating corporate data. They were run regularly and frequently usually by the developer. Output distribution was varied but in 35% of the cases it was restricted to just the developer. Interestingly 23% of these spreadsheets were judged by their developers to be of high importance.

S6 - Specialised Graphical spreadsheet

This medium sized cluster of nine members was clearly identifiable in the dendrogram and as cluster nine in the Kmeans analysis. All members had a high involvement with intermediate to complex graphics and most had links to other entities. Many used macros. However, formula and logical complexity was average. They were run frequently and regularly and their output was distributed. Some used and even changed corporate data but none created new corporate data, and 60% involved only private data.

4.3.6. Production Runs for the Environmental Categories of the Taxonomy

Group 25 runs analysed the environmental variables. Excellent correspondence between the clusters generated was obtained with Runs 25d and 25a giving exactly the same clusters. Runs 25f and 25g were used to develop the taxonomy and their output can be found in figures 7.7 and 7.8, and table 31 of Appendix D. These runs included the variable SPERSN, which described development for personal or recreational use. This variable had previously been discarded from the developer attributes, yet it was felt to be important enough to include in the development of the A.D.E. taxonomy, hence its inclusion in this section. The two methods clustered cases identically except for case 19.

The following environmental descriptive variables were used for these analyses.

- ENUFTIME - Sufficient development time
- SDPOLDC - Organisational Spreadsheet Development Policy and its availability in documented form
- SDENFORC - Enforcement of this policy
- LIBRARY - Presence of a library of high quality spreadsheets for sharing
- SPERSN - Development for personal or recreational use.

Six clusters were clearly identified by the dendrogram and confirmed by the Kmeans analysis. These lead to the development of the environmental section of the A.D.E. taxonomy comparing regulated and unregulated environments.

- R1 - Tight control
- R2 - Loose control
- R3 - Spreadsheet library exists
- U1 - Rushed development
- U2 - Uncontrolled development
- U3 - Personal or recreational use

4.3.7. Environmental Cluster Profiles

R1 - Tight control

This cluster had only one member but was left in the taxonomy because of its importance. It was clearly identifiable in the dendrogram and corresponded to cluster four of the Kmeans analysis. This environment had a documented spreadsheet development policy enforced either by an auditor or the I.T. department. A spreadsheet sharing library existed.

R2 - Loose control

This cluster of eight members was clearly identifiable both in the dendrogram and Kmeans analyses where it corresponded to cluster two. A spreadsheet development policy existed in this environment and was possibly documented. However it was enforced either by the developer only, or at departmental level with no auditor or I.T. department involvement. There was no spreadsheet sharing library.

R3 - Spreadsheet library exists

This cluster of eight members was clearly identifiable both in the dendrogram and Kmeans analyses where it corresponded to cluster five. It was characterised by the presence of a spreadsheet sharing library. There was no formal documented spreadsheet development policy, however 25% of developers were aware of an undocumented policy which they enforced themselves.

U1 - Rushed development

This cluster of fifteen members was clearly identifiable both in the dendrogram and Kmeans analyses where it corresponded to cluster seven. The environment had no control policy and the developers were rushed and felt that they did not have sufficient time available for completing their spreadsheet development as they would have liked.

U2 - Uncontrolled development

This large cluster of sixty nine members was clearly identifiable both in the dendrogram and Kmeans analyses where it corresponded to cluster one. The environment was uncontrolled but developers did have sufficient time available.

U3 - Personal or recreational use

This cluster of five members was clearly identifiable both in the dendrogram and Kmeans analyses where it corresponded to cluster three. This uncontrolled environment supported spreadsheets developed for personal or recreational use.

4.4. The A.D.E. Taxonomy

The A.D.E. taxonomy of spreadsheet applications development was arranged with respect to the cluster profiles identified in the cluster analyses described above.

4.4.1. The Developed Taxonomy

The taxonomy was arranged in three sections:

a) A the Application. This section categorised the spreadsheet application i.e. the product of a development project. It was further subdivided into spreadsheet applications that could be primarily considered as models and those whose main purpose was reporting.

b) D the Developer. This section categorised the skills and background of the developer of the spreadsheet application. Developers were further subdivided into those who acted as consultants (for this particular project), other I.T. professionals and other developers.

c) E the development Environment. This section categorised the development environment where the spreadsheet application was developed. This section was divided into two broad categories of environments with some form of external control and those without.

The A.D.E. Taxonomy of Spreadsheet Applications Development

A The Application

Models

M1	Models - "what if"
M2	Models - optimiser
M3	Models - very complex

Reports and other applications with non-developer data entry

O1	Data entry by data-entry clerk - unimportant spreadsheet
O2	Data entry by data-entry clerk - important spreadsheets
O3	Data entry by User - important spreadsheets.

Reports and other applications with data entry by the developer

S1	Three Dimensional - complex
S2	Three dimensional - simple
S3	Two dimensional - complex
S4	Two dimensional - create corporate data
S5	Two dimensional - general
S6	Specialised graphical spreadsheets

D The Developer

Consultants

C1	I.T. professional consultants - spreadsheet specialists
C2	I.T. professional consultants - not spreadsheet specialists.
C3	Spreadsheet consultants - not I.T. professionals.

Other I.T. Professionals

I1	Non consultant I.T. professionals - disinterested in spreadsheets
I2	Non consultant I.T. professionals - interested in spreadsheets

Other Developers

D1	User-group members
D2	Lay experts
D3	Lay knowledgeable developers
D4	Lay novice developers
D5	Self-employed developers

E The Environment

Controlled

R1	Tight control
R2	Loose control
R3	Spreadsheet library exists

Uncontrolled

U1	Rushed development
U2	Uncontrolled but not rushed development
U3	Personal or recreational use

4.4.2. Description of the Sample Using the Taxonomy

The distribution of the sample amongst the Application categories is shown below in Figure 4.24. The applications were predominantly developer run reports, The sample also contained a few models and reports prepared for others to run. Two dimensional general reports were the most common types of spreadsheet however 20% of the applications created new corporate data.

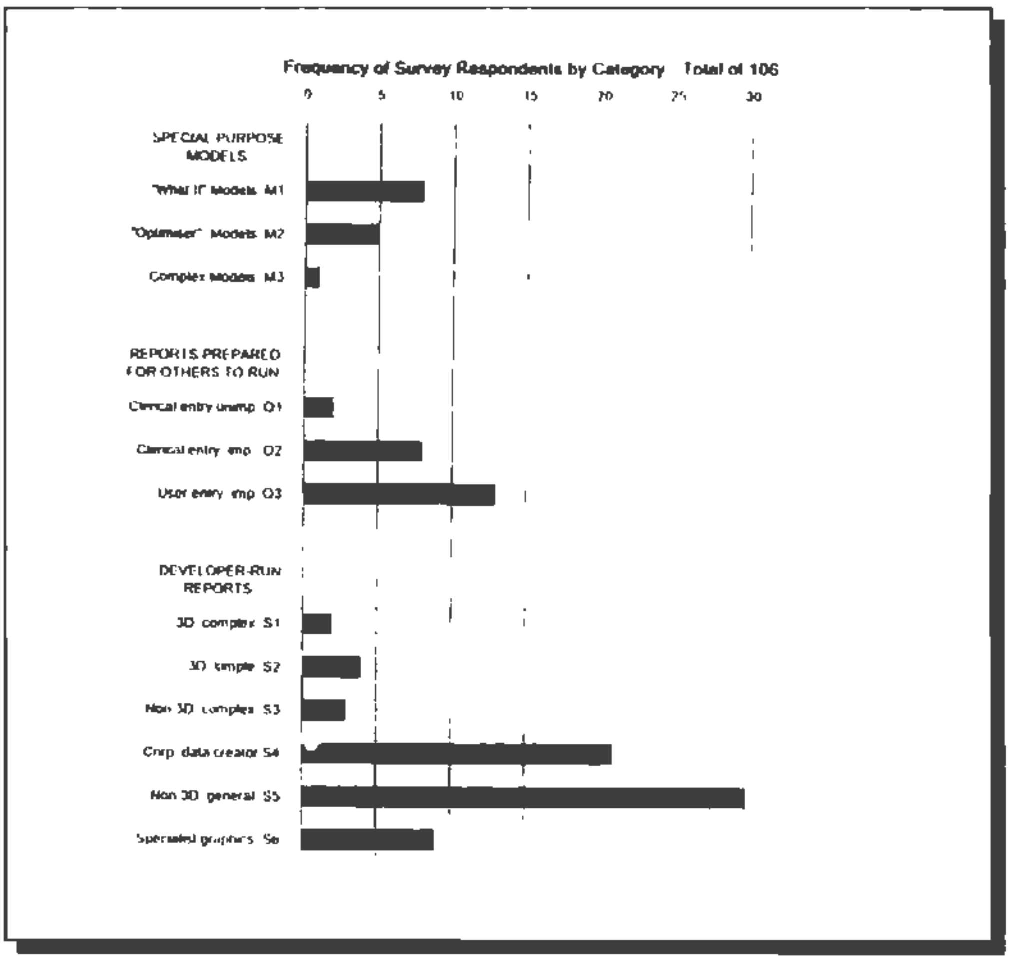


Figure 4.24 Spreadsheet Survey. Frequency distribution of cases amongst the A.D.E. Taxonomy Application categories

The distribution of the sample amongst the developer categories of the taxonomy is shown below in Figure 4.25. The sample was not particularly heterogeneous with most spreadsheets developed by lay knowledgeable developers with only a few consultants and I.T. professionals represented.

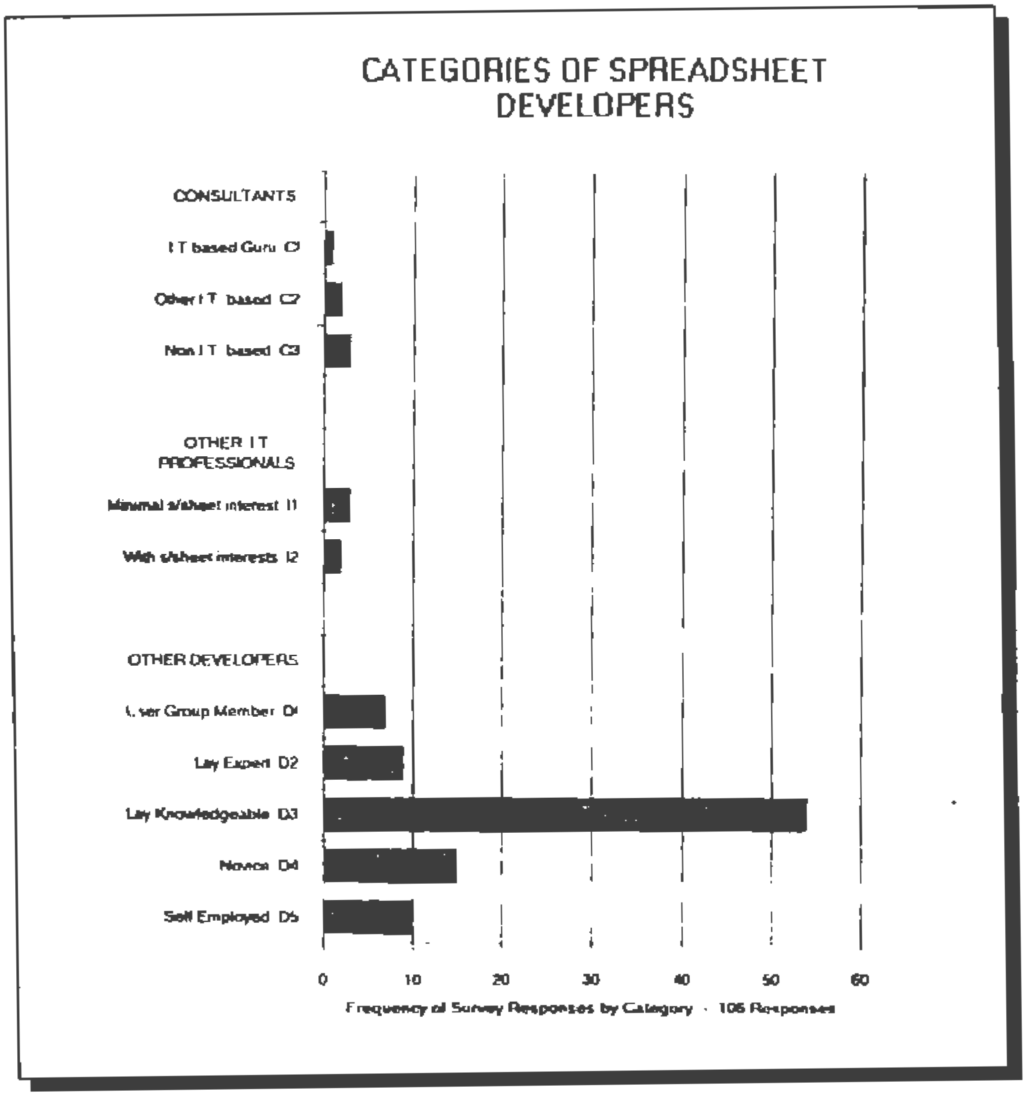
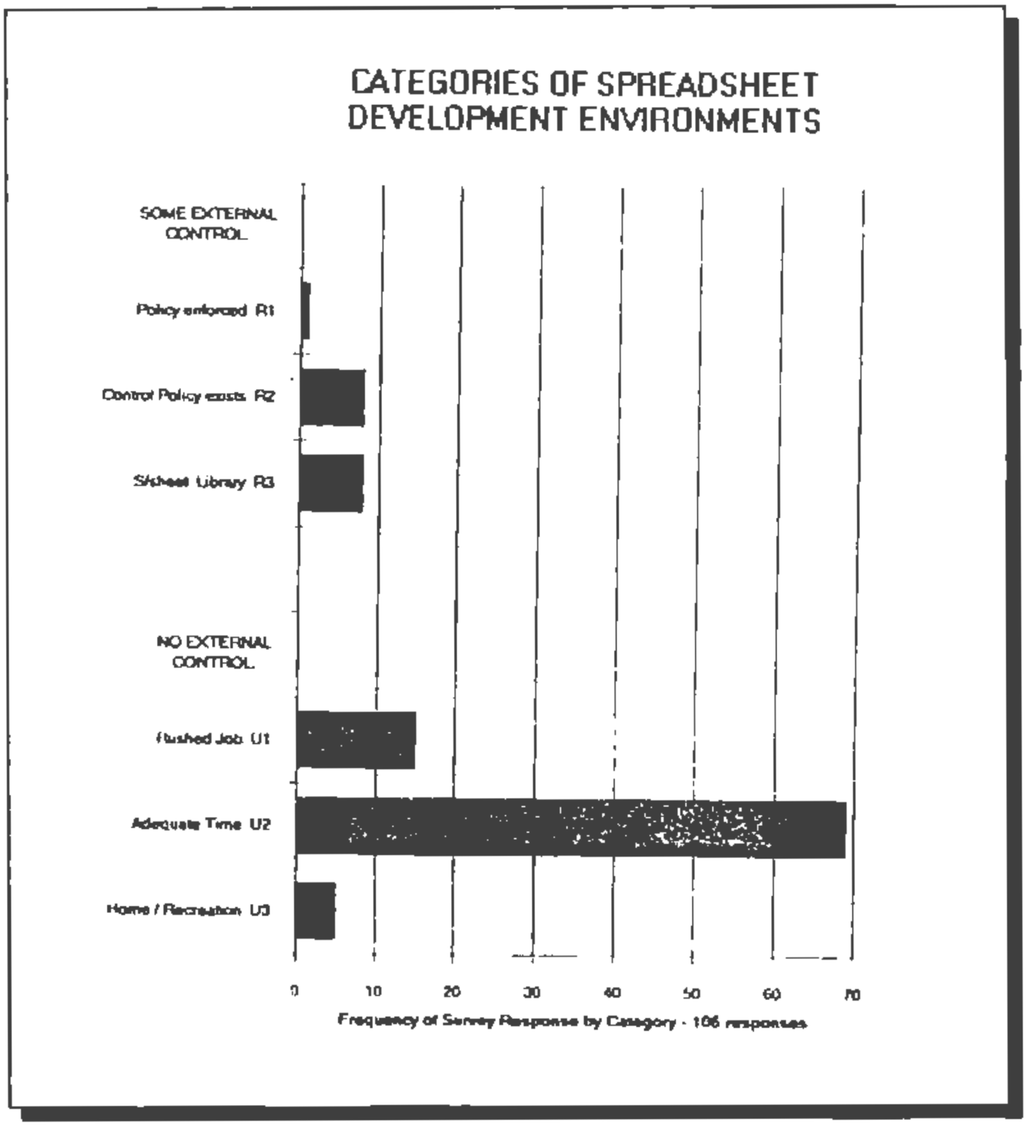


Figure 4.25 Spreadsheet Survey: Frequency distribution of cases amongst the A.D.E. Taxonomy developer categories

The distribution of the sample amongst environmental categories is shown below in Figure 4.26. Again the sample was not particularly heterogeneous with the majority of spreadsheets being developed in uncontrolled environments. 14% were developed as a rushed job. An enforced spreadsheet policy was only apparent in 1% of the sample.



<u>Figure 4.26</u> Spreadsheet Survey: Frequency distribution of cases amongst the A.D.E. Taxonomy environmental categories.

Graphical comparison of sample cases using the taxonomy

The A.D.E. taxonomy categories were subjectively ranked as shown below in Table 5. Applications were ranked from lowest to highest on importance and complexity, within type of model, developers on expertise, and the environment on control.

Table 5. A.D.E. Taxonomy categories ranked.

A	Complexity	Rank	D	Expertise	Rank	E	Control	Rank
S5	2D general	1	D4	novice	1	U3	personal or recreational	1
S2	3D simple	2	I1	IT prof. disinterested	2	U1	rushed job	2
S4	Corporate data created	3	D5	self-employed	3	U2	uncon-trolled	3
S6	graphical	4	D3	lay knowl-edgeable	4	R3	library exists	4
S1	3D complex	5	I2	IT prof inter-ested	5	R2	loose control	5
S3	2D complex	6	D1	user-group member	6	R1	tight control	6
O1	data entry by clerk unimp.	7	D2	lay expert	7			
O2	data entry by clerk imp.	8	C2	IT consultant Not spr/shts	8			
O3	data entry by user	9	C3	Consultant not IT prof	9			
M1	what if model	10	C1	Consultant IT expert	10			
M2	optimiser	11						
M3	complex model	12						

Graphical methods using SYSTAT's SYGRAPH module were used to further analyse the sample. The multivariate plot shown in figure 4.27 below, shows the combinations of CLENV3 (environmental category), CLDEV3 (developer category) and CLSSHT3 (application category). All combinations of codes present in the sample are shown.

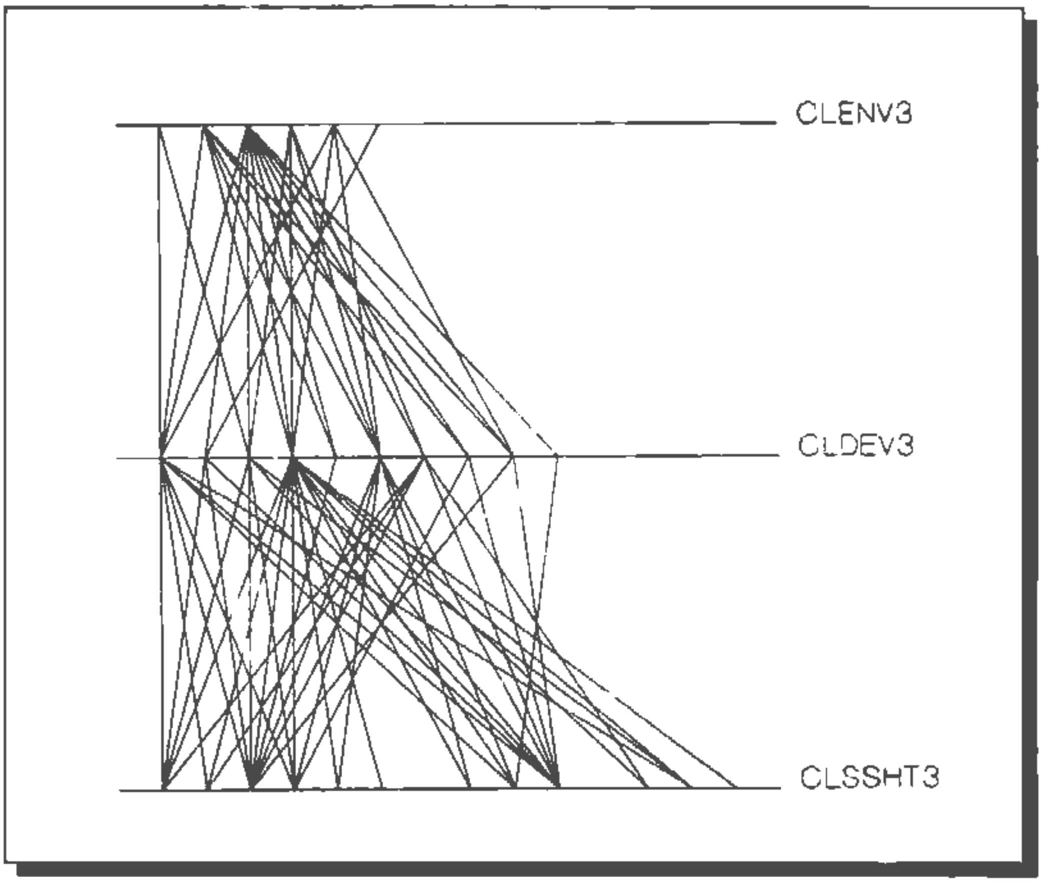


Figure 4.27: Multivariate plot of the spreadsheet sample. (CLENV3 - environmental code, CLDEV3 - developer code, CLSSHT3- application code)

Figure 4.27 does not show how many cases had a particular combination of codes but does show each pathway between the three variables where there was at least one occurrence. The graph shows a broad coverage of possible pathways for a sample of only 107 cases.

Figure 4.28 graphically seeks for a relationship between the application, developer and environmental variables. The environmental control rank (Y axis) was plotted against the ranked developer expertise (X axis). Each case was represented on this plot by a character representing the application category; M (model), O (spreadsheet prepared for others to run) or S (prepared for self to run).

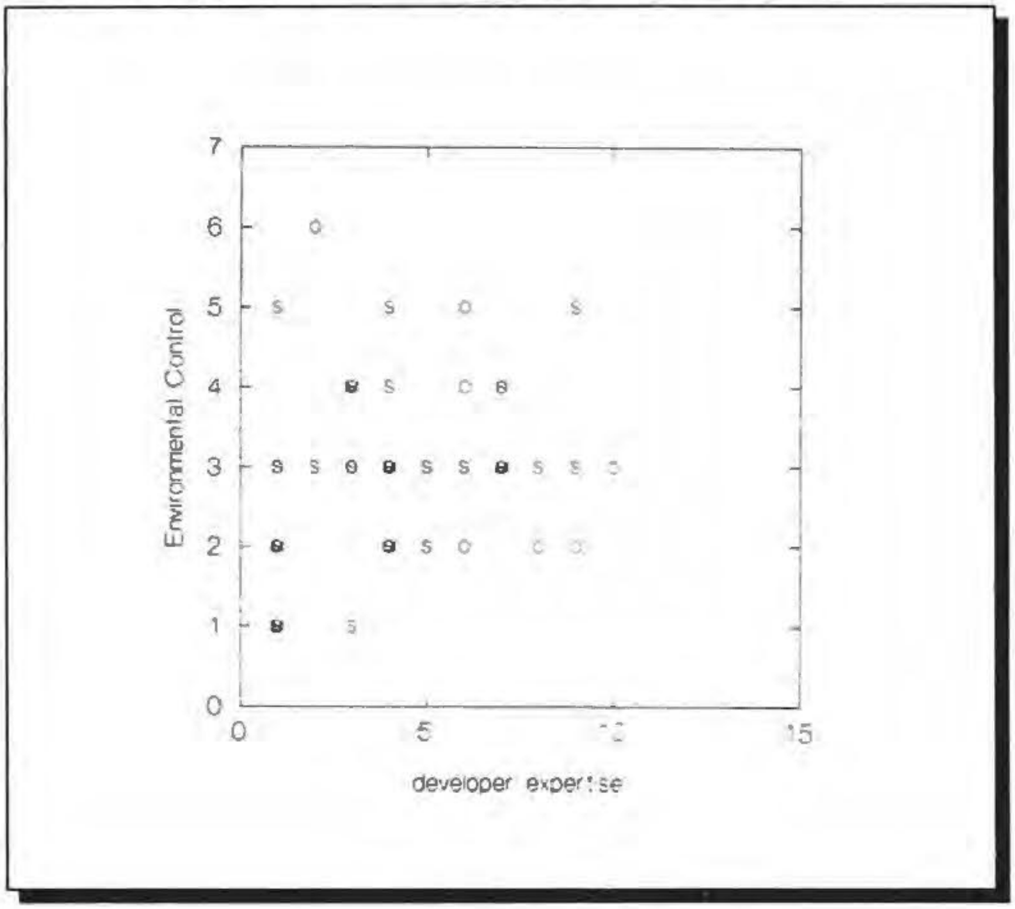


Figure 4.28 Surveyed spreadsheets. Spreadsheet Development scatter plot. M - model, O - prepared for others to run, S - self run

Figure 4.28 shows that models were developed by people of varying expertise but tended not to be developed in controlled environments or by consultants. However spreadsheets prepared for others to run tended to be developed by the more expert developers including consultants. Those few less expert developers, who prepared spreadsheets for others to run, worked in environments with at least some measure of control.

Figure 4.29 shows a scatter plot of developer categories (Y axis) against type of spreadsheet developed (X axis). The size of the point on this plot corresponds to the rank of the environmental control code.

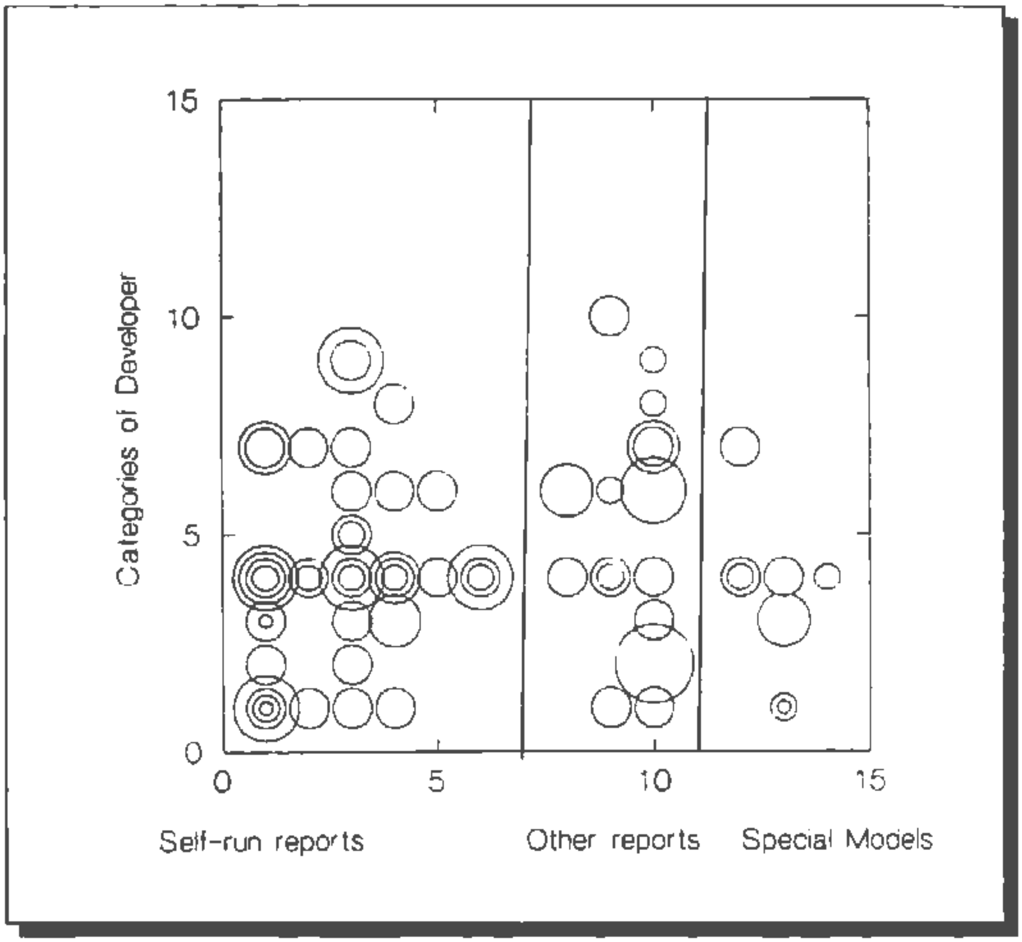
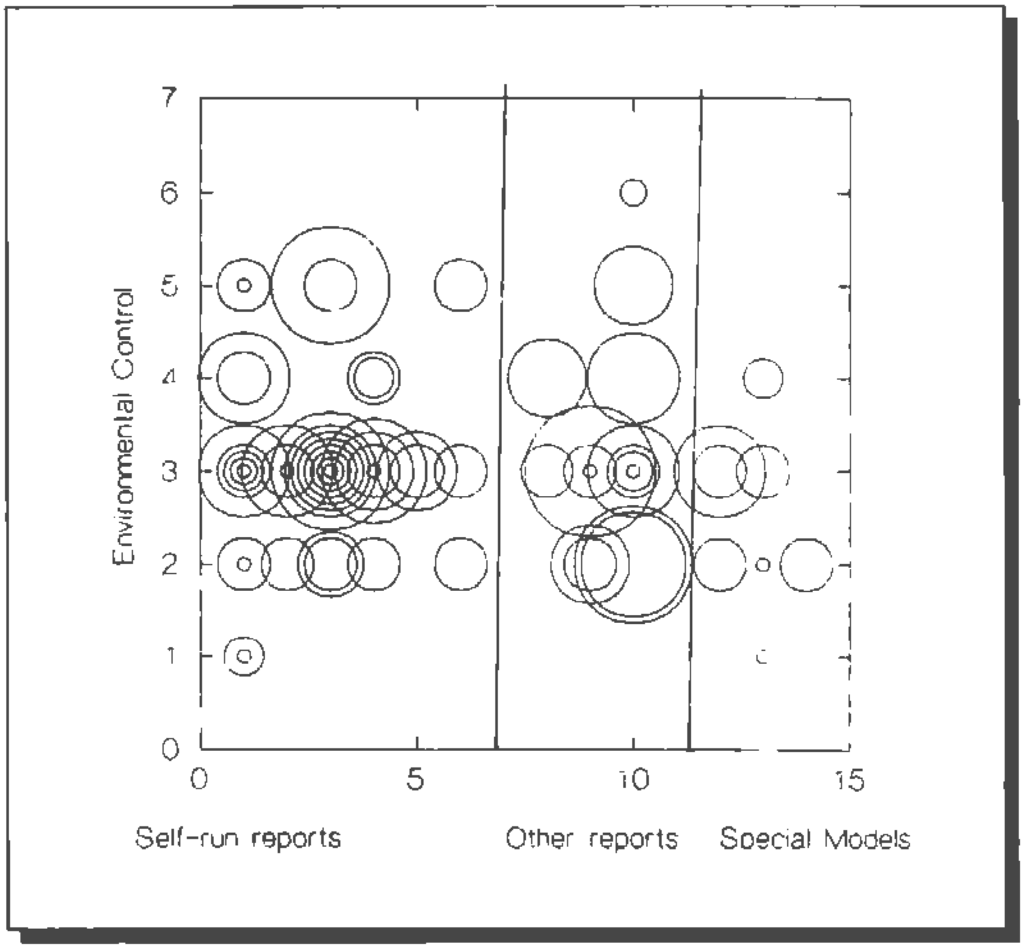


Figure 4.29: Spreadsheet sample. Plot showing types of spreadsheet developed by different categories of developer.

Interestingly, models tended to be developed by lay knowledgeable developers working in unregulated environments rather than by consultants. As might have been expected, half the reports prepared for others to run were developed by developers with higher expertise Self run reports were developed by all categories of developers. The degree of environmental control varied throughout the sample and no particular trend could be spotted by eye from this plot, except that it was low for the development of special models.

Figure 4.30 shows a scatter plot comparing environmental control (Y axis) to type of spreadsheet developed (X axis). In this plot, the developer expertise is represented by the size of the point.



<u>Figure 4.30</u>: Spreadsheet sample. Scatter plot showing types of spreadsheets developed and degree of environmental control. The size of the point represents developer expertise.

Again this plot demonstrated that developers, developing reports for others to run tended to have higher expertise than those developing models. There could be some relationship between environmental control and expertise. Spreadsheets developed either at home or as a rushed job tended to be developed by developers with lower expertise whilst developers working in environments with at least some measure of loose control tended to have a slightly higher level of expertise. However 8 out of 31 cases (25%) were exceptions to this trend.

Figure 4.28 plotted developer expertise against environmental control. Even when the one case representing a strictly controlled environment was considered an anomalous outlier and removed, the trend for expertise to increase linearly with environmental control was barely discernible. Also as the ordinal scales used to measure the variables were contrived, it can not be said that there is a linear relationship between developer expertise and level of environmental regulation, only that this relationship is perhaps worthy of future investigation with additional data.

Relationship between environmental regulation and the building of models

Figures 4.28 and 4.29 suggested that models were more likely to be built in unregulated environments. A contingency table was drawn up to test this.

Table 6

Spreadsheet Sample. Frequencies of model development in regulated and unregulated environments.

	Regulated Environment	Unregulated Environment	TOTAL
model	1	13	14
non model	16	76	92
TOTAL	17	89	106

A Chi square test could not be used on Table 6 as one of the cells contained a frequency less than 5; i.e. only one model had been developed in a regulated environment. However 7% of all models compared to 17% of all non models were developed in regulated environments. In this sample, spreadsheets developed in regulated environments were even less likely to be models than spreadsheets developed in unregulated environments.

Relationship between developer expertise and developing spreadsheets for others to run

Figures 4.28, 4.29 and 4.30 suggested that spreadsheets developed for others to run were more usually developed by developers with higher expertise.

Table 7:

Spreadsheet Sample. Frequencies of developer expertise and spreadsheets developed for running by others.

	EXPERT = 1	EXPERT = 2	EXPERT = 3	TOTAL
run by self	19	55	9	83
run by others	2	16	5	23
TOTAL	21	71	14	106

A contingency Table 7 was drawn up to statistically test the hypothesis:

H_0: Developers of different expertise do not differ on their rates of developing spreadsheets for themselves or for others to run.

As the smallest frequency was 2 and two degrees of freedom were involved, a Chi square analysis could be used.

χ^2 calculated statistic was 3.480 (χ^2 critical = 3.219, α = .2, 2 d.f.). At a confidence level of .2 H_0 can be rejected.

There is an association between the expertise of the spreadsheet developer and the rate of developing spreadsheets for others to use. We can say with only 80% certainty that spreadsheets designed for others to use, are more likely to be developed by more expert developers. If a higher confidence level is required, then H_0 would have to be accepted, and no such significant association would have been demonstrated.

4.5. A.D.E. Taxonomy Diagnostic Key.

A diagnostic key was developed separately for each section of the taxonomy. The keys took the form of hierarchical decision trees. An effort was made to design these trees with the minimum number of questions required to discriminate between categories. In so doing, a logical progression of categories across the foot of the key was sacrificed. As it was impossible to have both the minimum number of questions and also the final categories arranged in a logical manner, the choice was made to retain the minimum number of questions to simplify response.

The three keys were packaged together with a cover page giving a short description on their use. A copy of this key can be found in Appendix A with the questionnaire for the validation survey.

The three decision trees shown in figures 4.31, 4.32 and 4.33 demonstrate this key for the Application, Developer and Environmental categories of the A.D.E. taxonomy of spreadsheet applications development.

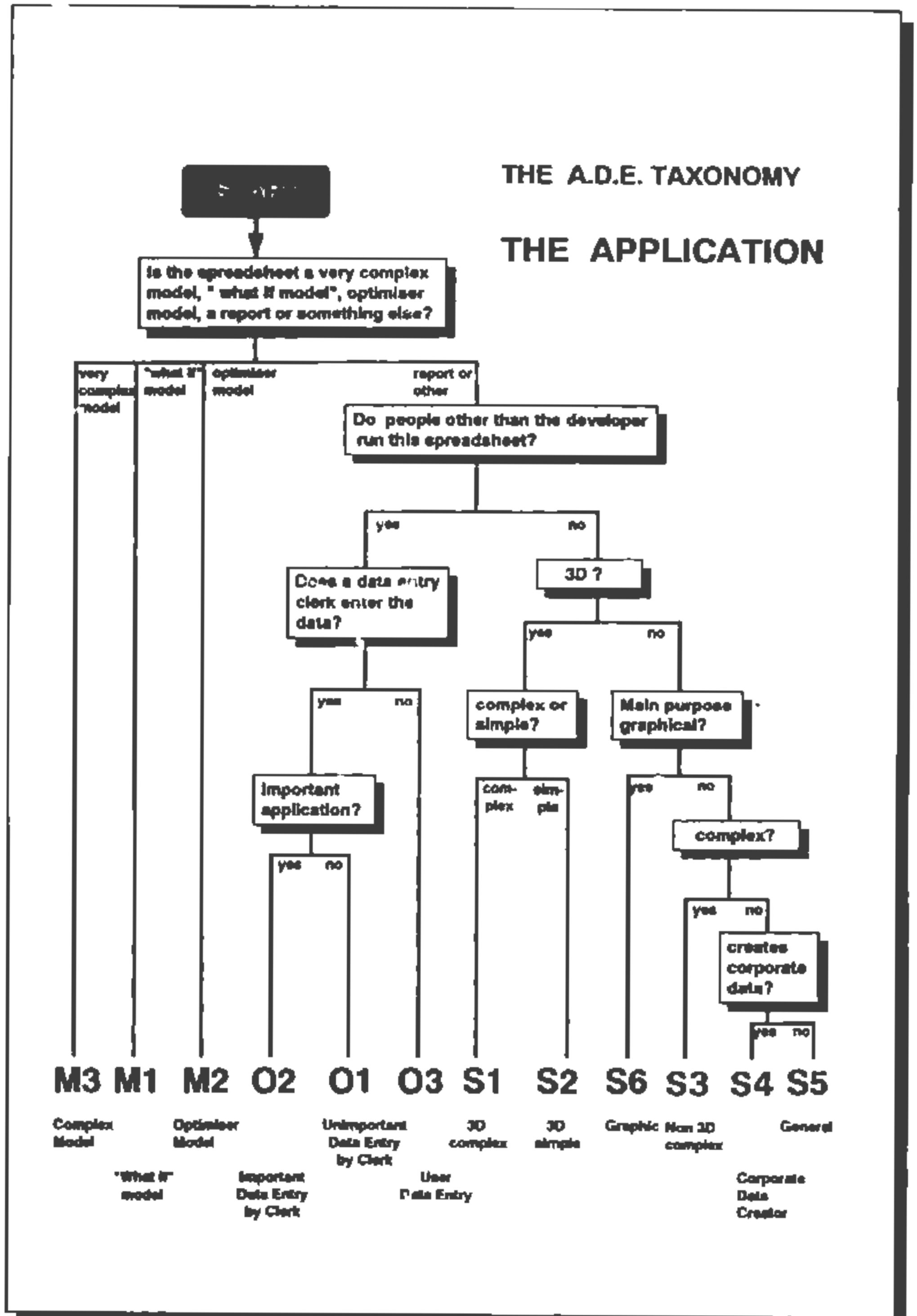


Figure 4.31 The A.D.E. taxonomy of Spreadsheet Applications Development: Diagnostic Key for the Application Codes.

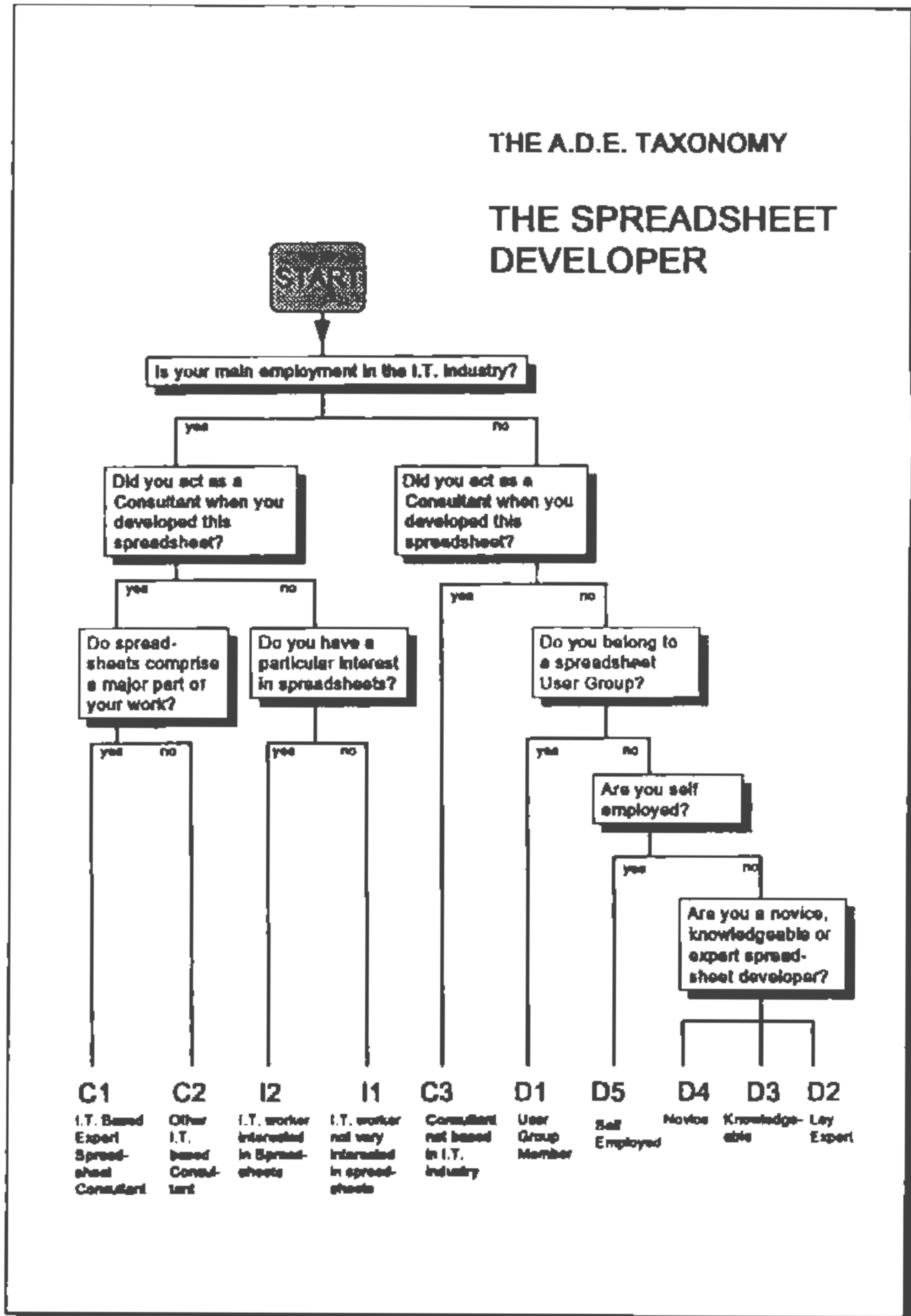


Figure 4.32 The A.D.E. Taxonomy of Spreadsheet Application development: Diagnostic key for the Developer Codes.

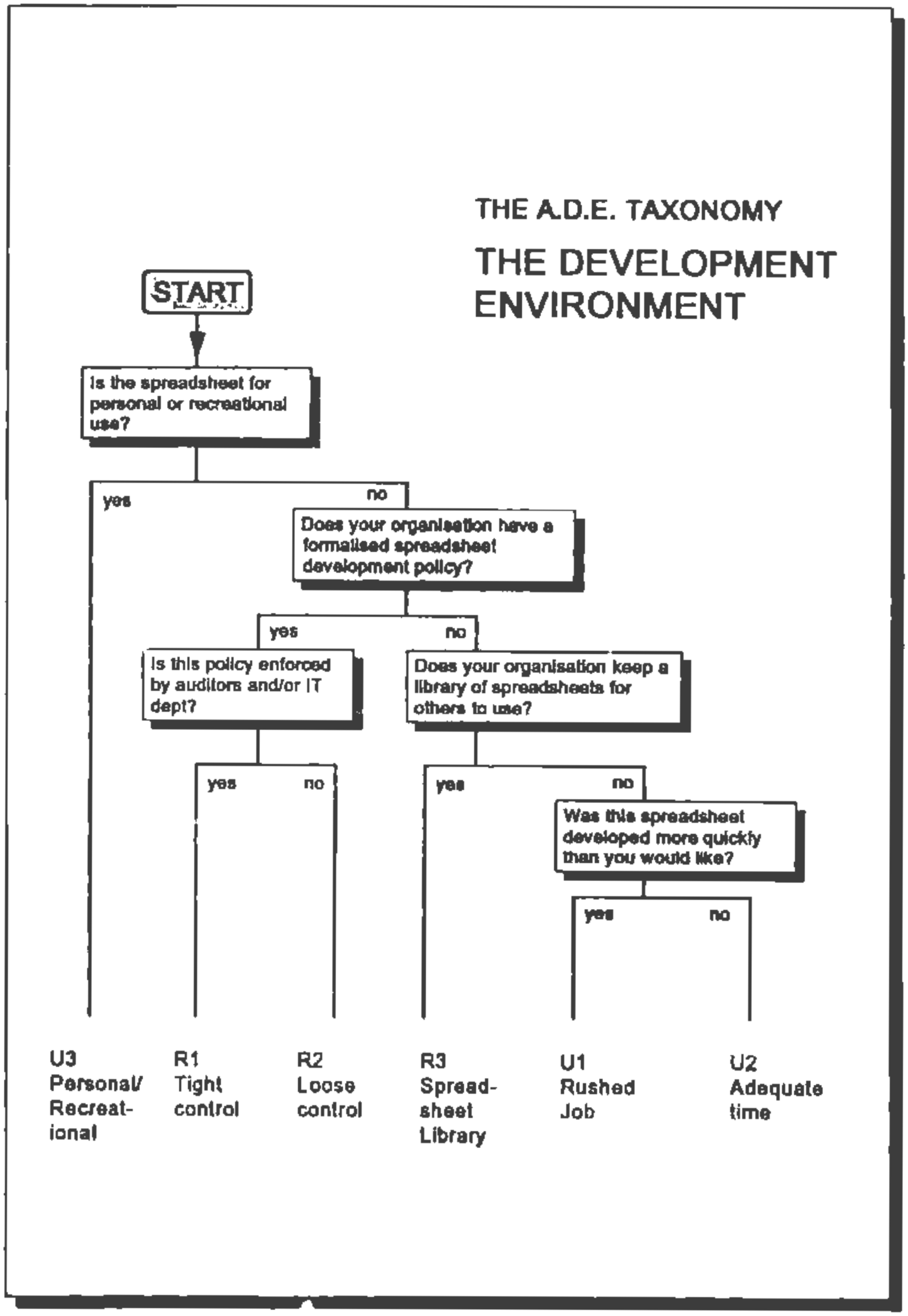


Figure 4.33 The A.D.E. Taxonomy of Spreadsheet Application Development: Diagnostic Key for the Environment Codes.

4.6. Taxonomy Validation

The validation of the taxonomy and its diagnostic key is described in detail in chapter 5.

4.7. Gender Differences in Spreadsheet Development

I had noticed in my lecturing career, that some female students appeared to have more difficulty learning how to use a spreadsheet package, than they experienced when learning a word processor or data base management system. I had not been able to determine why this was so and wondered if it was due to a lack of confidence in their capabilities.

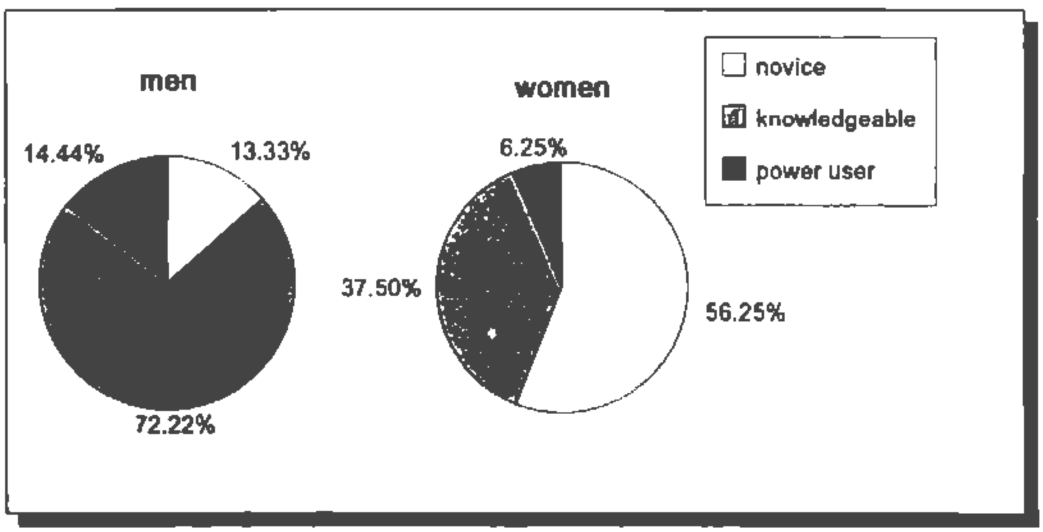


Figure 4.34: Spreadsheet survey. Comparison of developer gender and expertise.

Figure 4.34 compares the self ranking of spreadsheet development expertise by male and female survey respondents. The sample contained 16 women and 90 men. 56% of women and only 13% of men considered themselves to be novice developers. A contingency Table 8 was drawn up, showing the frequencies of gender and developer expertise. 'Knowledgeable' and 'power users' were combined in this table, because there was only one female 'power user', and one respondent had reported she felt that Schneiderman's (1980) term 'power user' may have discouraged women.

<u>Table 8</u> **Spreadsheet Survey. Gender and Developer Expertise.**

	novice developer	knowledgeable or power user	total
women	9	7	16
men	12	78	90
total	21	85	106

The frequencies in table 8 were used to test the hypothesis:

> H_0: There is no difference in the spreadsheet development expertise of women and men.

χ^2 calculated was 15.766 (χ^2 critical = 3.84146, α = .05, 1 d.f.), so H_0 was rejected. There is an association between gender and spreadsheet development expertise. Men report that they have higher expertise than that reported by women.

In an effort to determine why men in this sample reported they had a higher spreadsheet development expertise than that reported by women, a series of chi square analyses was conducted The detailed contingency tables and results can be found in Appendix E.

Gender was compared with employment status, organisation size, qualification and training. No association was found.

The possibility that men were using spreadsheets for more important tasks was canvassed as this may have had an influence on developers' perceptions of their expertise. Gender was compared to spreadsheet importance, range of spreadsheet distribution, rate of creating and changing corporate data. Again no association was found.

Finally gender was compared with variables which gave an indication of the technical sophistication of a spreadsheet. There was no association between gender and link complexity, use of graphics or use of macros. Associations were found between gender and spreadsheet size, logical complexity and formula complexity.

Men tended to design larger, more complex spreadsheets. However there is no indication that size or logical complexity is a measure of developer expertise. Smaller, simpler spreadsheets may result in less errors and be preferable from a control perspective.

Whilst these results are interesting, we can not infer anything about the spreadsheet expertise of women spreadsheet developers in the general population, due to the non-random nature of the sample, However, these results lead to some hypothesis which could be tested in a follow up study. This matter is discussed further in chapter 6.

4.8. Summary of this Chapter

This chapter described the results of this study. Initially statistics of the sample were reported. A series of cluster analysis runs was detailed leading to the evolution of the A.D.E. taxonomy of spreadsheet application development and its diagnostic key. The sample was described in terms of this taxonomy and multivariate graphs were drawn to identify associations between different categories within the taxonomy for cases in the sample. Finally some associations between gender and expertise were considered.

CHAPTER 5: STUDY VALIDATION

5.1. Chapter Overview

This chapter reports on the validation of this study. It begins with a review of some validation criteria suggested in the literature and shows how these relate to the study research goals established in chapter 1.

The validation of the data collection instrument used in the original spreadsheet survey is then considered. A validation survey and several validation exercises are described, leading to the validation of the taxonomy and its diagnostic key. The A.D.E. taxonomy is compared and contrasted with other partial taxonomies of the spreadsheet development process, reported in the literature. Finally, the usefulness of the A.D.E. taxonomy in an analysis of the pre-designing tendency of spreadsheet developers, is assessed.

5.2. Validation Criteria

Chapter 2 established that a taxonomy was a model of the system it was attempting to categorise. It is important to determine if a model agrees with the real system. i.e. the model requires validation. Two kinds of validation are possible, verification and falsification. Verification seeks to design a sequence of experiments to show sufficient agreement between the model and the real system. In contrast, falsification looks for a single example to disprove the model.

The A.D.E. taxonomy validation was conducted from the verification rather than falsification perspective. Verification was considered in two different ways. The taxonomy was validated with respect to the primary and secondary research goals set out in chapter 1. Validation of the taxonomy was also considered in terms of criteria established from reports in the literature e.g. content, construct, criterion referenced and 'face' validity. These two different validity methods were not in conflict. They simply represented two different 'validity' models of the same reality.

5.2.1. Validity with Respect to the Research Goals

The taxonomy was validated with respect to the goals of this study. The major research goals applicable to the validation of this taxonomy have been repeated below for convenience.

Primary research goals

The primary research goals were:

a) Improve the planning and management of spreadsheet applications development.

b) Develop a special purpose classification - Taxonomy of spreadsheet application development for use in controlling the development of spreadsheets.

Secondary research goals

The secondary research goals were considered in three groups, the first was concerned with the exploratory data analysis:

a) Identify a suitable sampling frame for use in the primary data collection.

b) Gain a better understanding of the underlying structure within the data-set through exploratory data analysis and data reduction.

c) Generate hypotheses for future study.

The second group was concerned with an 'ideal' solution to the Cluster Analysis procedures

a) Achieve a clustering solution from which a suitable taxonomy can be developed.

b) Achieve a clustering solution showing well structured clusters.

c) Achieve a clustering solution showing intuitive clusters.

The third group of Secondary Research goals was concerned with validating the taxonomy:

a) Demonstrate taxonomic stability.

b) Demonstrate taxonomic robustness.

c) Demonstrate taxonomic replicability.

d) Demonstrate agreement with other taxonomies reported in the literature.

e) Demonstrate agreement with the researcher's a priori expectations.

f) Demonstrate the usefulness of the taxonomy.

g) Validation of the Taxonomy Diagnostic Key.

5.2.2. Content, Construct, Criterion Referenced and 'Face' Validity

Many authors suggest criteria for the validation of taxonomies and/or data collection instruments. The concepts of content, construct, criterion referenced and 'face' validities were considered when planning the validation of both the A.D.E. taxonomy, and the data collection instruments.

Content Validity

Content validity of an instrument has been defined as:

> How well the material included in the instrument represents all possible material that could have been included. (Long, Conway and Chwalek, 1985, p. 90)

Content validity in this study was concerned with how well the taxonomy or instrument covered all the available material that might have been included.

Construct Validity

Construct validity has been defined as:

> How well the instrument measures the theoretical concept called a construct or trait that is assumed to explain the behaviour represented by this instrument (Long, Conway and Chwalek, 1985, P. 910)

Construct validity in this study would be determined by how well the taxonomy or instrument agreed with published theories.

These were demonstrated by reference to the published partial taxonomies described in the review of the literature in chapter 2. Content and construct validity were also established as the literature guided the choice of the original attributes used to develop the taxonomy.

Criterion referenced validity

The criterion referenced validity of an instrument has been defined as:

> How well this instrument correlates with some criterion external to it.
> (Long, Conway and Chwalek, 1985, p. 90)

Criterion referenced validity was established in this study considering both internal and external criteria. External criterion validity was established comparing this taxonomy to other taxonomies and internal criterion referenced validity ensured that the taxonomy modelled the underlying structure of the data-set, using tests from within the cluster analysis process.

Face validity

Mehrens and Lehmann (1978, p. 114) defined 'face' validity, as "valid on the face of it", i.e. it appears right. The A.D.E. taxonomy was developed making use of those clustering solutions that appeared 'right'. The use of the taxonomist's subjective opinion and intuition confirmed 'face' validity. The respondents' opinions on 'face' validity were also considered in the validation survey, when they were asked to comment on any difficulties they had experienced in completing a categorisation of a spreadsheet development project.

5.2.3. Other Validity Models

Troy and Moawad (1982, p. 29) considered three aspects of the adequacy of a software reliability model, which have been modified to address the validation of the A.D.E. taxonomy:

a) Utility - the relationship between the A.D.E. taxonomy and its user. Is it useful?

b) Applicability - the relationship between the A.D.E. taxonomy and reality. Does it depict reality well?

c) Validity - the internal accuracy of the A.D.E. taxonomy

Troy and Moawad (1982) considered three levels of validity, 'Operational', 'Structural' and 'Conceptual'. All three are pertinent to the validation of this study. The 'Operational' level related to the users' view of the taxonomy and was validated by their use of the diagnostic key. The 'Structural' level was concerned with the building of the model and was validated by the validation of the data collection instrument and the extensive procedures undertaken during the data-entry and pre-processing phases. The 'Conceptual' level was concerned with the theoretical basis for the taxonomy. 'Conceptual' validity was demonstrated as the taxonomy was evolved through well known Cluster Analysis methodologies, extensively documented in the literature.

Howard and Murray (1987, p. 181) summarised methodologies reported in the literature for use in human factors computer interface research and provided a taxonomy of evaluation methods:

a) Expert based - expert walk through of the system
b) Theory based - relate back to the theory
c) Subject based - requires a task, system, user and metric, user to validate the user affective, cognitive, behavioural and physiological levels
d) User based - personal evaluation
e) Market-based - final evaluation in the market-place

Expert based evaluation would have required the expert to have extensive knowledge of the user, the spreadsheet and the project environment. As this was impractical, expert based evaluation was not used. The taxonomy was validated with respect to theory as its development was based on published theories of end-user computing and cluster analysis. It would have been extremely difficult to evaluate the taxonomy's acceptance in the market-place as this would only be determined several years after publication. Accordingly subject and user-based methodologies were deemed more appropriate to evaluate the A.D.E. taxonomy.

The validation also considered the subject based criteria of 'communicability', 'reliability', 'usefulness' and 'suggestiveness' described by Bloom et al (1956) and Biggs and Collis (1982).

'Communicability' was demonstrated when different raters agreed on the classification of a spreadsheet project using the taxonomy. This would have allowed them to communicate with each other with the assurance that they were discussing the same type of spreadsheet.

The validation of the taxonomy with respect to its 'usefulness' is discussed later in this chapter, when the taxonomy is used to analyse whether developers pre-design their templates on paper. Future studies to demonstrate usefulness are outlined in the final chapter.

A taxonomy valid under the 'suggestiveness' criteria should stimulate thought and discussion. The validation survey prompted interested response from some participant validating the taxonomy under this criterion.

5.3. Questionnaire Validity and Reliability

The validity of the questionnaire determined whether it measured what it purported to measure. Content, construct and criterion referenced validity were considered:

Questionnaire Content validity

The suggestions of expert participants in the pilot test regarding questionnaire content and presentation, established the content validity of the data collection instrument. Many different partial taxonomies relevant to the spreadsheet development were reviewed in chapter 2. Attributes described in these articles were included in the questionnaire, validating its content. The validation of the A.D.E. taxonomy diagnostic key through the validation survey, described in this chapter, also attested to the content validity of the questionnaire on which its development was based.

Content validity of the third section of the questionnaire, dealing with spreadsheet design and control issues was established with reference to articles in the literature, where spreadsheet controls were discussed. These articles included Anderson and Bernard (1988), Ashworth (1987), Beitman (1986), Bromley (1985), Bryan (1986), Chan (1987), Davies and Ikin (1987), Ditlea (1987), Foye (1989), Gaston (1986), Hayen and Peters (1989), Kee and Mason (1988), Levine and Siegal (1987), Pearson (1988), Ronen, Palley and Lucas (1989), Schultz and Hoglund (1986), Spencer (1986), Stewart and Flanagan (1987), Weber (1986) and Williams (1989).

Questionnaire criterion referenced validity

Criterion referenced validity of the data collection instrument would have been demonstrated if this instrument could have been compared with a another instrument of known validity, developed for the same purpose. This was infeasible as no other instrument, designed for the same use, was available.

Questionnaire construct validity

Long, Conway and Chwalek consider the measurement of construct validity difficult (1985, p. 91), however an attempt was made to ensure construct validity of the data collection instrument. The spreadsheet SIZE.SSF calculated an effective size of a spreadsheet from the numbers of rows, columns and dimensions and the number of unfilled cells. This was compared to the reported storage size in bytes of a spreadsheet taken from the questionnaire. The ratio of the reported to the calculated size was examined for different brands of spreadsheet software, thus ensuring that the two different sets of questions included in the questionnaire both modelled the same trait - 'size'.

Questionnaire reliability

The reliability of the questionnaire, i.e. its consistency of measurement was also considered. Reliability comprises consistency between different measurements. The stability of the instrument was tested by the comparison of two measurements of the same case at different times. This was established when the original four 'one on one' participants were asked to repeat the questionnaire for the pilot test. Their two answers were compared and found to be similar.

5.4. Validation of the A.D.E. Taxonomy Diagnostic Key

The diagnostic key of the A.D.E. taxonomy was validated by several different exercises and comparisons based on data collected through a validation survey.

5.4.1. Validation Survey

A survey was conducted of developers categorising their spreadsheet projects using the diagnostic key to the A.D.E. taxonomy. This provided data for some of the validation exercises described in this chapter.

A taxonomy validation instrument was prepared, consisting of a simple cover-page including instructions and the three decision trees required to categorise a spreadsheet development project within the A.D.E. taxonomy. A copy of this instrument can be found in Appendix A.

This instrument was submitted to 25 spreadsheet developers chosen using random number tables and the frame constructed for the Preston stratum. They were asked to categorise a spreadsheet they had recently developed, and to comment if they had any difficulties using the diagnostic key. They were instructed to select a different spreadsheet for this exercise from the one they had analysed for the original survey.

Respondents were requested, where possible, to get an additional rater familiar with the spreadsheet and the situation in which it was developed, also to complete the validation instrument. The two categorisations were compared and analysed for inter-rater discrepancies.

Responses were received from 24 of the original sample of 25. In addition, 6 of the respondents also returned a response from an alternate rater. Half (12) of the original respondents repeated the validation survey instrument, six weeks after their first attempt using the same spreadsheet development project. These results were then compared to those obtained the first time they categorised their spreadsheet development. Six weeks allowed sufficient time for the developer to have forgotten their original decisions when using the diagnostic key, but was not long enough for the spreadsheet development project to have changed significantly. Balance was maintained between bias introduced by the respondent being familiar with the material having recently completed the validation survey and bias introduced by changes in the project being measured.

5.4.2. Validation Survey Results

The validation survey validated the diagnostic key as to ease of use. No difficulties in completing the instrument were reported by respondents. No respondent reported a spreadsheet project that they were unable to categorise within the taxonomy. The results of t.e validation survey are shown in Table 9.

Table 9: Validation Survey returns

No:	Rater 1 A D E	Rater 2 A D E	Inter Rater Match A D E	Rater 1 6 wks later A D E	Time Match A D E
1	S6, D3, R1				
2	O3, I2, U2				
3	S4, D2, R1			S5, D4, R1	n n y
4	M3, D3, U2	M3, D3, U2	y y y	S3, D3, U2	n y y
5	M3, D3, U2			O3, D3, U1	n y n
6	S5, I1, U3				
7	S5, I2, U1				
8	S4, I2, U2				
9	M2, I1, U3	M2, I1, U3	y y y		
10	S1, D3, U1			S1, D3, U1	y y y
11	S5, D5, U1	S5, D5, U1	y y y	S5, D5, U1	y y y
12	M1, D3, U3			M1, D3, U3	y y y
13	M3, C2, R1			M3, C2, R3	y y n
14	S4, D3, U2			S4, D3, U2	y y y
15	O3, D3, U2	O3, D1, U2	y n y	O3, D3, U1	y y n
16	O3, D3, U3				
17	S1, I2, U2			S1, I2, U2	y y y
18	M2, D3, U1	S5, D3, U1	n y y		
19	O2, D3, U1	O2, D3, U2	y y n	O2, D3, U1	y y y
20	S4, I2, U2				
21	S2, D3, U2				
22	S6, D3, U2				
23	O3, D3, U2			O3, D3, U1	y y n
24	S6, I1, U2				

5.4.3. Inter-Judge Agreement:

The validation survey described above validated the A.D.E. taxonomy diagnostic key on inter-judge agreement. Six pairs familiar with a spreadsheet project used the key to categorise it. Table 9 shows that in three cases the categorisations were identical. In the other three cases the categorisations differed in one dimension only. In two of these cases the differences were probably due to the alternate rater's lack of knowledge rather than instrument failure i.e. a misunderstanding of what the instrument was attempting to measure.

In the developer Dimension, case 15 was categorised *D3* (knowledgeable) by the developer and *D1* (user-group member) by the alternate rater. This difference was not considered a failure of the diagnostic key but rather a rater failure, as only the developer would know if they were a user group member. Similarly in the environment division, case 19 was categorised *U1* (rushed) by the developer and *U2* (sufficient time available) by the alternate rater. The developer considered this a rushed job. The alternate rater verified on follow up that he had not known this. This was not considered an instrument failure.

In case 18, the ratings differed in the application dimension and there was no indication whether this difference was caused by rater or instrument failure. Case 18 was categorised *M2* (optimiser model) by the developer and *S5* (general report) by the alternate rater.

Table 9 validated the A.D.E. Diagnostic Key instrument by inter-judge agreement as in 15 out of 18 categorisations (83%), the raters agreed. It would have been useful to extend this inter-rater validity exercise to more cases, but apparently, no other developers in the validation sample had a suitable alternate rater available. It would appear that spreadsheet development in Preston is a comparatively lonely activity. This has implications for the control of spreadsheet development. Further validation of inter-rater categorisations would be appropriate on a reasonably sized random sample. This would require a further study using a sample frame of spreadsheet applications which have alternate raters available. Such a frame was unavailable for this study.

5.4.4. Agreement over Time

Table 9 also shows the validation of the A.D.E. taxonomy Diagnostic Key over time, when the same developers recategorised their project using the key, six weeks after its first categorisation with 28 out of 36 (78%) categorisations agreeing.

The eight categorisations which differed were examined. Three of the differences, i.e. cases 5, 15 and 23 were due to a change in the categorisation of the environment dimension from *U2* (adequate time) to *U1* (rushed development, i.e. the raters perceptions of the time available changed over six weeks. A further three of the differing categorisations appeared to be rater error:

a) the developer dimension of case 3 changing from *D2* (expert) to *D4* (novice)
b) the application dimension of case 3 changing from *S4* (corporate data creator) to *S5* (no corporate data)
c) the environment dimension of case 13 changing from *R1* (tight control) to *R3* (no control except library)

The final two differing categorisations on the application dimension are worthy of further consideration.

a) the application dimension of case 4 changing from *M3* (complex model) to *S3* (non 3D complex report)
b) the application dimension of case 5 changing from *M3* (complex model) to *O3* (report prepared for user data entry)

Users of the diagnostic key may well need more guidance in what a complex model is. This matter is considered further in the final chapter.

To summarise these findings: The taxonomy was validated by agreement by the same rater over time as 78% of the categorisations agreed. A further 8% differed on the perception of the time available for development, which was quite likely to have been reconsidered, after a six week gap. A further 8% of the differences appeared to be due to rater error, In only 2 cases (6%) was their doubt as to the instrument validity, due to the definition of what constitutes a complex model. Chapter 6 discusses the problem of measuring model complexity.

5.5. Validation of the A.D.E. Taxonomy

Mezzich and Solomon (1980, p. 33) suggested that taxonomies should be evaluated with respect to a) external criteria, b) internal criteria, c) replicability, d) stability and e) inter-rater assignment of cases to categories. The validation exercises described in this chapter used all five of these criteria. The taxonomy was validated with respect to both external and internal criteria. External criterion validity was demonstrated when the A.D.E. taxonomy was compared to other published taxonomies. Internal criterion validity was demonstrated when material drawn from within the Cluster Analysis process supported the appropriateness of the clustering representation of the underlying data structure, i.e. by the comparison of hierarchical and kmeans clustering solutions and the demonstration of within cluster homogeneity and between cluster heterogeneity.

Validation of the A.D.E. taxonomy and its diagnostic key involved:

a) Assessing content, construct and criterion referenced validity
b) Assessing other validities as suggested by the literature
c) Assessing the achievement of the secondary research goals of this study
d) Demonstrating the usefulness of the taxonomy

5.5.1. Taxonomic Intuitiveness

The A.D.E. taxonomy, or more particularly its Diagnostic Key, was validated for 'intuitiveness' by the validation survey described above. Developers were asked to comment on any difficulties they had fitting their spreadsheet into the taxonomy using the diagnostic key. More than half the respondents did comment and all except for one, reported no difficulty. The one report of difficulty concerned the categorisation of a model as complex.

The comparison with partial categorisations reported in the literature review in chapter 2, and the researcher's a priori expectations, both discussed later in this chapter, also validated the intuitiveness of the taxonomy.

5.5.2. Cluster Validity

Four aspects of the validity of the Cluster Analysis solution were considered

a) Non homogeneous data-set i.e. do clusters exist?

b) Between cluster heterogeneity

c) Within cluster homogeneity

d) Comparison of the dendrogram with the cophenetic correlation matrix

Non homogeneous data-set

Bock (1985) suggested several mathematical significance tests for distinguishing between homogeneous and heterogeneous populations:

a) The (sth) largest gap between observations

b) Their mean distance from the cluster centre

c) Minimum within cluster sum of squares if k-means used

d) Maximum F statistic - least squared error criterion

The output of the three SYSTAT Kmeans procedures used to develop the A.D.E. taxonomy reported the between and within cluster sums of squares and F-ratios. These were examined using Bock's tests c) and d) on the Kmeans output of the cluster analysis runs found in Appendix D.

The sample as described by the Application variables in run 24j exhibited some heterogeneity as the within cluster sum of squares for PWHATIF and POPTIM were zero. An F-ratio of 15.157 for XMACRO showed this variable was a significant discriminator between clusters. Other discriminators were THREED with an F-ratio of 9.268, and RUNBY with an F-ratio of 8.755.

The sample as described by the Developer variables in run 20q exhibited heterogeneity as the within cluster sum of squares for STCONS was zero. Other variables including EXPERT (8.360) and STSELFEM (5.797) also had low values for the within cluster sum of squares. Large F-ratios in STSELFEM (121.109), EXPERT (81.803) and OIT (70.636) also validated the heterogeneous nature of the sample with respect to the Developer variables.

The sample as described by the Environmental variables in run 25g exhibited heterogeneity as the within cluster sum of squares for SPERSN and LIBRARY were zero. ENUFTIME with a F-ratio of 197.922, and SDENFORC with a F-ratio of 119.567 were excellent discriminators between classes.

The data-set was heterogeneous when analysed using Environmental and Developer variables and showed slight heterogeneity when examined using Application variables. The variability of the data-set was established particularly regarding the environmental and developer dimensions. The spreadsheet applications were more similar, however they too showed sufficient variability to be analysed using cluster analysis procedures.

<u>Between cluster heterogeneity</u>

Dubes and Jain were concerned with the validity of individual clusters i.e. what made them different from the remainder of the data-set. They defined a valid cluster:

> A cluster is "real" if it forms early in the dendrogram for its size and lasts a relatively long time before being swallowed up. (1979, p. 250)

They cited Ling's (1973) method to measure the isolation of hierarchical clusters:

> measuring the compactness of a cluster by its birth size and measuring the isolation of an individual cluster by the cluster's lifetime. (Dubes & Jain, 1979, p. 250)

In a hierarchical solution, this method considers clusters are valid if they combine early and have a life for some time before being swallowed up by other clusters. An example of this technique for the Environment variables in run 25f, is shown below in Table 10.

The dendrograms and Kmeans output in Appendix D resulting from cluster analyses procedures performed on environmental variables, were used for the following analysis.

Table 10: Lifetimes of average link clusters for Environmental variables cluster analysis

Cluster		Birth Level	Size	Life-time	E
C1	(83,20)	0	2	0.86	
C2	(85,37,43)	0	3	0.86	
C3	(57,76)	0	2	0.86	
C4	(23,78)	0	2	1.4	
C5	(105,64,11,41, 92)	0	5	1.16	
C6	(103,88,74,62,52,28,10,3,7,21,47,53,70,87, 99)	0	15	1.16	U1
C7	(106,102,100,97,95,93,90,86,82,80,77,68,6 6,61,59,56.54,50,48,45,42,39.36.34.32.30,2 7,25,18,16,1,8,5,2,1,4,6,9.13.17.22.26.29.3 1.33.35.38..40,44,46,49,51,55,58,60,63,67. 69.79,81,84,89,91,94,96,98,101,104,107)	0	69	1.16	U2
C8	(71,14,24)	0	3	1.16	
C9	(65,75)	0	2	1.16	
C10	(C1,73)	0.86	3	0.31	
C11	(C2,C3)	0.86	5	0.31	
C12	(C5,19)	1.16	6	0.24	
C13	(C6,C7)	1.16	84	0.65	
C14	(C8,C9)	1.16	5	1.18	U3
C15	(C10,C11)	1.17	8	0.85	R2
C16	(C4,C12)	1.4	8	0.41	R3
C17	(C16,C13)	1.81	92	0.21	
C18	(C15,C17)	2.02	100	0.32	
C19	(C18,C14)	2.34	105	1.27	
C20	(72)	0	1	3.67	R1
C21	(C19,C20)	3.67	106	*	

If a subjective criterion for the lifespan of a valid cluster is established as 30% of the maximum possible cluster lifespan then clusters in Table 10 with a lifespan of greater than 30% of 3.67, (i.e. 1.1) can be considered valid. Clusters *U1, U2, U3* and *R1* all have lifetimes greater than 1.1 and so can be considered valid as they are isolated for more than 30% of the possible cluster lifetime. Cluster *R3* is a combination of clusters *C4* and *C12*, also conforms to the criterion as *C4* has a lifetime of greater than 1.1. Only cluster *R2* (loose environmental control) was not validated by this method. However *R2* was intuitively appealing as a counter balance to category *R1* (tight control) and was retained in the taxonomy.

Table 10 shows that most of the clusters used to form categories within the environmental dimension of the A.D.E. taxonomy had comparatively long lifetimes before being combined to form new clusters in the hierarchical tree dendrogram. This validates the clusters on the 'heterogeneity between clusters' criterion.

The same exercise could have been completed for Application and Developer variables. The exercise would have been more complex as in these cluster analyses, only two cases combined at each stage. i.e. two tables, each with 106 entries would have been required to complete the exercise shown above for Environmental variables using a table of just 21 entries. This was not completed. The exercise on the Environmental variables had validated the Cluster Analysis method. The Application and Developer dendrograms were scanned by eye as an alternative. Both demonstrated a reasonable degree of cluster isolation.

Within cluster homogeneity

This criteria considered the compactness of the partition. Dubes and Jain (1979, p. 251) suggested comparing within individual cluster dissimilarities with the average dissimilarity within the cluster and outside the cluster. The SYSTAT output of the Kmeans partitioning cluster analysis algorithm provides an intuitively easy way of determining this. The output shows, for each variable within a cluster, the minimum, mean, maximum and standard deviation. The variables were standardised across the whole data-set prior to analysis, giving for each variable, a mean of 0 and a standard deviation of 1. This allowed an easy comparison between a cluster mean and standard deviation, and that of the whole data-set. Standard deviations of 0 within a cluster showed that all cluster members had identical values for that attribute i.e. they were homogeneous over that attribute. The value of the mean on the Kmeans output, gave the value of the attribute. Then it could be determined if the mean value within the cluster was greater, less or similar to the mean value for the data-set as a whole.

The within cluster standard deviation from the Kmeans runs in Appendix D was checked for each attribute. For most clusters and variables this was below 1, i.e. less than the standard deviation of that variable measured across the whole data-set. This validated the clusters according to the 'within cluster homogeneity' criteria, as within a cluster, cases were more alike than across clusters.

Comparison of the dendrogram with the proximity matrix

Romesburg (1984) and Dubes and Jain (1979) discussed demonstrating the internal criterion referenced validity of a clustering solution by establishing the "Global fit of hierarchy", i.e. establishing the similarity between the dendrogram and the proximity matrix from which it was derived. The cophenetic correlation coefficient was suggested as a standard for comparison (Dubes and Jain, 1979, p. 245).

Using the SYSTAT software, the dissimilarity matrix was readily available but unfortunately the solution to the cluster analysis was only available as a dendrogram and not as the underlying cophenetic matrix. The joining distances of each branch of the tree were available and the cophenetic matrix could have been calculated from them. With 108 cases, the production of a cophenetic matrix would have involved determining the value of 108 x 108 / 2 i.e. 5,832 cells. As three such matrices were required, this method was considered too time-consuming.

An alternative method, involving the validation of just a few assignations of cases to clusters, was devised to demonstrate internal criterion validity. For each of the three Cluster Analysis solutions used to develop the A.D.E. taxonomy, runs 24a, 20m and 25f, a proximity matrix of dissimilarity coefficients was produced.

a) Remove case labels from the ordinal data-set
b) Select the attributes used to develop the taxonomy, discard the others
c) Transpose the matrix
d) Calculate the correlation matrix using Euclidean distances as the dissimilarity measure.

In each of the three (A, D, and E.) dissimilarity matrices, five of the smallest Euclidean distances between two cases were selected and the dendrograms were checked to see if both cases were allocated to the same cluster. Two high euclidean distances were also checked, to ensure the cases were assigned to different clusters. The results of this validation exercise are shown below in Table 11.

Table 11: Comparison of Euclidean Distance measure between cases and allocation to clusters in Cluster Analysis solutions used the develop the A.D.E. taxonomy.

ADE	Euclidean distance correlation coefficient	1st case	2nd case	1st category	2nd category * = Different
A	0.24	75	89	S5	S5
A	0.35	57	75	S5	S5
A	0.57	84	72	O3	O3
A	0.39	101	58	S5	S5
A	0.55	39	27	S4	S4
D	0	6	84	D3	D3
D	0	3	44	D3	D3
D	0.21	3	4	D3	D3
D	0.3	23	55	D2	D2
D	0.42	1	2	D4	D4
E	0	1	2	U2	U2
E	0	9	18	U2	U2
E	0	26	56	U2	U2
E	0	3	7	U1	U1
E	0	37	43	R2	R2
A	2.31	7	103	M3	M1*
A	2.29	71	38	M2	S1*
D	2.83	25	79	C1	D5*
D	2.49	40	76	I1	C3*
E	3.57	20	75	R2	U3*
E	4.36	24	72	U3	R1*

The first section of Table 11 shows cases with small Euclidean distance correlation coefficients, representing small inter-case distances i.e. low dissimilarity. These cases have been placed in the same cluster. The final section of Table 11 shows dissimilar cases with high Euclidean distance correlation coefficients which have been assigned to different clusters. These assignations validate the internal criterion validity of the taxonomy by comparing the correlation matrix from which it was derived with the dendrogram in an attempt to establish Dubes and Jain (1979) "global fit of hierarchy".

5.5.3. Taxonomic Stability and Robustness

The taxonomy was validated for stability and robustness by repeating the cluster analysis with the addition of extra variables showing minimum variability over the data-set. Two dummy variables with values 0 and 1 for all cases, were added to the ordinal data-set. The Kmeans and hierarchical dendrograms were similar to the results obtained without the addition of the extra variables.

Gordon (1981, p. 129) discussed Fisher and Van Ness's (1971) approach to validation based on decision theory admissibility concepts. His criteria for admissibility included:

a) Point proportion admissibility: Duplicate an object and demonstrate the same clusters are present

b) Cluster omission admissibility. Remove all objects in one cluster and demonstrate the remaining clusters are still present

Point proportion admissibility was demonstrated by duplicating three cases prior to reclustering. The original clusters were still present.

Cluster omission admissibility was demonstrated by the deletion of all objects from a medium sized cluster in the Application, Developer and Environment variable data-sets. The results where then compared with the cluster analysis solutions used to develop the A.D.E. taxonomy. Again there was no appreciable difference in the clusters obtained, except for the absence of the discarded cases.

5.5.4. Taxonomic Replicability

Ideally validation of replicability should have involved the collection and analysis of another data-set, leading to the development of a second taxonomy. This could then have been compared with the A.D.E. taxonomy. However this was considered too expensive in terms of financial and time resources, particularly as no suitable sampling frame was available.

Gordon (1981, p. 132) cites Cormack (1971) "if clusters are really distinct, it would be hoped that any strategy worthy of use would find them." He suggests that if several different classification procedures agree closely, you can have confidence in the results. The sample described by Application, Developer and Environment variables underwent Cluster Analyses, using both the hierarchical agglomerative and the Kmeans procedures. The close agreement in the results obtained by these two different methods as described in Sections 4.32, 4.34 and 4,36 for the Developer, Application and Environment dimensions, validated the A.D.E. Taxonomy under the 'replicability' criterion.

5.5.5. Comparison with other Published Taxonomies

Biggs and Collis (1982) suggested taxonomy validation via reliability tests i.e. how well the taxonomy agreed with others. The A.D.E. taxonomy was validated by comparing it to other partial taxonomies prepared by experts and reported in the literature. These comparisons for Application, Developer and Environment categories are now considered separately as external referenced criteria for validation of the A.D.E. taxonomy.

Application categories

The A.D.E. taxonomy subdivided applications into *Models (M1-M3)* and reports and other applications written for use by *Self (S1 - S5)* or *Others (O1 - O3)*:

- Models were further subdivided into 'what if' *(M1)*, optimiser (*M2)* and very complex *(M3)*.
- The *'S'* series of reports was further subdivided into three dimensional complex (*S1*), three dimensional simple (*S2*), creating graphics (*S6*), creating new corporate data (*S4*), complex reports (*S3*) and other reports (*S5*).
- The *'O'* series of reports was further subdivided into data entry by a data entry clerk (unimportant *O1* and important *O2* functions) and data entry by a non-developer user (*O3)*.

Ballou and Pazer (1985), West & Lipp (1986) and Ronen, Palley and Lucas (1989) all differentiated between models and reports designed for the developer or for others to run. i.e. *'M'*, *'S'* and *'O'* categories.

Eom and Lee (1990) identified optimiser *(M1)* and 'what if' *(M2)* models.

Karten (1989), Weber (1986), Nesbit (1985), Buckland (1989) and Eom and Lee (1990) all recognised the category of self-run spreadsheets that create new corporate data *(S4)*. Anderson and Bernard (1988) identified simple self run spreadsheets *(S2* and *S5)*. Anderson and Bernard (1988) and Shneiderman (1980) identified complex spreadsheet categories (*S1* and *S3*). Miller (1989) recognised the differences between two (*S3* and *S5*) and three dimensional (*S1* and *S2*) worksheets.

Anderson and Bernard (1988) and Schmitt (1988) identified the *'O'* series of spreadsheets created for others to run. Karten (1989) and Weber (1986) recognised the sub-categories of important spreadsheets used for significant business decisions, (*O2* and *O3*).

The only category of spreadsheets application not readily identifiable in this review of the literature, was complex models (*M3*). All other categories in the Application section of the A.D.E. taxonomy were confirmed by other authors.

Developer categories

The A.D.E. taxonomy categorised Developers as *Consultants* (*C1-C3*), other *I.T.* professionals (*I1- I2*) or other *Developers* (*D1 - D5*).

- The *'C'* series of consultant developers were further divided into I.T. professionals (spreadsheet specialists, *C1* or other I.T. consultants *C2*) and non I.T. professional consultants (*C3*)
- The *'I'* series of I.T. based developers were further subdivided into non consultant I.T. professionals who were disinterested (*I1*) or interested (*I2*) in spreadsheets.
- The *'D'* series of developers were subdivided into user-group members (*D1*), expert (*D2*), knowledgeable (*D3*), novice (D4) and self-employed (*D5*) developers.

Gordon (1981) cites Martin (1982) and McLean (1974) who differentiated between D.P. professional developers (*C1*, *C2* or the *'I'* series) and non D.P. developers i.e the *'D'* series. Moskowitz (1987b) also identified the *'C'* and *'I'* series of developers.

Rockart and Flannery (1983) and Kasper and Cerveny (1985) developed a taxonomy of end-users divided into end-users and supporters of end-users. They differentiated between non D.P. functional support personnel (*C3*), end-user computing support personnel (*C1*), and professional D.P. programmers (*C2*).

Rockart and Flannery (1983) categorised end-user developers according to expertise identifying lay expert *(D2)* and knowledgeable developers *(D3)*. Page-Jones (1990) and Shneiderman (1987) also categorised end-user expertise identifying *(D2) and (D3)* and novice developers *(D4)*.

The only categories of the Developer section of the A.D.E. taxonomy not explicitly validated through the literature review were user-group members *(D1)* and self-employed developers *(D5)*.

Environment categories

Spreadsheet Development Environments in the A.D.E. taxonomy were categorised as either controlled, ***Regulated*** (*R1-R3*) or uncontrolled i.e. ***Unregulated*** (*U1 - U3*) environments.

- The *'R'* series of regulated environments was subdivided into tight (*R1*) or loose *(R2)* control and the existence of a spreadsheet library *(R3)*.
- The *'U'* series of unregulated environments was subdivided into rushed development *(U1)*, normal time development *(U2)* and personal or recreational use *(U3)*.

Dart, Ellison, Feiler and Haberman (1987), and Schneider and Hines (1990) in their taxonomy of medical software, recognised the concept of regulated and unregulated environments the *'R'* and *'U'* series of the A.D.E. taxonomy. Perry and Kaiser (1991) identified the concept of policies imposed during the development process i.e. *R1* and *R2* environments.

Karten (1989) identified spreadsheets with a rushed development time *(U1)* while Eom and Lee (1990) identified spreadsheets for personal use *(U3)*.

Dart, Ellison, Feiler and Haberman (1987) discussed the concepts of 'programming in the large' and 'programming in the many'. 'Programming in the large' involved support for the developer beyond that required for a single spreadsheet e.g. the inclusion of programmer assistance provided by a spreadsheet template library *(R3)*. (libraries, however were not explicitly mentioned but the implication was there).

The Environmental section of the A.D.E. taxonomy was valid with respect to the 'external referencing' criterion provided by the literature as all categories were also identified in expert writings.

5.5.6. Comparison with A Priori Expectations

Comparison of the A.D.E. taxonomy with the researcher's a priori expectations provided a more objective benchmark than that provided by the posteriori rationalisation of results.

The A.D.E. taxonomy was compared with the researcher's a priori expectations, set out in a letter to the Head of Department of Computer Science at the then West Australian College of Advanced Education in 1989 prior to the commencement of this study. An extract from this letter is included for comparison:

> In my view there are three major factors categorising spreadsheets. Complexity, Strategic Importance and Usage. Each of these factors can be further decomposed. None should influence spreadsheet controls in isolation, it is the interaction between them that is important in deciding the degree and rigour of control necessary in a spreadsheet model.
>
> 1) Complexity
>
> a) Size
>
> b) Structure - number of dimensions
>
> c) Macros
>
> d) Active links to other worksheets
>
> 2) Strategic Importance
>
> a) Corporate Decision Support value - Low / High
>
> b) Sphere of influence
>
> c) Data / Information Flow through, Sink or Source
>
> 3) Usage
>
> a) Once / infrequent / frequent
>
> b) By developer / by others
>
> c) Expertise of users/ developer
>
> (M.J. Hall, personal communication, 1989)

This multi-dimensional taxonomy was restricted to the Application aspects of the A.D.E. taxonomy. Environmental aspects were completely ignored and the developer was mentioned only briefly under the 'Usage' category. The A.D.E. taxonomy does include reference to all my a priori categories with the exception of 'Size', however, they have been clustered in a different manner.

5.5.7. Taxonomic Usefulness

Everitt suggested that a taxonomy would be validated if members of different groups differed on variables other than those used to derive them; i.e. conversely, if members of the same category had a similar range of values for an attribute that had not been considered when defining the categories, and if that attribute had different values in other categories. Another possibility he canvassed was whether members of different groups would respond differently to a stimulus and members of the same group respond in a similar way to a stimulus (Everitt, 1980. p. 74).

The A.D.E. taxonomy was validated under Everitt's 'stimulus' and 'usefulness' criteria, when it was used to see if members of different categories responded similarly (i.e. pre-planned or not) to a stimulus (the need to develop a spreadsheet).

The question of interest was, which factors were associated with experienced developers pre-planning their spreadsheets on paper. Respondents' answers to question 61a in part 3 of the survey questionnaire were analysed. This question asked whether the spreadsheet had been planned on paper prior to its development. Seventy eight expert and knowledgeable developers were selected from the data-set i.e. all novices *(D4)*, self-employed *(D5)* and I.T. workers who were disinterested in spreadsheets *(I1)* were excluded. The remaining were considered to be experienced developers.

The first analysis computed contingency Table 12 showing the frequencies of un-planned, and pre-planned on paper spreadsheets, developed in regulated *(R1, R2* or *R3*) and unregulated (*U1, U2* and *U3)* environments.

Table 12: Spreadsheet survey, experienced developers. Frequency of pre-planning spreadsheets on paper for developers working in regulated and unregulated environments.

	Not pre-planned on paper	Pre-planned on paper	Total
Regulated Environment	1	11	12
Unregulated Environment	37	29	66
Total	38	40	78

A chi-square test for differences was performed;

> H_0: Experienced developers show no significant difference in their rate of pre-planning their spreadsheets on paper when developing in a regulated or unregulated environment.

χ^2 calculated = 9.258 (χ^2 critical = 3.842, α = 0.05, d.f.= 1) therefore reject H_0. As one of the frequencies was less than 5, the chi-square test may be inappropriate. Wilkinson (1990, p. 510) suggests the use of Fisher's Exact test in these circumstances. This two tail test had a significant p value of .003 confirming the rejection of H_0. Environment regulation and the pre-planning spreadsheets may be dependent.

Spreadsheets prepared by experienced developers may be pre-planned more frequently when developed in a regulated environment.

The second analysis repeated the first restricting the sample to spreadsheets that were not simple or trivial, i.e. discarding three-dimensional simple *(S2)* and general *(S5)* spreadsheets. The contingency table for this analysis is shown in Table 13.

Table 13: Spreadsheet survey, experienced developers developing non-trivial spreadsheets. Frequency of pre-planning on paper in regulated and unregulated environments

	Not pre-planned on paper	Pre-planned on paper	Total
Regulated Environment	0	8	8
Unregulated Environment	28	22	50
Total	28	30	58

A chi-square test for differences was performed:

> H_0: Experienced developers show no significant difference in their rate of pre-planning on paper when developing non-trivial spreadsheets in a regulated or unregulated environment.

χ^2 calculated = 8.661 (χ^2 critical = 3.842, $\alpha = 0.05$, d.f. = 1) therefore reject H_0. As one of the frequencies was less than 5, the chi-square test may be inappropriate. Fisher's Exact two tail test had a significant *p* value of .005 confirming the rejection of H_0. Environmental regulation and pre-planning non-trivial spreadsheets may be dependent.

When considering non-trivial spreadsheets prepared by experienced developers, they may be pre-planned more frequently when developed in a regulated environment.

This developer behaviour might have been associated with the time available for developing the spreadsheet. A third analysis restricting developers to those working in unregulated environments was conducted. The pre-planning practices of experienced developers, who considered they had sufficient time, and those who considered they were rushed, were compared in Table 14.

Table 14: Spreadsheet survey, non-trivial spreadsheets developed by experienced developers working in an unregulated environment. Frequency of pre-planning on paper, when a spreadsheet development is rushed or sufficient time is available for development.

	Not pre-planned on paper	Pre-planned on paper	Total
Rushed development	6	5	11
Sufficient time available	22	17	39
Total	28	22	50

A chi-square test for differences was performed:

> H_0: Experienced developers working in an unregulated environment, developing non-trivial spreadsheets, show no significant difference in their rate of pre-planning on paper when their project is rushed or has sufficient time available.

χ^2 calculated = 0.012 (χ^2 critical = 3.842, α = 0.05, d.f. = 1) therefore H_0 could not be rejected.

When considering experienced developers working in an unregulated environment, the pre-planning of non-trivial spreadsheets, may be independent of the time available for development. There was no significant difference in pre-planning, if the development was rushed or not.

As 'time available' alone was not associated with a difference in pre-planning practice, it was considered that the importance of the spreadsheet under development might be. The fourth and final analysis in this series, repeated the third analysis after removing all unimportant application, i.e. those with the variable IMPORTAN = 1 i.e. cases 4, 20, 27, 44, 57, 94, 97 and 99. The developers represented in this sample, where experienced and developed non-trivial, not unimportant spreadsheets. Their frequencies for pre-planning their spreadsheets in regulated and unregulated environments are shown in Table 15.

Table 15: Spreadsheet survey, non-trivial, not unimportant spreadsheets developed by experienced developers working in an unregulated environment. Frequency of pre-planning on paper for spreadsheets when rushed or sufficient time available for development.

	Not pre-planned on paper	Pre-planned on paper	Total
Rushed development	5	5	10
Sufficient time available	20	17	37
Total	25	22	47

A chi-square test for difference was performed.

> H_0. Experienced developers working in an unregulated environment developing non-trivial, not unimportant spreadsheets, show no significant difference in their rate of pre-planning their spreadsheets on paper when their project is rushed or has sufficient time available.

χ^2 calculated = 0.052 (χ^2 critical = 3.842, $\alpha = 0.05$, d.f. = 1) therefore H_0 could not be rejected. The time available for development and the pre-planning of non-trivial not unimportant spreadsheets in an unregulated environment may be independent.

When considering non-trivial, not unimportant spreadsheets developed by experienced developers, working in an unregulated environment, there was no significant difference in pre-planning if the development was rushed or not.

Interpretation

The first analysis showed that experienced developers were less inclined to pre-plan their spreadsheets when working in an unregulated environment. The second analysis was restricted to non-trivial spreadsheets and still found experienced developers less inclined to pre-plan their spreadsheets in an unregulated environment. The third analysis was restricted to unregulated environments and determined that whether there was sufficient time available or not, did not significantly effect the rate of pre-planning spreadsheets. The fourth and final analysis considered only important, non-simple spreadsheets developed by experienced developers working in unregulated environments. It found that there was no significant difference to the rate of pre-planning spreadsheets, whether the development was rushed or not.

The rate of pre-planning spreadsheets prior to development by experienced developers was shown to be independent of the spreadsheet complexity, importance and development time available. The only factor demonstrated in these analysis that had a significant influence on the pre-planning rate of experienced developers was the presence of a regulated environment. This has considerable implications for the control of spreadsheet development.

These four analyses validated the taxonomy under the 'usefulness' criterion. They demonstrated how all three parts of the taxonomy could be used to provide a framework for the comparison of spreadsheet development. The first analysis used the Developer categories of the taxonomy to discard developers who had low expertise. The Environmental categories were used to differentiate between spreadsheets developed in regulated or unregulated environments in all analyses. The Spreadsheet categories were used to identify and discard simple or trivial spreadsheets in the last three analyses and to discard unimportant spreadsheets in analysis four.

A further major validation of this taxonomy as to its usefulness is planned for a future project, extending the work of this study. This project is outlined in the final chapter. A spreadsheet control model consisting of design and control mechanisms will be formulated. The A.D.E. taxonomy together with the control model will be used to suggest appropriate design criteria and control mechanisms for spreadsheet applications.

5.6. Conclusion

This chapter discussed the validation of the data collection instruments and the A.D.E. taxonomy and its diagnostic key. The data-set was shown to be non-homogeneous and the clusters were demonstrated to be valid. The replicability, robustness and stability of the taxonomy were also validated. The taxonomy was validated with respect to external and internal criteria. It was compared to other taxonomies in the literature and to the researcher's a priori expectations. Finally the usefulness of the taxonomy was demonstrated.

CHAPTER 6: CONCLUSIONS, RECOMMENDATIONS AND FUTURE DIRECTIONS

6.1. Introduction

This chapter shows how this study has met the primary research goal of developing a special purpose taxonomy of spreadsheet application development and how this will lead to the achievement of the second primary research goal, i.e. improving the management and control of spreadsheet development projects. This study's findings are compared with those of other studies into end-user computing. Some questions remain unanswered and future research avenues to find some answers are suggested. The dissertation concludes by foreshadowing a future study to derive a 'distributed control model' for the management of end-user developed spreadsheets.

6.2. Summary of the Study

Context of this study

Chapter 1 outlined the context of this study. Personal Computing is the fastest growing sector of the computing industry. End-user computing can involve the development of spreadsheets by non-professional programmers working outside the traditional controls associated with application development within an I.T. department. This study set out to develop a taxonomy of the spreadsheet development process as a suitable taxonomy could not be identified in the literature. The A.D.E. taxonomy was intended to be of sufficient scope to be useful in categorising spreadsheet development projects, in order to suggest appropriate design and control measures.

Study method

Chapter 3 described a survey of spreadsheet development projects. This was conducted using a stratified but non-random sample chosen to represent the population variability, and explicitly including smaller, rarer categories of spreadsheet projects. The survey established measures of different attributes of the spreadsheet development process. These attributes were chosen for their suitability of use in developing a taxonomy that would be of relevance in the control of spreadsheet development.

The spreadsheet development projects, represented in *n* dimensional space by the values of their *n* attributes, were submitted to 150 cluster analyses with variable input parameters. The A.D.E. taxonomy of spreadsheet applications development and its diagnostic key were developed from these runs. Chapter 5 described the subsequent validation of this taxonomy.

Limitations of the study

The limitations of this study have already been detailed in Section 3.9 and the discussion on sample bias in Section 3.4.3. They are here briefly summarised for the convenience of the reader.

The major limitation of the A.D.E. taxonomy, lies in its intended use. It is a special purpose taxonomy that has been developed for use with a control model to suggest application appropriate design and control measures.

Another limitation, is the non-probabilistic base of the development of the taxonomy. As no complete frame of the spreadsheet project population was available, the taxonomy was developed from a non-probability based sample. The representativeness of the cases input to the cluster analysis has not been directly validated however the clusters obtained were shown in Section 5.5.5 to agree with those reported by other authors. Because of its basis in a non-probabilistic sample, the A.D.E. taxonomy should not be generalised to the population of all spreadsheet development projects without further confirmation using inferential statistical methods. The validation survey validated the use of the diagnostic key on a restricted sample, and this requires extension to a random sample of spreadsheet development projects.

Other limitations to this study's generalisability are provided by respondent bias, due to the inclusion of volunteers in the sample, and their self-assessment of their expertise and the importance of their work to their organisations.

These limitations do not lessen the usefulness of the taxonomy as a basis for future research, however they should be reconsidered whenever an attempt to generalise the findings of this study is made.

6.3. Results of the Study

The study results were detailed in Chapters 4 and 5. They are summarised here for convenience prior to a discussion on their implications. There were five main areas of results:

a) Sample statistics showing the variability of the sample are discussed in sections 4.2.5 and 6.3.1.

b) The A.D.E. taxonomy is discussed in sections 4.4 and 6.3.3.

c) Gender differences in spreadsheet developer expertise are discussed in sections 4.7 and 6.4.4.

d) Differences in pre-designing spreadsheets on paper in controlled and uncontrolled environments were discussed in detail in section 5.5.7 when taxonomic usefulness was validated.

e) Validation survey results described in section 5.4.2.

6.3.1. Sample Statistics

Developer organisations

The developers in the sample were drawn from all three strata; 60% from Preston, 30% from Perth and 10% Interstate. Less than 5% of the developers developed personal or recreational applications, 63.2% worked in the private sector and 32% in the public sector. The industries represented were almost evenly divided into four categories; mining, finance, education or computing, other. Developers tended to work for either small uni-departmental organisations (45%) or very large organisations with many departments (42%).

Developer

Most (85%) of the developers were male. They were older than might have been expected, with less than 10% under 25 and most (58%) over 35. The developers were well qualified with 71% having a degree and nearly half of these also having post-graduate qualifications. Half the developers were members of professional organisations e.g. Australian Computer Society or Australian Association of Accountants. About half the sample classified themselves as employees rather than management.

The developer's formal spreadsheet training was low. A higher than expected 52% of the developers were self trained and a further 8% were trained by workmates leaving only 40% of the sample who had received professional training in spreadsheet development. Most of the developers had a comparatively low interest in spreadsheets with only 11% belonging to a spreadsheet user-group and most (60%) reading less than three articles a year about spreadsheets. However a definite subset of about 20% were very interested in spreadsheets.

Software

Most applications were DOS based and about 60% were developed using LOTUS 123 or a clone. 21% of the spreadsheet applications used Excel. Most spreadsheets were developed using stand-alone packages, although a few (15%) used integrated packages.

Environmental controls and regulations

There was minimal regulatory control in the spreadsheet development environment. 11% of developers were aware of a spreadsheet development policy within their organisation but only a third of them had a copy of this policy. Controls, if they existed, were usually self-enforced and only one respondent reported I.T. departmental involvement. No respondent specified that a spreadsheet control policy was enforced by an auditor. A few developers (8%) had access to libraries of quality spreadsheets. A worrying 18% of spreadsheets had a rushed development, which may have resulted in a lack of care and inclusion of user-defined controls.

Applications

In spite of the lack of control reported in the sample, most applications (92%) were classified by their developers as of moderate or major importance. Nearly half the spreadsheets created new corporate data and a further 27% modified existing shared data. Only 17% of the spreadsheets produced information solely for the developer's own use. The output of the reminder was passed on to others, even beyond the developer's organisation in 29% of cases. The spreadsheet output remained in circulation for greater than a month in half the sampled cases. Applications tended to be run regularly (67%) with a further 16% being run occasionally after a long gap. Most templates were developed to be self-run, however 10% were prepared for data entry by a clerk, and a further 18% for running by another user.

Spreadsheets varied considerably in size and complexity. The developer categorised formulas as simple in less than half the sample. Logical 'if' functions, links to other applications, graphs and macros were well represented.

Summary

The sample consisted largely of important spreadsheets developed in environments where regulation was almost non-existent, by developers who had a 60% chance of having had no formal spreadsheet training. Chapter 2 discussed reports of about a 30% error rate in spreadsheets. The need for controlling spreadsheet development is apparent.

6.3.2. Comparison with other Studies

A survey restricted to spreadsheet development, could not be identified in the literature, however broader surveys of end-user computing have been conducted by several researchers, and their results are comparable to the results of this study.

Rockart and Flannery's study of end-users

Rockart and Flannery (1983), working at the Sloan School of Management at M.I.T., selected seven major organisations and interviewed 200 end-users and 50 I.T. professionals who supported these end-users. Their sample was not random and was not restricted to spreadsheet developers. Although their survey is now dated, a comparison of some of their findings with that of the current study is of interest.

Table 16 compares the range of output of the spreadsheet applications of this study with the end-user developed general applications surveyed by Rockart and Flannery

Table 16: Spreadsheet Survey. Comparison of Application scope with that reported by Rockart and Flannery

	Rockart and Flannery	This study
Beyond the organisation		30%
Multi-departmental	17%	22%
Single Department	52%	31%
Personal	31%	17%

The current study shows a trend away from purely personal applications towards applications with a wider distribution. This is in line with the increase in popularity of end-user computing over the last ten years.

Table 17: Spreadsheet Survey. Comparison of Primary Source of Data with that reported by Rockart and Flannery

	Rockart and Flannery	This study
Electronic Transfer	36%	9%
Keyed in ex reports	34%	42%
Private data	17%	39%
Other	13%	10%

Rockart and Flannery's study of end-user computing showed a much higher rate of electronic transfer of data than this study. More of the applications in this study dealt with only private data. Rockart and Flannery's developers were those identified as "heavy and or frequent users of time-sharing" (1983, p. 778) i.e. probably working on mini computers or mainframes. Today's P.C. based spreadsheet developers are less likely to be working with electronically downloaded corporate data.

TABLE 18: Spreadsheet Survey. Comparison of frequency of use of applications with that reported by Rockart and Flannery

	Rockart and Flannery	This study
One shot	6%	4%
Daily	6%	7%
Weekly	12%	11%
Monthly	10%	29%
As needed	66%	49%

The frequency of use of applications in this study shown in Table 18 was similar to that reported by Rockart and Flannery.

Rockart and Flannery reported a use of graphics in only 10% of their applications. The current study reports graphics used in 38% of applications. This increase could have been expected. Graphics are now easily accessible in modern spreadsheet packages, and the increased use of graphical user interfaces running on readily available and by now comparatively inexpensive, supporting hardware has popularised the use of graphics.

Rockart and Flannery categorised their end-users. Table 19 shows a comparison of their end-user categorisations matched with categories from the developer dimension of the A.D.E. taxonomy.

TABLE 19: Spreadsheet Survey. Comparison of developers with the end-user categories reported by Rockart and Flannery

Rockart and Flannery's End-users		This Study's D dimension	
Other	9%	D5	9%
Command level End-users	16%	D4	15%
end-user programmers	21%	D1 + D2 + D3	65%
functional support personnel	38%	C3	3%
end-user computing support persons	5%	C1 + C2	3%
DP Programmers	11%	I1 + I2	5%

The current study did not explicitly differentiate between end-user programmers and functional support personnel in the developer dimension, rather using the application dimension to differentiate between their products. If these two categories are combined, Rockart and Flannery's 59% is not dissimilar to this study's 68%. There were less professional I.T. persons in the current sample (i.e. 5% as against 11%). This seems reasonable as Rockart and Flannery's sample was not random and they had explicitly targeted I.T. professionals and end-user support persons.

Rockart and Flannery noted structures and processes that were absent from the seven large organisations where their survey was conducted. (1983, p 781)

- A strategy for end-user computing
- Development of end-user computing priorities
- Policy recommendations for top management
- Control methods for end-user computing

Rockart and Flannery make several recommendations including the distribution of technical support to departmental level. They considered that the control of end-user computing should not reside with I.T. personnel but rather be distributed to the functional line managers. I.T. personnel still have a part to play in aiding line management in deciding whether an application is suitable for end-user development, suggesting software and controls, and undertaking technical consultancy when requested to do so.

Rockart and Flannery suggested that I.T. personnel should have input to the development of an end-user computing environment. The establishment of standards and controls, with motivational incentives for end-user compliance, should be the responsibility of the I.T. professional.

Powell and Strickland's study of microcomputer security

Chartered Accountants Powell and Strickland, surveyed half the Forbes' 1987 list of the 1,004 largest American public companies trying to assess data security in a microcomputer environment. They received responses from 108 companies or 22% of those canvassed. Among other issues, their survey canvassed controls over application development. (Powell and Strickland, 1989, p. 22)

Powell and Strickland queried the existence of a company micro-computer security awareness program:

> The primary objective of a security awareness program is to keep microcomputer users, who are often previously inexperienced in computer applications, informed of the necessity to follow procedures that will maintain the security of data. (1989, p. 21)

Less than half these large, successful companies had such a program. Among those that did have a security awareness program, it was only documented in 69% of cases. Powell and Strickland report that in 13% of the companies, the control policy was not disseminated to the end-user. Less than one quarter of the companies provided a security education program for end-users. The awareness of the end-users in this survey of security and control procedures may well have been even lower than reported, as Powell and Strickland's respondents were not the end-user developers themselves, but the chief financial officers of the chosen companies, who presumably were responsible for the implementation of controls.

Powell and Strickland asked if controls were applied to application development:

> Is the development of new major applications for microcomputers controlled so as to ensure proper design, inclusion of control features and prevention of duplication of effort by different individuals or departments within the company? (1989, p. 23)

The results of Powell and Strickland survey of controls for major applications are compared with the non-trivial applications of the current study in Table 20.

The current study identified a spreadsheet library in 9% of cases surveyed. It queried end-users rather than their managers and found that a spreadsheet development and control policy existed in only 11% of cases, with one third of the end-users having a documented copy. In one third of the cases, where there was a spreadsheet development policy, it was enforced by the developer's line manager. The I.T. department was involved in only one case. No auditor involvement was reported, i.e. the majority of the cases were controlled solely by their developer.

Table 20: Application development policy for non trivial applications: Comparison of the results of the spreadsheet survey with Powell and Strickland's 1989 survey of microcomputer environments.

	Powell and Strickland	This study
Application control policy exists	34%	11%
Documented Control Policy exists	23%	3%
Control by IT department	16%	1%
Control by internal auditor	4%	0%
Application library exists	6%	9%

Powell and Strickland's rate of control was low, but still much higher than that shown by this study. Powell and Strickland surveyed financial managers rather than end-users. They restricted their sample to large, very successful companies, and important applications rather than the broader variety of companies and applications covered by this study. While the current study's figures are lower than the figures reported by Powell and Strickland, the same trend to lack of regulation is apparent, confirming Powell and Strickland's findings.

Like Rockart and Flannery, Powell and Strickland suggest control procedures for microcomputer application development. They too suggest distributing control to functional "business units". They suggest that:

> Because microcomputer users do not necessarily understand or appreciate controls, they must be educated on the importance of security controls and should be required to follow written control policies. (1989, p. 23)

The current study confirmed the results of the prior surveys of Rockart and Flannery, and Powell and Strickland. The conclusions reached by both sets of authors involved the distribution of the control of end-user computing away from a centralised I.T. department to the functional area where the developer works. Section 6.4.1 describes how a control model to achieve this might be developed, using the A.D.E. taxonomy.

6.3.3. A D E Taxonomy

The purpose of the A.D.E. taxonomy is to categorise spreadsheet development projects prior to suggesting application appropriate controls. Chapters 3 and 4 described the development of this taxonomy in three dimensions:

- A - the Application
- D - the Developer
- E - the development Environment

A detailed description of each category in the taxonomy can be found in section 4.4.1 and will not be repeated here. The survey sample showed considerable variability when described by the taxonomy. Table 21 below, shows the variation of the sample when categorised in the application, developer and environment dimensions. This variability is shown graphically in figures 4.24, 4.25 and 4.26 of chapter 4.

Table 21: Spreadsheet survey. Percentages of respondents in each category of the A.D.E. taxonomy

Application	M1	M2	M3	O1	O2	O3	S1	S2	S3	S4	S5	S6
%	8	5	1	2	8	12	2	4	3	20	28	8
Developer	**C1**	**C2**	**C3**	**D1**	**D2**	**D3**	**D4**	**D5**	**I1**	**I2**		
%	1	2	3	7	8	51	14	9	3	2		
Environment	**R1**	**R2**	**R3**	**U1**	**U2**	**U3**						
%	1	8	8	14	65	5						

The sample showed a broad variation in the type of applications developed. The developer dimension was less varied with just over half the sample categorised as *D3* (knowledgeable). In the environment dimension, the sample exhibited an extremely low rate of environmental regulation, with 8% categorised *R2* (loose control) and only 1% of the cases categorised as *R1* (tight control). 65% of the cases were categorised *U2* (no control, adequate time) and a worrying 14% of developers were categorised *U1* (no control, rushed job).

The validation of the taxonomy was discussed in chapter 5. The taxonomy was validated with respect to construct, content and external and internal criterion referenced validity. It was validated on inter-judge agreement and by the same rater after a time lapse. It was also validated with respect to the secondary research goals and usefulness. The A.D.E. taxonomy was compared to other taxonomies reported in the literature and all the categories of the A.D.E. taxonomy were confirmed by other authors except the application category *M3* representing complex models.

Category *M3* had only one member in the sample, but was retained as a separate category in the taxonomy as it was so different from all other clusters. It easily qualified under Dubes and Jain's (1979) definition of a valid cluster, as it was born at the first join of the dendrogram, and had a long lifetime, remaining isolated from all other categories until the second last join of the dendrogram. However respondents in the validation survey had problems with assigning projects to this category, (see section 5.4.4) and clearly more work is required to establish metrics for assessing the complexity of a spreadsheet application. This matter is discussed further in section 6.4.3.

6.3.4. Lack of Environmental Control

The major finding in the study was the low incidence of any form of environmental control (11%). This was of concern, considering the significance of the applications developed and the fact that only 40% of the developers had received professional spreadsheet training. With the likelihood of spreadsheet errors, clearly some form of control of the spreadsheet development process is desirable.

Pre-designing applications on paper prior to implementation is an appropriate control for some categories of spreadsheet development projects. The exercises to validate the usefulness of the taxonomy described in section 5.5.7. had shown that the only factor that encouraged experienced developers to pre-design significant spreadsheets on paper prior to implementation, was the presence of environmental regulation or the existence of control procedures.

The studies reported by Rockart and Flannery, and Powell and Strickland had both suggested the distribution of the control function to the functional work area of the end-user developer. They had suggested that the responsibility for assuring such controls are adhered to, be given to the functional line manager, rather than the I.T. department. Clearly both the end-user and their manager will need guidance as to suitable design features and controls to include in spreadsheet projects.

> The growth of end user computing in organisations is inevitable and management cannot effectively prohibit its use. Indeed major opportunities may be lost if an antagonistic stance is adopted. Consequently management should seek to formulate policies for end user computing that can be promulgated and enforced throughout their organisation. (Weber, 1986, p. 159)

6.4. Recommendations for Future Research

The study, due to its predisposition to data exploration rather than hypothesis testing, has highlighted a considerable number of areas for further research.

6.4.1. Development of a Control Model

The necessity for a control model to assist in the management and control of spreadsheet development projects has clearly been established in this dissertation. The lack of environmental regulation, and the importance of the applications being developed, highlights the need for a 'protocol' that the developer can use to suggest the appropriate design and control measures for their spreadsheet application. Thus the responsibility for control should be transferred from the centralised I.T. department to the functional business area and the end-user developer.

> Distribute or "download" responsibilities together with the distribution of processing capability. It is fruitless to hold the information systems department responsible for matters that are completely out of its control. Each individual must be held accountable for what he or she is doing. (Krull, 1986)

A study could be conducted to develop a model of suitable controls for developers to include in their spreadsheets. This study would build upon the results of the current study. Suggested controls for microcomputer spreadsheet development have already been collected by reviewing the literature and were included in the third section of the data collection questionnaire used in the current study (see Appendix A).

Issues canvassed included:

a) Spreadsheet Design

b) Formula issues

c) Input data control

d) Output data control

e) Review and Testing

f) Documentation

g) Security Issues

Survey respondents recorded which spreadsheet controls and design measures they had used, and their opinion whether they were unnecessary, useful or essential for their particular type of spreadsheet. This data held in the CONTROLS database will form the basis of the proposed control model.

The current study categorised survey respondents' spreadsheet projects using the A.D.E. taxonomy. Romesburg's (1984, p. 54) method could be used to develop the control model. The appropriateness of a specific control for a particular category in the taxonomy will be hypothesised. e.g. three dimensional spreadsheets (*S1, S2*) require compilation to prevent accidental alteration. Contingency tables, using the data from the CONTROLS database, will be used to test the hypothesis. This will establish if there is a statistically significant relation between the A.D.E. category and the qualitative variables representing the inclusion of a control. Where such a significant relation exists, the design and control criteria will become part of the control model for that particular category within the A.D.E. taxonomy.

Not all cases in the CONTROLS data base will be suitable for use in defining the control model. e.g. the developer dimension of the A.D.E. taxonomy might be used to exclude the opinions of novice developers. Certain categories of spreadsheet projects are sparsely represented in the sample and an effort will be made to target specific categories where more cases are required, and collect more data.

The Control Model will not attempt to recommend rigorous control for all spreadsheet applications. It will still allow end-users to be creative with their personal computers. However certain categories of spreadsheets do require control and the model will identify relevant controls where appropriate.

The resulting control model will require to be refined. Interviews will be held with both academic and industry based experts in appropriate disciplines, including end-user computing, software quality assurance, risk management and security. Spreadsheet experts and knowledgeable users will be identified, and be asked to categorise samples of their work within the A.D.E. Taxonomy. They will then be shown the list of model recommended spreadsheet controls, and be asked to validate each control's appropriate usage for their particular spreadsheet and to suggest other appropriate controls.

A profile of expert validity will be gathered for each category in the A.D.E. Taxonomy of Spreadsheet Applications Development and will be packaged into a Spreadsheet Development Control Model. This control model can be used with the A.D.E. taxonomy by end-user developers and their line managers, to suggest application appropriate spreadsheet control and design criteria.

This control model will allow the distribution of the control of end-user developed spreadsheets away from a centralised I.T. department to the functional business units where the end-user developer works. It could be used by a functional line manager, and is also appropriate for use by the developer i.e. distributing control 'to the coalface'. This further validates the usefulness of the A.D.E. taxonomy and the primary research goal of improving Australian spreadsheet development practice.

6.4.2. Confirm the A.D.E. Taxonomy

The A.D.E. taxonomy requires further confirmation. This could be achieved by a repeat study using either similar or new cluster analysis algorithms on a fresh data-set. If the data set could be based on a random sample, inferential statistical methods could be used to generalise the taxonomy to the population of all spreadsheet development projects.

Alternatively, artificial Intelligence pattern recognition techniques either using a neural network or Michalski and Stepp's (1983a) method of conceptual clustering could be used to cluster either the original, or a new data-set.

The continued attempt to invalidate the A.D.E. taxonomy through falsification, i.e. finding a case that cannot be fitted into a category, is also appropriate.

6.4.3. Spreadsheet Metrics

This study has highlighted the need for metrics to measure variables associated with the spreadsheet development process. Some metrics, applicable to general software application development have been reported in the literature, but they are often unsuitable for use by end-user developers to evaluate their spreadsheet projects. Further research to establish suitable metrics is required.

Spreadsheet Complexity

The identification of spreadsheet complexity and metrics for measuring it, have posed problems throughout this study. The term 'complex model' also caused difficulty for end-users in the validation survey. Section 2.9.7 discussed definitions of application complexity in the literature and defined spreadsheet complexity as used in this study. This comprised design, formula, link and logical complexity. Section 3.5.6 expanded on this definition to produce super-variables that measured complexity. Complexity of the user interface, was not included but is also worthy of consideration. More work needs to be done in this area and end-users and computer professionals require metrics to assess the complexity of spreadsheet applications.

Template Size

Measuring the size of a spreadsheet can be done in different ways. The file storage size, the number of occupied cells, the product of rows, column and dimensions etc. The problem is compounded as different spreadsheet products have different file structures for storing spreadsheets. Some store only occupied cells, while others store all cells. Macros and graphics are treated differently by different spreadsheet products. Some products use data compression techniques. This study recognised the problem and introduced an ordinal scale based on the 'useful' size of a spreadsheet i.e. the number of cells containing data or formulas, ignoring cells that were blank, contained labels or constants. A simple to use metric needs to be developed to measure spreadsheet size.

Application Criticality

The survey respondents reported the importance of the application to their organisation subjectively by categorising it as 'unimportant' or of 'moderate' or 'major' importance. In arriving at this decision, they were asked to consider the value of the decisions made using the spreadsheet and the ramifications to their organisation should the spreadsheet contain errors. The distribution range of the spreadsheet output and its creation or modification of corporate data were considered separately. The number of times a template was used, who used it, who entered data and the retention of the data were all considered. Application criticality needs further investigation and metrics are required to measure it.

Developer Expertise

Developers also subjectively categorised their spreadsheet development expertise using Shneiderman's (1980) terminology of 'novice', 'knowledgeable' or 'power user'. Sections 6.4.4 and 4.7 identified possible problems for women with this terminology as some respondents reported they were uncomfortable categorising themselves as a 'power user' as they disliked the association of expertise with power. Expertise is a difficult feature to assess particularly for an end-user who may have no overall understanding of the variation within the spreadsheet developer population. Qualifications, training, experience, time taken to complete a standard task, error rate etc. could be used to measure expertise. Further work to develop a metric is required.

6.4.4. Hypotheses Generation

The exploratory data analysis nature of this study has lead to the generation of hypotheses for testing in future studies, using inferential statistical methods.

The A.D.E. taxonomy divides spreadsheets into models and reports. An analysis of the sample data in section 4.4.2 and Table 6 suggested that models were more likely to be developed in an unregulated environment. This leads to a hypothesis:

> H_0: Spreadsheet models are no more likely to be developed in unregulated, than regulated environments.

Section 5.5.7 established the usefulness of the taxonomy in analysing the pre-designing tendency of non-novice developers developing important spreadsheets. Developers in this sample were more likely to preplan their spreadsheets when developing in an unregulated environment. This leads to the hypothesis:

> H_0: There is no difference in the rate of preplanning spreadsheets on paper for expert developers working in regulated or unregulated environments.

This dissertation has assumed that the application of controls will reduce spreadsheet error rates. This assumption has not been tested, and will require testing for each suggested design and control criteria, involving a large body of work.

> H_0: There is no difference in the error rate of spreadsheets where control '*n*' is applied or not applied.

Gender inequity among spreadsheet developers was explored in section 4.7 and Appendix E. Women in the sample reported a much lower expertise than men did. Developer gender was independent of the status, qualification or training of the developer, the importance of the task, or the size of the organisation where the developer worked. Women in the sample did not seem disadvantaged in their work functions or be less prepared for performing their duties. Yet women still perceived they had a low spreadsheet development expertise. This matter is worthy of further investigation using measures for expertise other than developer self-rating to test the hypothesis:

> H_0: There is no difference in the spreadsheet development expertise of women and men.

Appendix E discussed how men tended to design larger more complex spreadsheets. This could be a measure of the expertise of the developer, with developers of higher expertise, designing more complex spreadsheets. An alternative interpretation is possible, with the expert developers avoiding large and complex spreadsheets, rather restricting their templates to smaller cohesive worksheets possibly linked to other spreadsheets. Moskowitz attributes the following to Dale Christensen product manager for Microsoft Multiplan:

> Anyone who thinks they understand what is going on in a model bigger than 100 by 100 cells is probably fooling themselves. (Moskowitz, 1987b, p.36)

Structured software development promotes the concept, that small is manageable. These considerations lead to a hypothesis worth testing:

> H_0: The complexity of a spreadsheet is not related to the expertise of its developer.

If this hypothesis can be rejected, it would be interesting to determine whether more expert spreadsheet developers tend to build larger or smaller spreadsheets, than less expert developers.

6.5. Implications of this study for Spreadsheet Development Practice

This study has considerable implication for the management of spreadsheet development practice. It has described current spreadsheet development practice. It has established the variability of spreadsheet development projects. It has highlighted the serious situation of important spreadsheets being developed in almost completely unregulated environments by developers who have a high probability of not having undergone formal spreadsheet training. The validation survey also highlighted the loneliness of the spreadsheet developer when it had difficulty in finding a second person familiar enough with a spreadsheet, to act as an alternate rater. Another point of concern was the higher than expected 14% of developers who reported that they did not have sufficient time available for the development of their spreadsheet application.

Organisational spreadsheet control policies were in place in 11% of the respondents' organisations but only 3% of developers had a documented copy of this policy. If the policy was enforced, it was enforced either at the departmental level or by the developer. Only 1 developer out of 107 reported the involvement of the I.T. department in validating their spreadsheet and none reported internal auditor involvement.

Spreadsheet development would appear to be a lonely, uncontrolled activity with few checks and balances applied. Clearly spreadsheet development policies are required and to be effective, they should be designed to assist end-user control of their own spreadsheet development projects.

This study has developed the first part of a tool to be used to solve these problems. The A.D.E. taxonomy will allow the categorisation of spreadsheet projects by the developer prior to implementation. The development of the second part of the tool - a control model, has been foreshadowed.

This study should result in an improved awareness for those responsible for the management of spreadsheet development.

6.6. Conclusion

The primary research goals of this study established in Section 1.4.1 involved the improvement of the planning and management of spreadsheet development projects, and the development of a special purpose taxonomy of spreadsheet application development, for use in controlling spreadsheet development. These goals have been achieved with the development of the A.D.E. taxonomy and the foreshadowing of its use in a control model.

The secondary research goals of this study were established in three groups in section 1.4.2. The first group of these involved the construction of a sampling frame, exploratory data analysis and hypothesis generation, all of which have been achieved. The second group involved finding clusters that were intuitive, well structured and suitable for developing a taxonomy. These goals were also attained. The third group of secondary research goals considered the validation of the taxonomy and its diagnostic keys in terms of stability, robustness, replicability, agreement with other taxonomies in the literature and with my own a priori expectations. The final goal involved demonstrating the usefulness of the taxonomy which has been established both with the analysis of developer pre-designing tendency and with the foreshadowed development of a control model. These goals were also realised.

The study set out to implement a project to produce a product and satisfy research goals. This has been achieved, but the study also produced more than originally foreseen, highlighting areas of current spreadsheet development practice that are a cause of concern and opening up avenues for future research and development.

To conclude on a personal note, the work involved in preparing this dissertation has increased my knowledge of the research process, particularly data collection, multivariate statistics and clustering procedures. I have realised that the study of structure within data has much in common with the Computer Science discipline of Informatics particularly Data Modelling, which also seeks to gain an understanding of structure using techniques such as Entity Analysis (E.R. modelling) and data normalisation. Both Data Analysis in the computer science frame of reference, and Cluster Analysis when considered from a statistical point of view, seek to let the data 'speak' for itself and bring out its underlying structure. Both disciplines have the same goal.

The final words of this dissertation are borrowed from Winston Churchill's My early life:

> Thus I got into my bones the essential structure . . . which is a noble thing.

References

Anderberg, M.R. (1973). Cluster analysis for applications. New York: Academic Press.

Anderson, K. & Bernard, A. (1988). Micros in accounting: Spreading mistakes. Journal of Accounting, 3(4), 42-45.

Arabie, P., Douglas, C. & Desararbo, W.S. (1987). Three way scaling and clustering. Newbury Park, CA: Sage.

Ashworth, A. (1987). Building model models (spreadsheet programs). Accountancy (GB), 99(1122), 136-137.

Bailey, R.W. (1982). Human performance engineering: A guide for systems designers. Englewood Cliffs, NJ: Prentice Hall.

Bailey, R.W. (1983). Human error in computer systems. Englewood Cliffs, NJ: Prentice Hall.

Ballou, D.P. & Pazer, H.L. (1985). Modelling data and process quality in multi-input, multi-output information systems. Management Science, 31(2), 150-162.

Ballou, D.P., Pazer, H.L., Belardo, S. & Klein, B. (1987). Implication of data quality for spreadsheet analysis. Data Base (USA), 18(3), 13-19.

Beitman, L. (1986). Reviewing electronic spreadsheets. EDPACS (USA), 13(10), 8-9.

Benson, D.H. (1983). A field study of end user computing: Findings and issues. MIS Quarterly, 7(4), 35-45.

Berry, T. (1986, December). Who's to blame for spreadsheet errors? Business Software, p. 36-37.

Biggs, J.B. & Collis, K.F. (1982). Evaluating the quality of learning - The SOLO taxonomy (Structure of the Observed Learning Outcome. New York: Academic Press.

Bloom, B.S. (Ed.), Englehart, M.D., Furst, E.J., Hill, W.H., & Krathwohl, D.R. (1956). Taxonomy of educational objectives - Book 1 cognitive domain. London: Longmans.

Bock, H.H. (1985). On some significance tests in cluster analysis. Journal of Classification, 2, 78-107.

Bromley, R.G. (1985, December). Template design and review: How to prevent spreadsheet disasters. Journal of Accountancy, p. 130-142.

Brown, P.S., Gould, J.D. (1987). An experimental study of people creating spreadsheets. ACM Transactions of Office Information Systems, 5(3), 258-272.

Bryan, M. (1986, December). Bug-proofing your spreadsheets. Business Software, p. 38-41.

Buckland, J.A. (1989). Supporting the microcomputer end user. In J.E.W. Masterson (Ed.), Critical issues in information processing, management and technology (Vol. 3) (pp. 63-170). Wellesley MA: QED Information Sciences.

Canning, R.G. (1984). Coping with end user computing. EDP Analyser, 22(2), 1-12.

Card, S.K., Moran, T.P. & Newell, A. (1983). The psychology of human-computer interaction. Hillsdale, NJ: L. Erlbaum & Associates.

Chambers, A.D. & Court, J. M. (1986). Computer Auditing (2nd ed.). Sydney: CCH Australia.

Chan, W. (1987, December). Sorting out spreadsheets. Australian Accountant, p. 49-53.

Clifford, H.T. & Stephenson, W. (1975). An introduction to numerical classification. New York: Academic Press.

Connors, S.G. (1984, November). Microcompu' : software: Who uses what? NAA Research. Management Accounting, p. 16, 65.

Cotterman, W.M. & Kumar, K. (1989). User cube: A taxonomy of end-users. Communications of the ACM, 32 (11), 1313-1320.

Creeth, R. (1985, June). Microcomputer spreadsheets: Their uses and abuses. Journal of Accountancy, p. 90-93.

Dart, S.A., Ellison, R.J., Feiler, P.H. & Habermann, A.N. (1987). Software development environments. Computer, 20 (11), 18-28.

Davies, N. & Ikin, C., (1987). Auditing spreadsheets. Australian Accountant, 57(11), 54-56.

Davis, D. & Cosenza, R.M. (1985). Business research for decision making. Boston MA: Kent.

Diday E. & Simon J.C. (1976). Clustering analysis. Communication and Cybernetics, 10, 47-92.

Ditlea, S. (1987). Spreadsheets can be hazardous to your health. Personal Computing, 11(1), 60-69.

Dubes, R. & Jain, A.K. (1979). Validity studies in clustering methodologies. Pattern Recognition, 11, 235-254.

Dunn, G. & Everitt, B.S. (1982). An introduction to mathematical taxonomy. Cambridge: Cambridge University Press.

Edge, W.R. & Wilson, E.J.G. (1990). Avoiding the hazards of microcomputer spreadsheets. Internal Auditor, 47(2), 35-39.

Eom, Hyun, B. E. & Lee, Sang. M (1990). A survey of decision support system applications (1971-April 1988). Interfaces, 20 (3), 65-79.

Everitt, B. (1980). Cluster analysis (2nd ed.). New York: Halstead Press.

Fisher, D. & Langley, P. (1986). Conceptual clustering and its relation to numerical taxonomy. In W.A. Gale (Ed.), Artificial Intelligence and Statistics (pp.77-115). USA: Addison Wesley, AT & T Bell Labs.

Flower, J. R. (1988). Arithmetic errors in spreadsheet arithmetic. Signum Newsletter (USA), 24(1), 13-14.

Fox, J.M. (1982). Software and its development. Englewood Cliffs, NJ: Prentice Hall.

Foye, P. (1989, September). Spreadsheet quickie. Printscreen, Sydney PC User Group Newsletter, (Available from Box E162, St James, Sydney 2000).

Gaston, S.J. (1986). Controlling and auditing small computer systems. Toronto: CICA.

Gerrity, T.P. & Rockart, J.F. (1986). End-user computing: are you a leader or a laggard? Sloan Management Review, 27(4), 25-34.

Ghosal, M. & Caster, A. (1990). A disciplined approach to spreadsheet development. Business, 40(4), 39-44.

Gilb, T. (1977). Software metrics. Cambridge MA: Winthrop.

Godehardt, E. (1990). Graphs as structural models. The applications of graphs and multigraphs in cluster analysis (2nd ed.). Braunschweig: Vieweg.

Gordon, A.D. (1981). Classification - Methods for the exploratory analysis of multivariate data. London: Chapman & Hall.

Goss, E., Dillon, T. & Kendrick, J. (1989). Bittersweeet spreadsheets: Application development needs control. Woman CPA, 51(3), 20-24.

Grant, C., Colford, J., Daly, D. & Ingle, J. (1984). Managing microcomputers: A guide for financial policymakers (Report prepared by Price Waterhouse). New York: National Association of Accountants.

Guimares, T. & Ramanujam, V. (1986). Personal computing trends and problems: An empirical study. MIS Quarterly, 10, 179-187.

Halstead, M.H. (1977). Elements of software science. Amsterdam: Elsevier.

Hartigan J.A. (1985) Statistical theory in clustering. Journal of Classification, 2(1), 63-76.

Hassinen, K., Sajaniemi, J. & Väisänen, J. (1988, August). Structured spreadsheet calculation. 1988 IEEE Workshop on Languages for Automation. p. 129-133.

Hayen, R.L. & Peters, R.M. (1989). How to ensure spreadsheet integrity. Management Accounting, 70(10), 30-33.

Hoglund, D.D. (1984). Policies and issues concerning requirements to document the user input/output of micro spreadsheets, Unpublished Master's thesis, (Colorado State University, Fort Collins).

Howard, S. & Murray, D.M. (1987). An outline of techniques for evaluating the human computer interface. Proceedings of the 4th symposium on empirical methods of evaluation. New York: Plenum.

Howitt, D. (1985). Avoiding bottom-line disaster - Increased use of electronic worksheets heightens risk of serious errors. Infoworld, 7(6), 26-30.

Jackson, B. B. (1983). Multivariate data analysis. Homewood, IL: Irwin.

Jardine, N. & Sibson, R. (1971). Mathematical taxonomy. London: Wiley.

Karten, N. (1989, Summer). Disasters begin at home. Information Strategy, The Executive's Journal, p. 29-30.

Kasper, G.M. & Cerveny, R.P. (1985). Laboratory study of user characteristics and decision making performance in end-user computing. Information and Management, 9(2), 87-96

Kaufman, L. & Rousseeuw, P.J. (1990). Finding groups in data: An introduction to cluster analysis. New York: Wiley.

Kee, R. (1988). Pro...mming standards for spreadsheet software. CMA (Canada), 62(3), 55-60.

Kee, R. C. & Mason, J.O. jnr (1988). Preventing errors in spreadsheets. Internal Auditor (USA), 45(1), 42-47.

Kish, L. (1987). Statistical design for research. New York: John Wiley & son.

Krull. A.R. (1986, Winter). Management controls for personal computer . An internal auditor's overview. Computer Control Quarterly, p. 35-40.

Lakoff, G. (1987). Women, fire and dangerous things: What categories reveal about the mind. Chicago: University of Chicago Press.

Levine, M. & Siegal, J. (1987). Small business computer systems: Identifying the risk factor. National Public Accountant, 32(5) 38-43.

Licklider, T.R. (1989, December). Ten years of rows and columns. Byte, p. 324-331.

Long, T.J., Convey, J.J. & Chwalek, A.R. (1985). Completing dissertations in the behavioural sciences and education. San Francisco: Jossey-Bass.

Lorr, M. (1983). Cluster analysis for social scientists. San Francisco: Jossey-Bass.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. 5th Berkley Symposium on Mathematical Statistical Probability.

Macro, A. (1990). Software engineering concepts and management. New York: Prentice Hall.

McCabe, T. (1976). A complexity measure. IEEE Transactions on Software Engineering, SE-2(4), 308-320.

Mehrens, W.A. & Lehmann, I.J. (1978). Measurement and evaluation in education and psychology (2nd ed.). New York: Holt, Rinehart & Winston.

Meyer, M.M. & Curley, K.F. (1989). A methodology for classifying the complexity of expert systems: A pilot study. Proceedings of the 10th international conference on information systems (pp. 30-41). New York: ACM.

Mezzich, J.E. & Solomon, H. (1980). Taxonomy and behavioural science - Comparative performance of grouping methods. London: Academic Press.

Michalski, R. S. & Stepp, R. (1983a). Automated construction of classifications: Conceptual clustering versus numerical taxonomy. IEEE Transactions of Pattern Analysis & Machine Intelligence, PAMI-5(4), 396-410.

Michalski, R.S. & Stepp, R.A. (1983b). Learning from observation: Conceptual clustering, In R. Michalski, J Carbonell & T. Mitchell (Eds.) Machine Learning. 331-363. Palo Alto, CA: Tioga.

Miller, M. (1989, October). DOS spreadsheets - Consolidation and linking. PC World. p. 58.

Moskowitz, R. (1987a). Spreadsheets' quiet horror. Computerworld, 21(18), 35-40.

Moskowitz, R. (1987b). Unspoken nightmare: S nid spreadsheets. Software News (USA), 7(5), 51.

Nesbit, I.S. (1985). On thin ice: Micros and data integrity. Datamation, 31(21), 80-85.

Page-Jones, M. (1990, July/August). The one, two, three, four, five, six, seven stages of expertise in software engineering. American Programmer, p. 36-43.

Parker R.G. (1988). Microcomputer security and control. E D P Auditor Journal, 1, 13-20.

Paxton, W. (1991). Microcomputers in operating departments - Controlling the risks. Woman CPA, 53(1), 20-24.

Pearson, R. (1988, December). Lies, damned lies and spreadsheets. Byte, p. 299-304.

Perry, D.E. & Kaiser, G.E. (1991). Models of software development environments. Transactions on Software Engineering, 17(3), 283-295.

Powell, N.C. & Strickland, S.G. (1989, Autumn). Security in the microcomputer environment. Ohio CPA Journal, p. 20-23.

Rockart, J.F. & Flannery, L.S. (1983). The management of end user computing. Communications of the ACM, 26(10), 776-784.

Romesburg, H.C. (1984). Cluster analysis for researchers. Belmont, CA: Lifetime Learning.

Ronen, B., Palley, M.A. & Lucas, H.C. Jr. (1989). Spreadsheet analysis and design. Communications of the ACM, 32(1), 84-93.

Sato, O. (1989). Controlling end user computing. An analytical framework. SIGSAC Review, 7(3), 6-12.

Schiffman, S.S., Reynolds, M.L., & Young, F.W. (1981). Introduction to multidimensional scaling. New York: Academic Press.

Schmitt, T. (1988). Recognising and controlling the development of distributed programming. EDPACS, the EDP audit, control and security newsletter, 16(4), p. 1-20.

Schneider, P. & Hines, M.L.A. (1990). Classification of medical software. In Proceedings of 1990 symposium on applied computing (IEEE). p. 20-27.

Schultz, N.O. & Hoglund D.D. (1986). Microcomputer spreadsheets. A case for controls. Internal Auditor (USA), 43(1), 46-50.

Shneiderman, B. (1980). Software psychology. Human factors in computer and information systems. Boston: Little, Brown.

Shneiderman, B. (1987). Designing the user interface - Strategies for effective human-computer interaction. Reading MA: Addison Wesley.

Simkin, M.G. (1987). Micros in accounting: How to validate spreadsheets. Journal of Accountancy, 164(5), 130-138.

Sneath, P.H.A. & Sokal, R.R. (1973). Numerical taxonomy: The principles and practice of numerical classification. San Francisco: W H Freeman.

Sokal, R.R. & Sneath, P.H.A. (1963). Principles of numerical taxonomy. San Francisco: W.H. Freeman.

Sommerville, I. (1985). Software engineering (2nd ed.). London: Addison Wesley.

Spencer, C. (1986). Weeding out worksheet errors. Personal Computing, 10(11), 160-162.

Steenbergen, E. (1989, June). Personal computing versus software quality? Offline (Editorial, reprinted from the South Australian branch news of the Australian Computer Society "Leading Edge", available from West Australian Branch, Australian Computer Society, GPO Box F320 Perth, W.A. 6001).

Steinaker, N.W. & Bell, M.R. (1979). The experiential taxonomy: A new approach to teaching and learning. New York: Academic Press.

Stewart, W. & Flanagan, J. (1987, December). Spreadsheet design - Some simple principles. Australian Accountant, p. 56-59.

Stone, D.N. & Black, R.L. (1989). Using microcomputers: Building structured spreadsheets. Journal of Accountancy, 168(4), 131-142.

Stopher, P.R. & Meyburg, A.H. (1979). Survey sampling and multivariate analysis for social scientists and engineers. Lexington MA: D.C. Heath.

Tabachnick, B.D. & Fidell, L.S. (1989). Using multivariate statistics (2nd ed.). London: Harper & Row.

Thompson, B. & B. (1991) .Overturning the category bucket. - Creating conceptual clusters. Byte 16(1), p. 249, 256.

Troy, R. & Moawad, R. (1982). Assessment of software reliability models. In Proceedings of COMPSAC 82, the 6th annual computer software & applications conference. New York: IEEE.

Tucker, S. (1987).Design worksheets that communicate. Lotus, 3(9), 74-78.

Udell, J. (1990, April). OS/2 2.0: It's a family affair. Byte, p. 119-123.

Ware, E. (1986, September). Effective spreadsheet design. Australian Computing, p. 63-65.

Weber, R. (1986). Planning and control issues in end user computing. Australian Computer Journal, 18(4), 159-165.

West, M.G . (Ed) & Lipp, M.E. (1986). A taxonomy of prototyping - Tools and methods for database, decision support and transaction systems. Prototyping, state of the art report, p. 105-121. New York: Pergamon Infotech.

Wilkinson, L. (1990). SYSTAT: The system for statistics. Evanston IL: SYSTAT Inc.

Williams, T. (1989). Hidden danger in spreadsheet balancing act. Computer Control Quarterly, 7(4), 45-47.

Wright, R. (1990, September). Downunder. PC User, p. 102-108.

Yager, T. (1990, October). What's next after 123? Byte, p. 147-149.

Zwicky, F. & Wilson, A.G. (Eds.) (1967). The morphological approach to discovery. in New methods of thought and procedure (pp. 273-330). New York: Springer Verlag.

APPENDIX A

DATA COLLECTION INSTRUMENTS

EDITH COWAN
UNIVERSITY
PERTH WESTERN AUSTRALIA
BUNBURY CAMPUS

Robertson Drive, Bunbury
Western Australia 6230
Telephone (097) 910 222
Facsimile (097) 216 994

21st September, 1991

Spreadsheet Applications Survey.

A research project funded by Edith Cowan University in Western Australia.

Spreadsheet applications are developed in many sites all over Australia. Some are subjected to rigid design and implementation controls and others are developed in a free and easy 'ad hoc' manner. Some are the basis for major decision making. Others handle purely private information of little significance to anyone other than the developer. The developers are just as varied in terms of employment, qualifications and spreadsheet experience.

Some spreadsheet applications have rigorous controls and checks and balances built in, whilst others have little or none. Some obviously require rigid control. In other cases controls seem entirely inappropriate and a waste of time and effort to implement and enforce.

What types of spreadsheets are being developed? Who uses them? For what purpose? And what about controls. How many are used? In what kind of Spreadsheets? What types of controls are appropriate? How does a developer decide?

This project seeks to provide some answers. It will show what types of application are being developed locally and the degree of standardisation and control they contain. Your opinion as a spreadsheet developer is sought. Is there any need to include particular design and control measures in your application? Of course, there are no overall correct answers. Each situation is different.

As a spreadsheet developer you will be interested in furthering our knowledge in this area to give guidance to developers in the identification and implementation of relevant controls when their application really requires these.

A questionnaire is enclosed. Would you please complete it referencing any spreadsheet application or template (small or large) which you have developed and with which you are familiar. You will need computer access to determine aspects such as spreadsheet size and storage. The survey form should take about twenty to thirty minutes to complete. Would you please return it within two weeks in the reply paid envelope enclosed. Extra forms are readily available on request.

Thank you for agreeing to help in this project. The donation of your valuable time is appreciated and will help provide some answers leading to a better understanding of spreadsheet applications and their control requirements. Just a little of your time will eventually be of benefit to many other spreadsheet developers and I hope you will pick up a few new ideas from this survey that you can put into good use.

Yours sincerely,

Jean Hall
Researcher
Department of Computer Studies

EDITH COWAN UNIVERSITY
WESTERN AUSTRALIA
BUNBURY CAMPUS

SPREADSHEET SURVEY

This survey is in three parts. Please answer all the questions with regard to a spreadsheet application or template that **YOU** have developed and are familiar with. You will need to have computer access to the spreadsheet to answer part 2. The survey should take about 30 minutes to complete.

Place a cross in one and only one answer box for each question.

24 Does your template display the run date?

[X] Yes [ ] No —> question 26

25 In which format is the run date displayed?

[ ] DD/MM/YY
[ ] YY/MM/DD
[ ] DD MMM YY
[X] Other

Please specify...... November 21st, 1991

26 Does your template include the author's name?

[ ] Yes [X] No

Please return this survey in the reply-paid pre-addressed envelope provided. For further information contact:

Mrs Jean Hall, Lecturer in Computer Studies
Edith Cowan University, Bunbury Campus
Robertson Drive, Bunbury W.A. 6230.
Telephone (097) 910222

Environmentally friendly: Printed on Australian made 100% Recycled paper.

1 What is the prime use of this spreadsheet?

- ☐ Communication / Explanation
- ☐ Report generation
- ☐ Classification
- ☐ "What if" analysis
- ☐ Optimisation
- ☐ Prediction / Forecasting
- ☐ Other.

Specify ....

2 In which sector is it used?

- ☐ Public (Government)
- ☐ Private
- ☐ Recreation / Personal

3 In which industry is the spreadsheet used?

- ☐ Agriculture / Forestry / Fishing
- ☐ Mining / Refining
- ☐ Manufacturing
- ☐ Electricity / Gas / Water
- ☐ Construction / Engineering
- ☐ Wholesale / Retail
- ☐ Finance / Banking
- ☐ Business
- ☐ Public administration
- ☐ Education
- ☐ Computing
- ☐ Other

Specify:..

4 How large is the organisation where this spreadsheet is used?

- ☐ Single person
- ☐ Single Department
- ☐ Many Departments - One site
- ☐ Many departments - Many sites
- ☐ Multinational

How important is this spreadsheet to the user organisation?

Consider the value of decisions made using this spreadsheet. Also consider the ramifications to your organisation if the spreadsheet were to contain errors or be withdrawn.

- ☐ Unimportant
- ☐ Moderate importance
- ☐ Major importance

6 Did you have enough time available to develop this spreadsheet?

☐ Yes ☐ No - a rush job

7 Are you aware of a spreadsheet development policy within the user organisation for whom you developed this spreadsheet?

☐ Yes ☐ No ---> question 10

8 Did you have a documented copy of this policy when you developed the spreadsheet?

☐ Yes ☐ No

9 How is this policy enforced?

- ☐ Guidelines only - not enforced
- ☐ Departmental responsibility
- ☐ D.P. Departmental responsibility
- ☐ Internal Auditor
- ☐ Other

Specify..

10 Does the user organisation keep a library of sample templates and quality spreadsheets for distribution?

☐ Yes ☐ No

THE USER ORGANISATION

Please state your name and a contact address and telephone number. This information will not be processed with the data nor published. It will be used by the researcher solely for the purpose of contacting you if necessary.

Name:

Address:

Telephone:

11 Are you a member of a spreadsheet user group?

- [ ] Yes
- [ ] No

12 Gender?

- [ ] Male
- [ ] Female

13 Age?

- [ ] <25
- [ ] 25-34
- [ ] 35-44
- [ ] >45

14 Spreadsheet Development Experience?

- [ ] Novice
- [ ] Knowledgeable
- [ ] Power User

15 Training received in Spreadsheet Development. **Cross one box only.**

- [ ] D.P. Professional
- [ ] D.P. Amateur trained by courses
- [ ] D.P. Amateur trained by work-mates
- [ ] D.P. Amateur largely self taught

16 How many books, newspapers or magazine articles about spreadsheets do you read?

- [ ] <3/yr
- [ ] 3-8/yr
- [ ] > 8/yr

17 Highest level of qualification?

- [ ] School
- [ ] Trade
- [ ] Diploma
- [ ] Degree
- [ ] Postgraduate

18 Do you hold a membership of a professional body? e.g. C.P.A. , M.A.C.S.

- [ ] No
- [ ] Yes

Specify..

19 Your occupation when developing this spreadsheet?

- [ ] Manager / Administrator
- [ ] Scientist / Engineer
- [ ] Academic / Teacher
- [ ] Accountant / Finance
- [ ] Data Processing Professional
- [ ] Tradesperson
- [ ] Clerk
- [ ] Other

Specify..

20 Your employment status when developing this spreadsheet?

- [ ] Consultant
- [ ] Executive
- [ ] Section / Department Manager
- [ ] Employee
- [ ] Self Employed
- [ ] Unpaid Helper
- [ ] Other

Specify..

YOU, THE SPREADSHEET DEVELOPER

EDITH COWAN UNIVERSITY

SPREADSHEET SURVEY

PART 3 Design and Control Issues

The following questions seek your opinion as a developer on including various design and control measures in your spreadsheet. Different spreadsheets require a different selection of these measures. There are no universally correct answers.

We wish to find out which control methods you think are worthwhile for your type of spreadsheet. Reply for your particular template not spreadsheets in general.

The questions are in two parts:

1) Did you use a particular design feature in your spreadsheet?

2) How useful could the same design feature be in your spreadsheet in your particular circumstances?

A 'no' reply to the first part of the question, does not prevent you from picking 'essential' or 'useful' for the second answer.

				Your opinion			
		Yes	No	Essential	Useful	Unnecessary	Undecided
12	Do you carry a spare fan belt in your motor car?	☒	☐	☐	☒	☐	☐
13	Did you wear a seat-belt last time you travelled in a motor car?	☐	☒	☒	☐	☐	☐
14	Do you normally check your car tyre pressure weekly?	☐	☒	☐	☒	☐	☐

It is important to answer these questions with regard to your spreadsheet and circumstances not spreadsheet applications in general.

PART 2. PLEASE CHECK YOUR SPREADSHEET

Please state the name of your spreadsheet application (template) and any associated files. This information will not be published. It will be used by the researcher solely for the purpose of identification if further communication with you is necessary.

....

21 Spreadsheet Software used?

....

Version? ....

22 State any add on programs used eg Auditing, note taking, text enhancement

23 Operating system used?

....

24 Main template file storage size ?

....Bytes

Spreadsheet dimensions?

25 No. of Rows ....

26 No. of Columns ....

27 ☐ 3D ☐ 2D ---> question 29

28 No. of worksheets in 3D? ....

Please examine your spreadsheet and estimate the percentage of cells occupied by each type of content:

	<20%	20-40%	40-60%	60-80%	>80%
29 Constant / Lookup field	☐	☐	☐	☐	☐
30 Data entry at runtime	☐	☐	☐	☐	☐
31 Formula	☐	☐	☐	☐	☐
32 Label	☐	☐	☐	☐	☐
33 Blank cell	☐	☐	☐	☐	☐
34 Other (macros etc)	☐	☐	☐	☐	☐

35 Does this spreadsheet use both absolute and relative cell referencing?

☐ Yes ☐ No

36 Does this spreadsheet have split screens?

☐ Yes ☐ No

37 Does this spreadsheet have frozen horizontal and / or vertical borders?

☐ Yes ☐ No

38 Does this spreadsheet have links for data transfer to or from other spreadsheets?

☐ Yes ☐ No

39 Does this spreadsheet have links for data transfer to or from its own or an external database?

☐ Yes ☐ No

40 Does this spreadsheet use Windows 3 D.D.E. (Dynamic Data Exchange)?

☐ Yes ☐ No

41 Does this spreadsheet use graphics?

☐ Yes ☐ No ---> question 43

42 How sophisticated are the graphics?

☐ Simple e.g. pie or bar

☐ Intermediate e.g. XY

☐ Complex e.g. 3D, contour

43 Does this spreadsheet use macros?

☐ Yes ☐ No ---> next page

44 How complex are the macros?

☐ Simple

☐ Significant

☐ Extensive or Complex

45 Is the spreadsheet design modular?

☐ Yes ☐ No --->question 47

46 Module arrangement

☐ Diagonal e.g.
☐ Blocked e.g

47 Does the spreadsheet include 'LOOKUP' table functions?

☐ Yes ☐ No

48 Does it include logical 'IF' functions?

☐ Yes ☐ No ----> question 50

49 Does the spreadsheet include nested 'IF' functions?

☐ Yes ☐ No

50 How complex are the spreadsheet's formulas?

☐ Simple
☐ Average
☐ Complex

51 Who runs this spreadsheet?

☐ self only
☐ two or three others
☐ many users

52 Who enters the data?

☐ Self only
☐ Data entry clerk who does not use the spreadsheet output
☐ Those who use the output.

53 Does this spreadsheet contain only private data used by yourself?

☐ Yes ☐ No

54 How far is the immediate output of the spreadsheet run distributed?

☐ Self only
☐ Single department
☐ Multi department
☐ Beyond the user organisation

55 How often is the spreadsheet run?

☐ One shot model
☐ Just a few times
☐ Daily
☐ Weekly
☐ Monthly
☐ Occasionally after long intervals e.g. end of financial year.
☐ Frequently, whenever needed

56 Does this spreadsheet input corporate data ? i.e. data that belongs to the whole organisation not just to the template user?

☐ Yes ☐ No ---> question 59

57 Where does the corporate data come from?

☐ electronic transfer
☐ keyed in from reports
☐ Other

Specify...

58 Does this spreadsheet modify the corporate data before output?

☐ Yes ☐ No

59 Does this spreadsheet create new corporate data?

☐ Yes ☐ No

60 For how long is the spreadsheet output used?

☐ < 1 week
☐ 1 week to a month
☐ > 1 month

	DESIGN	Yes	No	Your opinion: Essential	Useful	Unnecessary	Undecided
61	Did you plan this spreadsheet on paper before implementing it with a software package?	☐	☐	☐	☐	☐	☐
62	Does the spreadsheet have a separate entry area where data is input at run time?	☐	☐	☐	☐	☐	☐
63	Does the spreadsheet have a separate area for storing seldom changed parameters and constants?	☐	☐	☐	☐	☐	☐
64	Does the spreadsheet have a separate area for storing look-up tables?	☐	☐	☐	☐	☐	☐
65	Does the spreadsheet have a separate area for storing macros?	☐	☐	☐	☐	☐	☐
66	Does the spreadsheet have separate areas for output reports?	☐	☐	☐	☐	☐	☐
67	Does the spreadsheet have separate calculation or work areas?	☐	☐	☐	☐	☐	☐
68	Does the spreadsheet have a header module containing author details?	☐	☐	☐	☐	☐	☐
69	Does the spreadsheet have a header module or 'help' macro giving instructions for use?	☐	☐	☐	☐	☐	☐
70	Does the spreadsheet have a separate on-line area where assumptions and /or known limits to the model's validity are described?	☐	☐	☐	☐	☐	☐
71	Does the spreadsheet have a separate on-line area where details of changes to the template such as date revised and revisions made are recorded?	☐	☐	☐	☐	☐	☐
72	Is an on-line record kept of the file-names of previous versions of this spreadsheet?	☐	☐	☐	☐	☐	☐

	FORMULAS	Yes	No	Your opinion: Essential	Useful	Unnecessary	Undecided
73	Did you use paramaterised constants in formulas? i.e. use a reference to the cell where the constant is stored rather than the numerical value of the constant.	☐	☐	☐	☐	☐	☐

SPREADSHEET CONTROLS USED

		Yes	No	Essential	Useful	Unnecessary	Undecided
74	Did you point out formulas rather than type in cell addresses?	☐	☐	☐	☐	☐	☐
75	Did you use range names?	☐	☐	☐	☐	☐	☐
76	When specifying a range addition e.g. with the SUM function did you also include a blank row above and / or below the range to be summed?	☐	☐	☐	☐	☐	☐
77	Did you ensure that no formulas are stored on the same screen as cells requiring input?	☐	☐	☐	☐	☐	☐
78	Did you turn on cell protection on cells containing formulas?	☐	☐	☐	☐	☐	☐
79	Did you consider rounding errors when implementing your formulas?	☐	☐	☐	☐	☐	☐
80	Does your spreadsheet have check totals reconciling in two directions (cross footing)?	☐	☐	☐	☐	☐	☐

	INPUT CONTROLS			*Your opinion:*			
		Yes	No	Essential	Useful	Unnecessary	Undecided
81	Do your spreadsheet's data entry screen areas resemble a paper form familiar to the person responsible for data entry?	☐	☐	☐	☐	☐	☐
82	Do your data entry screens have cells requiring data entry arranged in rows or columns permitting data entry in one direction only?	☐	☐	☐	☐	☐	☐
83	Are cells requiring data entry differentiated from other cells? e.g. by colour or highlighting?	☐	☐	☐	☐	☐	☐
84	Did you build in range and / or reasonableness checks on input data cells?	☐	☐	☐	☐	☐	☐
85	Does your spreadsheet use batch totals to check numeric data input? i.e. the spreadsheet electronically totals data entered. This is compared with a batch total obtained by summing the data from the input documents.	☐	☐	☐	☐	☐	☐

	OUTPUT CONTROLS	Yes	No	*Your opinion:* Essential	Useful	Unnecessary	Undecided
86	Does this spreadsheet have built in range and / or reasonableness checks on output cells?						
87	Does each printout or output screen include the date it was produced?						
88	Does each printout or output screen include the name of the spreadsheet?						
89	Is each printout signed before distribution?						
90	Is a record kept of who received copies from each run?						

	REVIEW AND TESTING	Yes	No	*Your opinion:* Essential	Useful	Unnecessary	Undecided
91	Does this spreadsheet comply with the user organisation's policy on design and documentation?						
92	Was this spreadsheet checked with the data entry person to ensure that they understand what to do?						
93	Have you printed out the formulas used, to check them by eye?						
94	Have you checked your formulas using test data?						
95	Did you work out in advance, manually or with a calculator, the test's expected results?						
96	Did you use test data for normal and predictable answers?						
97	Did you use test data with errors included?						
98	Did you use test data that was at the limits of normal range?						
99	Did you document and keep both the test's expected and actual results?						
100	Have you checked this spreadsheet with a separate auditing package or built in spreadsheet auditing functions?						

		Yes	No	Essential	Useful	Unnecessary	Undecided
101	Has another spreadsheet developer checked this template?	☐	☐	☐	☐	☐	☐
102	Has an internal auditor checked this spreadsheet?	☐	☐	☐	☐	☐	☐
103	Has an external auditor checked this spreadsheet?	☐	☐	☐	☐	☐	☐
104	Was there a formal procedure of sign off before the spreadsheet was put into use?	☐	☐	☐	☐	☐	☐

	HARDCOPY (PAPER) DOCUMENTATION			*Your opinion:*			
		Yes	No	Essential	Useful	Unnecessary	Undecided
105	Are the author details documented?	☐	☐	☐	☐	☐	☐
106	Is the design layout documented?	☐	☐	☐	☐	☐	☐
107	Is a printout kept of all formulas used?	☐	☐	☐	☐	☐	☐
108	Are any associated macros documented?	☐	☐	☐	☐	☐	☐
109	Are assumptions made and/or known limits to the spreadsheet's validity documented?	☐	☐	☐	☐	☐	☐
110	Are instructions for spreadsheet use included in the documentation?	☐	☐	☐	☐	☐	☐
111	Is there a written record of spreadsheet versions detailing changes made to the original template?	☐	☐	☐	☐	☐	☐

	SECURITY			*Your opinion:*			
		Yes	No	Essential	Useful	Unnecessary	Undecided
112	Is a backup copy of this spreadsheet kept in the same office as the computer?	☐	☐	☐	☐	☐	☐
113	Is a backup copy of this spreadsheet kept in another location?	☐	☐	☐	☐	☐	☐
114	Are normal access and distribution lists kept for this spreadsheet?	☐	☐	☐	☐	☐	☐
115	Has this spreadsheet been compiled to prevent unauthorised alteration?	☐	☐	☐	☐	☐	☐

THANK YOU FOR COMPLETING THE SURVEY

THE A.D.E. TAXONOMY OF SPREADSHEET APPLICATION DEVELOPMENT

This taxonomy has been developed at the Edith Cowan University to categorise spreadsheet development projects. Each spreadsheet development project can be categorised in three parts concerning:

- The ***APPLICATION*** that was developed
- The ***DEVELOPER*** who created the spreadsheet template or application
- The ***ENVIRONMENT*** in which the spreadsheet was developed

A key for each of these three parts is included. A complete categorisation of a spreadsheet would involve three codes (e.g. M3, C1, U3), the first for the Application, a second for the Developer and the third for the Environment - the A.D.E. taxonomy.

Please choose any spreadsheet application or template that you have developed and select the three codes. Then complete the form below. The spreadsheet chosen can be large or small, simple or complex, important or not. Your help is appreciated.

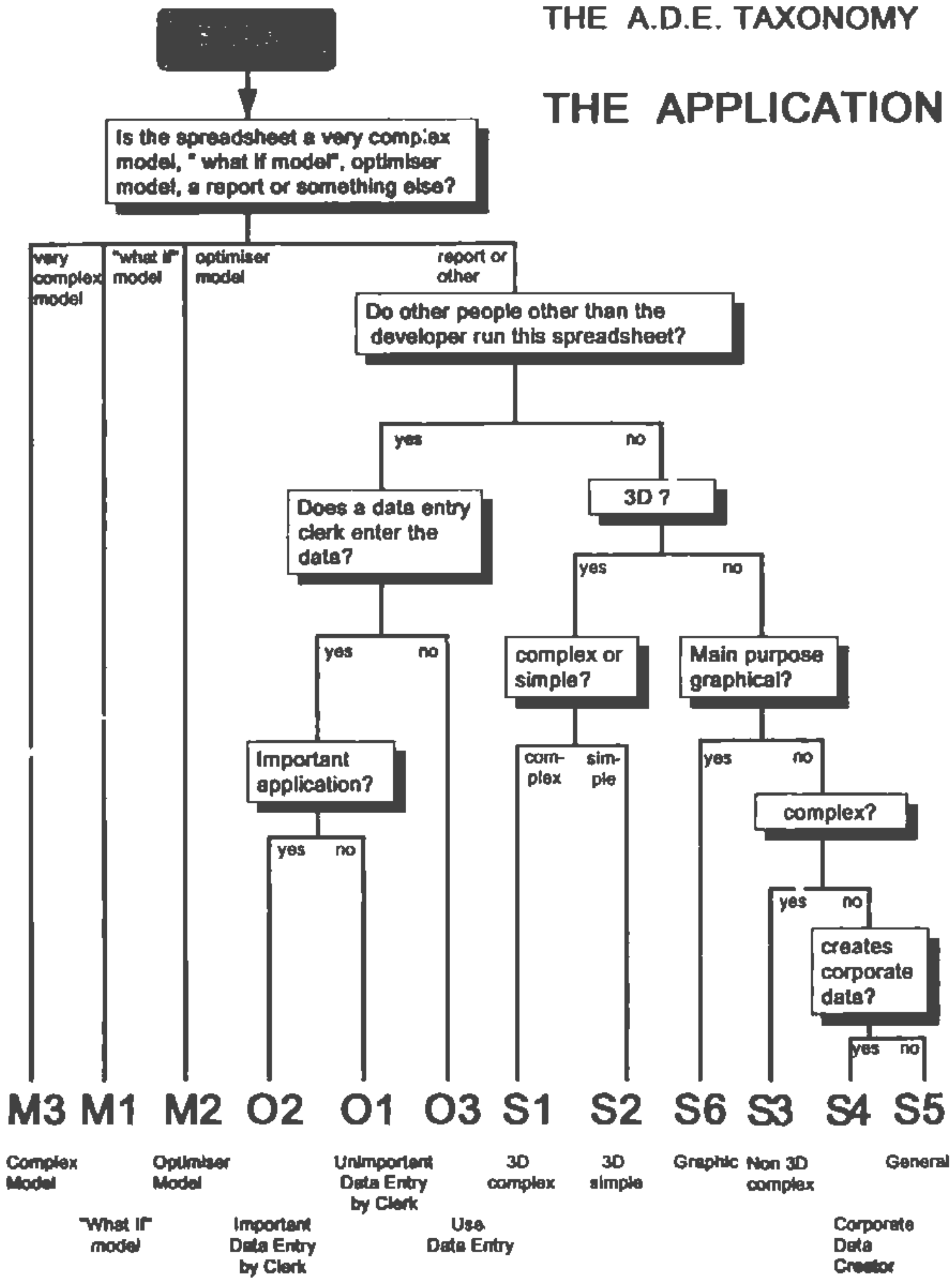
Your Name		**Telephone Contact**	
Spreadsheet Application Name		**Today's Date**	

Application code	**Developer Code**	**Environment Code**
A	**D**	**E**

Please comment on any difficulties you had coding your spreadsheet.

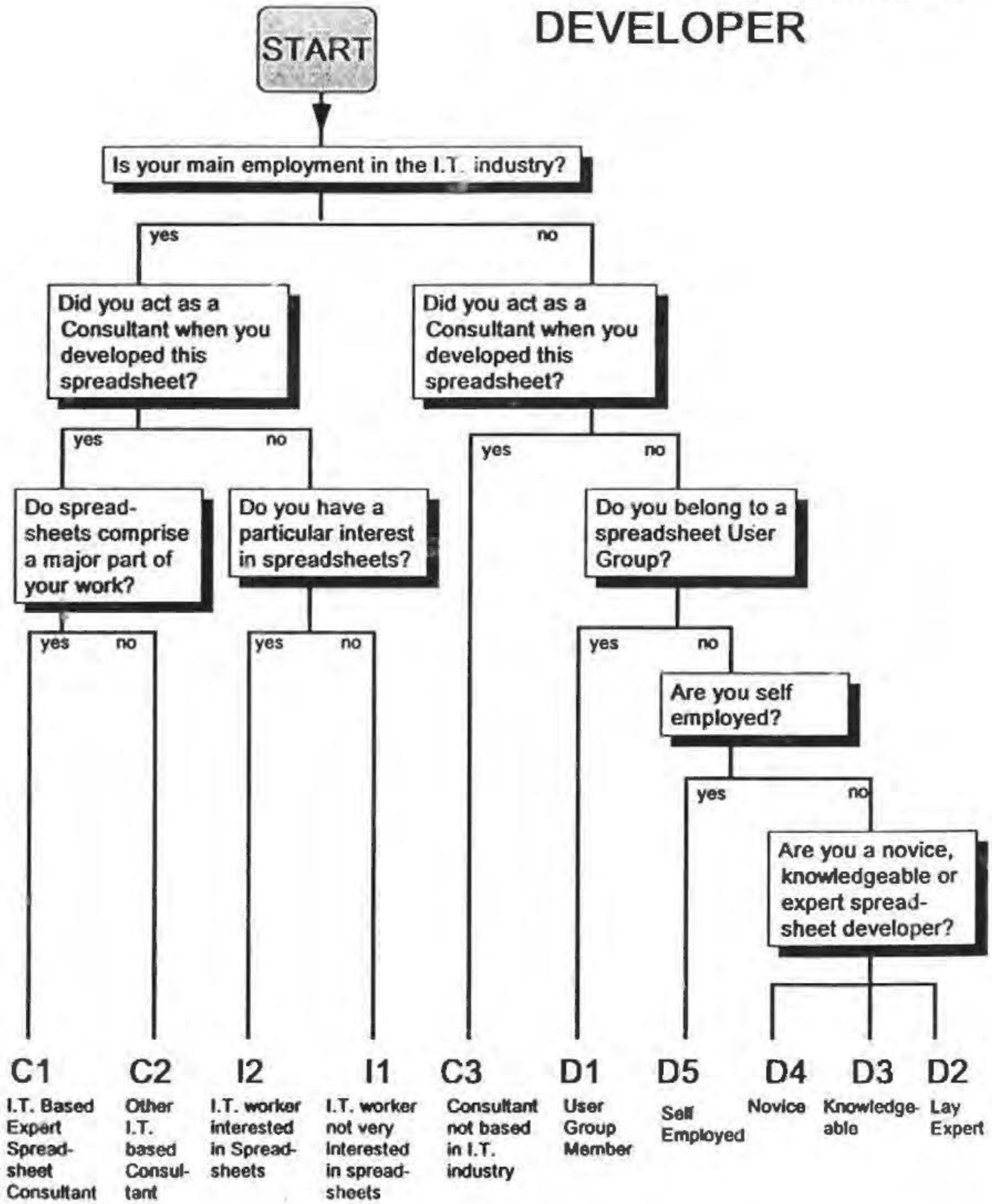
THE A.D.E. TAXONOMY

THE APPLICATION



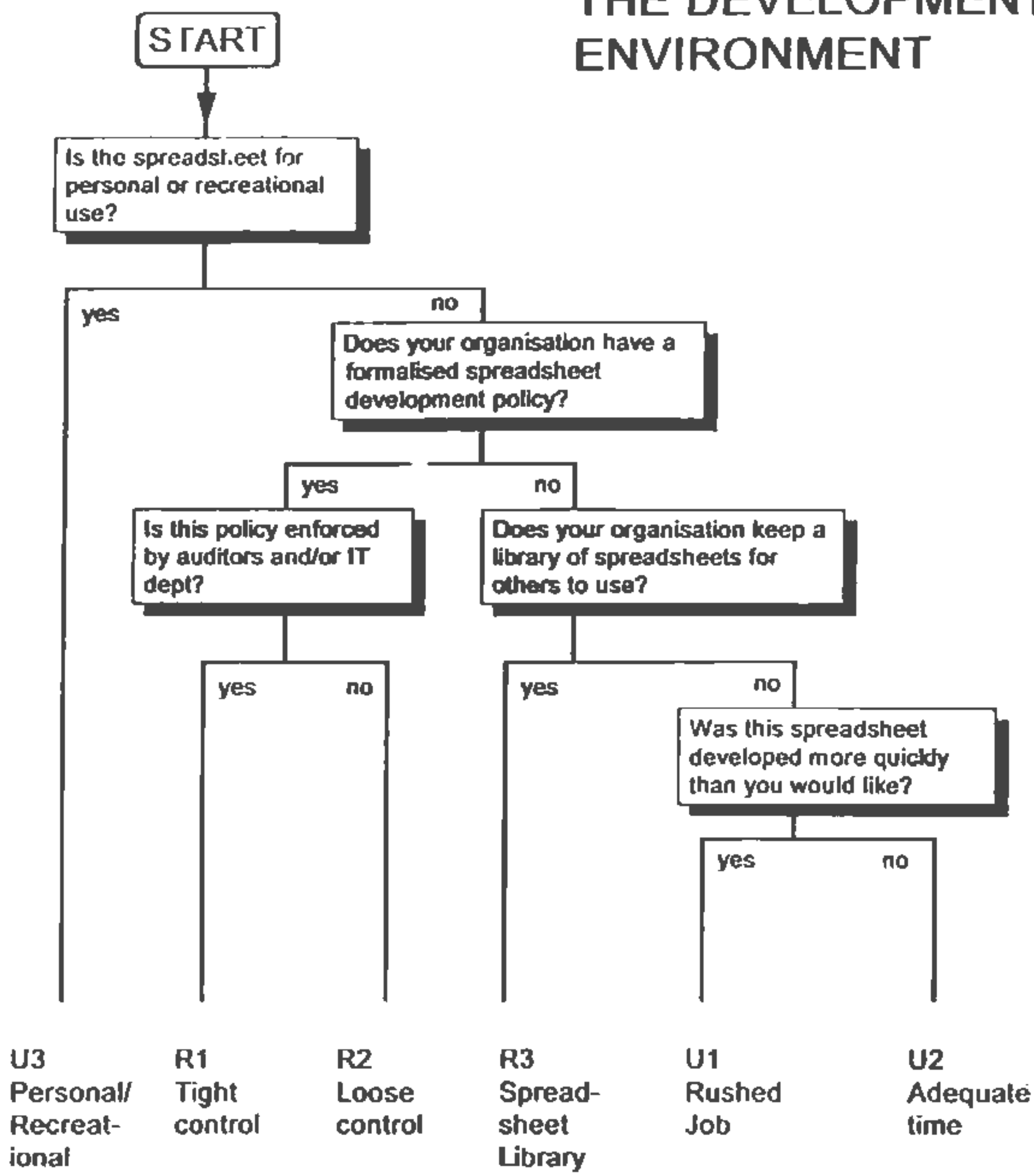
THE A.D.E. TAXONOMY

THE SPREADSHEET DEVELOPER



THE A.D.E. TAXONOMY

THE DEVELOPMENT ENVIRONMENT



APPENDIX B

VARIABLES & CODE BOOKS

Table 22 Survey Code Book: Fields for SURVEY Database

Question	Topic	DBMS field	Code	Meaning
	Identifier	LABEL$	numeric	Unique identifier
1	Spreadsheet use	PURPOSE	1	Comm/ Explain
			2	Report
			3	Classification
			4	"What if"
			5	Optimise
			6	Predict/ Forecast
			7	Other
2	Sector	SECTOR	1	Public
			2	Private
			3	Rec/Personal
3	Industry	INDUSTRY	1	Ag/ Forest/ Fish
			2	Mining/Refinery
			3	Manufacturing
			4	Elec/ Gas/ Water
			5	Construct/ Eng.
			6	Wholesale/ Retail
			7	Finance/ Banking
			8	Business
			9	Public Admin
			10	Education
			11	Computing
			12	Other
4	Organisation size	ORGSIZE	1	Single person
			2	Single dept
			3	Depts one site
			4	Depts many sites
			5	Multinational

Question	Topic	DBMS field	Code	Meaning
5	Spreadsheet importance	IMPORTAN	1	Unimportant
			2	Moderate imp
			3	Major imp.
6	Sufficient Development time available	ENUFTIME	0	No
			1	Yes
7	Organisational Spreadsheet Policy	SDPOLICY	0	No
			1	Yes
8	Documented Policy	SDDOCO	0	No
			1	Yes
9	Policy Enforcement	SDENFORC	1	Guidelines only
			2	Deoartmental
			3	DP Department
			4	Internal Auditor
			5	Other
10	Spreadsheet Library	LIBRARY	0	No
			1	Yes
11	User Group membership	USERGRP	0	No
			1	Yes
12	Gender	GENDER	0	Female
			1	Male
13	Age	AGE	1	<25
			2	25-34
			3	35-44
			4	>45
14	Spreadsheet experience	EXPERT	1	Novice
			2	Knowledgeable
			3	Power User
15	Training	TRAINING	1	Professional
			2	Courses
			3	Work-mates
			4	Self-taught

Question	Topic	DBMS field	Code	Meaning
16	Spreadsheet reading	READ	1	< 3/yr
			2	3-8/yr
			3	>8/yr
17	Highest qualification	QUALIFY	1	School
			2	Trade
			3	Diploma
			4	Degree
			5	Postgraduate
18	Professional Membership	PROFMEMB	0	No
			1	Yes
		PROFBDY$	alpha	
19	Occupation	JOB	1	Manager/ Admin
			2	Science/ Engineer
			3	Academic/ Teacher
			4	Accountant/Finance
			5	DP Professional
			6	Trade
			7	Clerk
			8	Other
20	Employment status	STATUS	1	Consultant
			2	Executive
			3	Section Manager
			4	Employee
			5	Self Employed
			6	Unpaid Helper
			7	Other
21	Spreadsheet Software Used	PROGRAM$	alpha	
		VERSION$	alpha	
22	Add on Programs	ADDONS$	alpha	
23	Operating System	OS$	alpha	
24	Size in bytes	SIZE	numeric	
25	No. of rows	ROWS	numeric	
26	No. of columns	COLUMNS	numeric	

Question	Topic	DBMS field	Code	Meaning
27	Dimensions	DIMENSIO	0	2D
			1	3D
28	no. of worksheets	WSHEETS	1	2D
			num>1	3D no. of sheets
29	% cells - Constant / Lookup	CELLCONS	1	<20%
			2	20-40%
			3	40-60%
			4	60-89%
			5	>80%
30	% cells - Data entered	CELLDATA	1-5	as above
31	% cells - Formulas	CELLFORM	1-5	as above
32	% cells - Labels	CELLLABL	1-5	as above
33	% cells - Blank	CELLBLNK	1-5	as above
34	% cells - Other	CELLOTHR	1-5	as above
35	Absolute / relative referencing	ABSREL	0	No
			1	Yes
36	Split Screens	SPLITSCRN	0	No
			1	Yes
37	Borders	BORDERS	0	No
			1	Yes
38	Links to spreadsheets	LINKSS	0	No
			1	Yes
39	Links to data bases	LINKDB	0	No
			1	Yes
40	Links to Windows DDE	LINKDDE	0	No
			1	Yes
41	Graphics	GRAHICS	0	No
			1	Yes

Question	Topic	DBMS field	Code	Meaning
42	Graphic sophistication	GRAPHSOP	1	Simple
			2	Intermediate
			3	Complex
43	Macros	MACROS	0	No
			1	Yes
44	Macro complexity	MACROCOM	1	Simple
			2	Significant
			3	Extensive/Complex
45	Modular Design	MODULAR	0	No
			1	Yes
46	Module arrangement	MODARRAN	0	Diagonal
			1	Blocked
47	LOOKUP functions	LOOKUPS	0	No
			1	Yes
48	"IF" functions	IFS	0	No
			1	Yes
49	Nested "IF" functions	NESTEDIF	0	No
			1	Yes
50	Formulas	FORMCOMP	1	Simple
			2	Average
			3	Complex
51	Spreadsheet run by	RUNBY	1	Self only
			2	2 or 3 others
			3	Many users
52	Data entered by	ENTERER	1	Self only
			2	Clerk
			3	Users
53	Private data only	PRIVATE	0	No
			1	Yes

Question	Topic	DBMS field	Code	Meaning
54	Spreadsheet distribution	OUTSCOPE	1	Self
			2	Single dept.
			3	Multi dept.
			4	Ex organisation
55	Spreadsheet run schedule	HOWOFTEN	1	One shot model
			2	Few times
			3	Daily
			4	Weekly
			5	Monthly
			6	Occasionally
			7	Frequently
56	Corporate data input	CORPDATA	0	No
			1	Yes
57	Source of corporate data	WHEREFRM	1	Electronic transfer
			2	Keyed in ex reports
			3	Other
58	Modifies corporate data	CDCHNG	0	No
			1	Yes
59	Creates corporate data	CDMODIFY	0	No
			1	Yes
60	Output retention	KEPT	1	< 1 week
			2	1 - 4 weeks
			3	> 4 weeks
	Postcode	POSTCODE$	alpha	Identifies stratum

Table 23 **Survey Code Book: Fields for CONTROLS Database**

Question	Topic	DBMS Field	Code	Meaning
61a	Design and Control issues	Q61A	0 1	Yes No
61b	Designers Opinion	Q61B	1 2 3 4	Essential Useful Unnecessary Undecided
62-115a	As for 61a	Q62A-Q115A	0-1	As above
62-115b	As for 61B	Q62B-Q115B	1-4	As above

Table 24: Variables used to develop the Taxonomy.

Variable included in dataset: RD - raw data, BD - binary dichotomous data, OD - ordinal data

Variable	Scale	Source	Topic	Code	Meaning	RD	BD	OD
LABEL$	nominal	derived	unique key	1-105		Y	Y	Y
PURPOSE	nominal	question 1	spreadsheet use	1-6		N	N	N
PCOMMS	bd	PURPOSE	communication	0, 1	no, yes	Y	Y	N
PREPORT	bd	PURPOSE	report	0, 1	no, yes	Y	Y	N
PCLASS	bd	PURPOSE	classification	0, 1	no, yes	Y	Y	N
PWHATIF	bd	PURPOSE	"What if"	0, 1	no, yes	Y	Y	Y
POPTIM	bd	PURPOSE	optimisation	0, 1	no, yes	Y	Y	Y
PFORCST	bd	PURPOSE	prediction / forecast	0, 1	no, yes	Y	Y	Y
PREST	bd	PCOMMS, PREPORT, PCLASS	non model	0, 1	no, yes	N	N	Y
SECTOR	nominal	question 2	sector	1-3		N	N	N
SPUBLIC	bd	SECTOR	public	0, 1	no, yes	Y	Y	Y
SPRIVT	bd	SECTOR	,private	0, 1	no, yes	Y	Y	Y
SPERSN	bd	SECTOR	personal	0, 1	no, yes	Y	Y	Y
INDUSTRY	nominal	question 3	Industry	1-12		N	N	N
IAG	bd	INDUSTRY	agriculture/ forestry	0, 1	no, yes	Y	Y	N
IMINE	bd	INDUSTRY	mining	0, 1	no, yes	Y	Y	N
IMANUF	bd	INDUSTRY	manufacturing	0, 1	no, yes	Y	Y	N
IELECT	bd	INDUSTRY	electricity /gas/ water	0, 1	no, yes	Y	Y	N
ICONST	bd	INDUSTRY	construction/ engineer	0, 1	no, yes	Y	Y	N
ISELL	bd	INDUSTRY	wholesale/ retail	0, 1	no, yes	Y	Y	N
IFINCE	bd	INDUSTRY	finance/ banking	0, 1	no, yes	Y	Y	N
IBUSNS	bd	INDUSTRY	business	0, 1	no, yes	Y	Y	N
IPUBAD	bd	INDUSTRY	public administration	0, 1	no, yes	Y	Y	N
IEDUC	bd	INDUSTRY	education	0, 1	no, yes	Y	Y	N
ICOMP	bd	INDUSTRY	I.T.	0, 1	no, yes	Y	Y	Y

Variable	Scale	Source	Topic	Code	Meaning	RD	BD	OD
IOTHR	bd	INDUSTRY	other	0, 1	no, yes	Y	Y	N
ORGSIZE	nominal	question 4	organisation size	1-5		Y	N	Y
OS1	bd	ORGSIZE	single person	0, 1	no, yes	N	Y	N
OS2	bd	ORGSIZE	single dept.	0, 1	no, yes	N	Y	N
OS3	bd	ORGSIZE	many depts one site	0, 1	no, yes	N	Y	N
OS4	bd	ORGSIZE	multi sites	0, 1	no, yes	N	Y	N
OS5	bd	ORGSIZE	multi national	0, 1	no, yes	N	Y	N
IMPORTAN	ordinal	question 5	spr/sht importance	1-3		Y	N	Y
IMP1	bd	IMPORTAN	unimportant	0, 1	no, yes	N	Y	N
IMP2	bd	IMPORTAN	moderate	0, 1	no, yes	N	Y	N
IMP3	bd	IMPORTAN	major	0, 1	no, yes	N	Y	N
ENUFTIME	bd	question 6	enough time	0, 1	no, yes	Y	Y	Y
SDPOLICY	bd	question 7	development policy	0, 1	no, yes	Y	Y	N
SDDOCO	bd	question 8	policy document	0, 1	no, yes	Y	Y	N
SDPOLDC	ordinal	SDPOLICY, SDDOCO	development policy rater	1	no policy	N	N	Y
				2	no doco			
				3	doc policy			
SDENFORC	nominal	question 9	dev policy enforced	1-5		Y	N	N
*SDENFORC	nominal	SDENFORC	development policy enforcement rater	0	not enforced	N	N	Y
				1	self enforced			
				2	dept enforced			
				3	other			
SDENF0	bd	*SDENFORC	not enforced	0, 1	no, yes	N	Y	N
SDENF1	bd	*SDENFORC	self enforced	0, 1	no, yes	N	Y	N
SDENF2	bd	*SDENFORC	dept enforced	0, 1	no, yes	N	Y	N
SDENF3	bd	*SDENFORC	other enforced.	0, 1	no, yes	N	Y	N
LIBRARY	bd	question 10	spreadsheet library	0, 1	no, yes	Y	Y	Y
XSDENVRN	ordinal	LIBRARY, SDPOLICY, SDDOCO and SDENFORC	spreadsheet development environment general rater	1 - 5		Y	N	N

Variable	Scale	Source	Topic	Code	Meaning	RD	BD	OD
USERGRP	bd	question 11	user group	0, 1	no, yes	Y	Y	Y
GENDER	bd	question 12	gender	0, 1	female, male	Y	Y	Y
AGE	ordinal	question 13	age	1 - 4		Y	N	Y
AGE1	bd	AGE	<25	0, 1	no, yes	N	Y	N
AGE2	bd	AGE	25 - 34	0, 1	no, yes	N	Y	N
AGE3	bd	AGE	35 - 44	0, 1	no, yes	N	Y	N
AGE4	bd	AGE	45 +	0, 1	no, yes	N	Y	N
EXPERT	ordinal	question 14	sp/sht expertise	1 - 3		Y	N	Y
EXPERT1	bd	EXPERT	novice	0, 1	no, yes	N	Y	N
EXPERT2	bd	EXPERT	knowledgeable	0, 1	no, yes	N	Y	N
EXPERT3	bd	EXPERT	power user	0, 1	no, yes	N	Y	N
TRAINING	nominal	question 15	spr/sht training	1 - 4		Y	N	N
TRAIN1	bd	TRAINING	prof DP	0, 1	no, yes	N	Y	N
TRAIN2	bd	TRAINING	course	0, 1	no, yes	N	Y	N
TRAIN3	bd	TRAINING	workmates	0, 1	no, yes	N	Y	N
TRAIN4	bd	TRAINING	self	0, 1	no, yes	N	Y	N
XTRAIN	bd	TRAINING	training rater	0	self	N	N	Y
				1	workmates			
				2	course			
				3	prof DP			
READ	ordinal	question 16	reads about spr/shts	1 - 3		Y	N	Y
READ1	bd	READ	<3 /yr	0, 1	no, yes	N	Y	N
READ2	bd	READ	3 - 8 /yr	0, 1	no, yes	N	Y	N
READ3	bd	READ	> 8 /yr	0, 1	no, yes	N	Y	N
QUALIFY	ordinal	question 17	highest qualification	1 - 5		Y	N	Y
QUAL1	bd	QUALIFY	school	0, 1	no, yes	N	Y	N
QUAL2	bd	QUALIFY	trade	0, 1	no, yes	N	Y	N
QUAL3	bd	QUALIFY	diploma	0, 1	no, yes	N	Y	N
QUAL4	bd	QUALIFY	degree	0, 1	no, yes	N	Y	N
QUAL5	bd	QUALIFY	postgraduate	0, 1	no, yes	N	Y	N

Variable	Scale	Source	Topic	Code	Meaning	RD	BD	OD
PROFMEMB	bd	question 18	prof membership	0, 1	no, yes	Y	Y	Y
PROFBODY$	alpha	question 18	Professional Body			N	N	N
XPROF	ordinal	QUALIFY, PROFMEMB	professionalism general rater	1 - 5		Y	N	N
JOB	nominal	question 19	occupation	1 - 8		Y	N	N
OMANAGR	bd	JOB	manager	0, 1	no, yes	Y	Y	N
OSCIENCE	bd	JOB	scientist	0, 1	no, yes	Y	Y	N
OTEACH	bd	JOB	academic / teacher	0, 1	no, yes	Y	Y	N
OACCNT	bd	JOB	accountant	0, 1	no, yes	Y	Y	N
OIT	bd	JOB	DP Professional	0, 1	no, yes	Y	Y	Y
OTRADE	bd	JOB	tradesperson	0, 1	no, yes	Y	Y	N
OCLERK	bd	JOB	clerk	0, 1	no, yes	Y	Y	N
OOTHER	bd	JOB	other	0, 1	no, yes	Y	Y	N
STATUS	nominal	question 20	employment status	1 - 7		Y	N	N
STCONS	bd	STATUS	consultant	0, 1	no, yes	Y	Y	Y
STEXEC	bd	STATUS	executive	0, 1	no, yes	Y	Y	N
STDMAN	bd	STATUS	dept manager	0, 1	no, yes	Y	Y	N
STEMP	bd	STATUS	employee	0, 1	no, yes	Y	Y	N
STSELFEM	bd	STATUS	self employed	0, 1	no, yes	Y	Y	Y
STHELP	bd	STATUS	unpaid helper	0, 1	no, yes	Y	Y	N
XSTATUS	ordinal	STATUS	status rater	0	cons / self employed	N	N	Y
				1	unpaid helper			
				2	employee			
				3	dept manager			
				4	executive			
PROGRAM$	alpha	question 21	software			N	N	N
VERSION$	alpha	question 21	version			N	N	N
ADDONS$	alpha	question 22	addons			N	N	N
OS$	alpha	question 23	operating system			N	N	N
SIZE	ratio	question 24	size in bytes			N	N	N
ROWS	ratio	question 25	no of rows			N	N	N

Variable	Scale	Source	Topic	Code	Meaning	RD	BD	OD
COLUMNS	ratio	question 26	no of columns			N	N	N
THREED	bd	question 27	3D	0,1	no, yes	Y	Y	N
WSHEETS	ratio	question 28	no of worksheets			Y	N	N
THREED*	ordinal	THREED	w/sht dimens.rater	0	2D	Y	N	Y
				1	2-3 w/shts			
				2	4-10 w/shts			
				3	>10 w/shts			
CELLLFORM	ordinal	question 29	% cells - formulas	1 - 5		N	N	N
CELLDATA	ordinal	question 30	% cells - data	1 - 5		N	N	N
CELLBLNK	ordinal	question 31	% cells - blank	1 - 5		N	N	N
CELLCONS	ordinal	question 32	% cells - constants	1 - 5		N	N	N
CELLLABL	ordinal	question 33	% cells - labels	1 - 5		N	N	N
CELLOTHR	ordinal	question 34	% cells - other	1 - 5		N	N	N
XSIZE	ordinal	calculated by sp/sht	Useful size	1 - 6		Y	N	Y
XSZ1	bd	XSIZE	XSIZE ----> 5000	0,1	no, yes	N	Y	N
XSZ2	bd	XSIZE	XSIZE ---> 10000	0,1	no, yes	N	Y	N
XSZ3	bd	XSIZE	XSIZE --->100000	0,1	no, yes	N	Y	N
XSZ4	bd	XSIZE	XSIZE---->500000	0,1	no, yes	N	Y	N
XSZ5	bd	XSIZE	XSIZE -->2000000	0,1	no, yes	N	Y	N
XSZ6	bd	XSIZE	XSIZE > 2000000	0,1	no, yes	N	Y	N
ABSREL	bd	question 35	abs/rel referencing	0,1	no, yes	Y	Y	N
SPLITSCRN	bd	question 36	split screens	0,1	no, yes	Y	Y	N
BORDERS	bd	question 37	borders	0,1	no, yes	Y	Y	N
LINKSS	bd	question 38	links to sp/shts	0,1	no, yes	Y	Y	Y
LINKDB	bd	question 39	links to DBMS	0,1	no, yes	Y	Y	Y
LINKDDE	bd	question 40	DDE	0,1	no, yes	Y	Y	Y
LINKED	ordinal	LINKSS, /DB, LINKDDE	link rater	0 - 3	.	N	N	Y
XCOMPLEX	ordinal	LINKED, ABSREL,	complexity rater	0 - 8		Y	N	N

Variable	Scale	Source	Topic	Code	Meaning	RD	BD	OD
		SPLITSCRN						
GRAPHICS	bd	question 41	graphics	0,1	no, yes	N	N	N
GRAPHSOP	ordinal	question 42	sophistication	1 - 3		N	N	N
XGRAPH	ordinal	GRAPHICS ,	graphics sophisti-	0	none	Y	N	Y
		GRAPHSOP	cation rater	1	simple			
				2	intermediate			
				3	complex			
XGRAPH0	bd	XGRAPH	no graphics	0,1	no, yes	N	Y	N
XGRAPH1	bd	XGRAPH	simple graphics	0,1	no, yes	N	Y	N
XGRAPH2	bd	XGRAPH	intermediate. graphics	0,1	no, yes	N	Y	N
XGRAPH3	bd	XGRAPH	complex graphics	0,1	no, yes	N	Y	N
MACROS	bd	question 43	macros	0,1	no, yes	N	N	N
MACROCOM	ordinal	question 44	sophistication	1 - 3		N	N	N
XMACRO	ordinal	MACROS,	macro sophistication	0	none	Y	N	Y
		MACROCOM	rater	1	simple			
				2	intermediate			
				3	complex			
XMACRO0	bd	XMACRO	no macros	0,1	no, yes	N	Y	N
XMACRO1	bd	XMACRO	simple macros	0,1	no, yes	N	Y	N
XMACRO2	bd	XMACRO	intermediate macros	0,1	no, yes	N	Y	N
XMACRO3	bd	XMACRO	complex macros	0,1	no, yes	N	Y	N
MODULAR	bd	question 45	modular	0,1	no, yes	N	N	N
MODARRANG	bd	question 46	arrangement	0,1	diag / block	N	N	N
MODARRANG	nominal	MODULAR,	module type	0	no modules	Y	N	N
		MODARRANG		1	blocked			
				2	diagonal			
MODBLOC	bd	MODARRANG	blocked modules	o, 1	no, yes	N	Y	N
MODDIAG	bd	MODARRANG	diagonal modules	0, 1	yes, no	N	Y	N
LOOKUPS	bd	question 47	LOOKUP functions	0, 1	yes, no	Y	Y	N
IFS	bd	question 48	IF function	0, 1	yes, no	Y	Y	N
NESTEDIF	bd	question 49	nested IF	0, 1	yes, no	Y	Y	N

Variable	Scale	Source	Topic	Code	Meaning	RD	BD	OD
XLOGIC	ordinal	IFS, NESTEDIF , LOOKUPS	logical complexity general rater	0 - 4		N	N	Y
FORMCOMP	ordinal	question 50	formula complexity	1 - 3		Y	N	Y
FORMCOMP1	bd	FORMCOMP	simple formulas	0, 1	no, yes	N	Y	N
FORMCOMP2	bd	FORMCOMP	average formulas	0, 1	no, yes	N	Y	N
FORMCOMP3	bd	FORMCOMP	complex formulas	0, 1	no, yes	N	Y	N
XFORMULA	ordinal	FORMCOMP, XLOGIC	general formula complexity	1 - 7		Y	N	N
RUNBY	ordinal	question 51	spreadsheet run by	1 - 3		Y	N	Y
RUNBY1	bd	RUNBY	self	0, 1	no, yes	N	Y	N
RUNBY2	bd	RUNBY	2 - 3 others	0, 1	no, yes	N	Y	N
RUNBY3	bd	RUNBY	many	0, 1	no, yes	N	Y	N
ENTERER	nominal	question 52	data entered by	1 - 3		N	N	N
ENTSELF	bd	ENTERER	self	0, 1	no, yes	Y	Y	N
ENTCLRK	bd	ENTERER	clerk	0, 1	no, yes	Y	Y	N
ENTUSER	bd	ENTERER	user	0, 1	no, yes	Y	Y	N
ENTKNOW	ordinal	ENTERER	enterer's spreadsheet knowledge	1 2 3	user clerk self	N	N	Y
PRIVATE	bd	question 53	private data used	0, 1	no, yes	Y	Y	Y
OUTSCOPE	ordinal	question 54	output range	1 - 4		N	N	Y
OUTSELF	bd	OUTSCOPE	self only	0, 1	no, yes	Y	Y	N
OUT1DEP	bd	OUTSCOPE	intra dept	0, 1	no, yes	Y	Y	N
OUTMDEP	bd	OUTSCOPE	inter dept	0, 1	no, yes	Y	Y	N
OUTEXORG	bd	OUTSCOPE	inter organisation	0, 1	no, yes	Y	Y	N
HOWOFTEN	nominal	question 55	run frequency	1 - 7		N	N	N
XFREQ	nominal	HOWOFTEN	run frequency	1 2 3 4	once few day / week / frequently month	Y	N	N

Variable	Scale	Source	Topic	Code	Meaning	RD	BD	OD
				5	occasional / long gap			
XORDFREQ	ordinal	HOWOFTEN	frequency rater	1	once	N	N	Y
				2	few/ long gap			
				3	month			
				4	day / week / frequently			
XFREQ1	bd	XFREQ	one shot model	0, 1	no, yes	N	Y	N
XFREQ2	bd	XFREQ	run few times	0, 1	no, yes	N	Y	N
XFREQ3	bd	XFREQ	frequent / regular	0, 1	no, yes	N	Y	N
XFREQ4	bd	XFREQ	monthly	0, 1	no, yes	N	Y	N
XFREQ5	bd	XFREQ	occasional / gap	0, 1	no, yes	N	Y	N
CORPDATA	bd	question 56	input corporate data	0,1	no, yes	Y	Y	N
WHEREFROM	nominal	question 57	where from	1 - 3		N	N	N
CDETRAN	bd	WHEREFROM	electronic transfer	0, 1	no, yes	Y	Y	N
CDRPTS	bd	WHEREFROM	ex reports	0, 1	no, yes	Y	Y	N
CDOTHR	bd	WHEREFROM	other	0, 1	no, yes	Y	Y	N
CDCHNG	bd	question 58	corp data changed	0, 1	no, yes	N	N	N
CDCHNG*	ordinal	CORPDATA,	corp data rater	0	no Corp data	N	N	Y
		CDCHNG		1	read only			
				2	changed			
XCDMOD	bd	CORPDAT,	corp data changed	0	None or unchanged	Y	Y	N
		CDCHNG		1	CD changed			
CDNEW	bd	question 59	new corp data	0, 1	no, yes	Y	Y	Y
KEPT	ordinal	question 60	how long kept	0, 1		Y	N	Y
KEPT1	bd	KEPT	< 1 week	0, 1	no, yes	N	Y	N
KEPT2	bd	KEPT	< 1 month	0, 1	no, yes	N	Y	N
KEPT3	bd	KEPT	> 1 month	0, 1	no, yes	N	Y	N
POSTCODE$	alpha	derived	postcode			N	N	N
STRATUM	nominal	POSTCODE$	sample stratum	1	Preston	Y	Y	Y
				2	Perth	Y	Y	Y

Variable	Scale	Source	Topic	Code	Meaning	RD	BD	OD
				3	Eastern States	Y	Y	Y

TAXONOMY SYSTAT RUN BINARY DICHOTOMOUS VARIABLES				
DATE		TIME		NO:
IN FILE:		STANDARDISED CORRELATED		TRANSPOSED
OUT FILE:		LOGFILE		PRINTED
KMEANS		NUMBER		
JOIN		ROWS	COLUMNS	MATRIX
DISTANCE	PCT	GAMMA	PEARSON	EUCLIDEAN
LINKAGE	SINGLE MEDIAN	COMPLETE WARD	CENTROID	AVERAGE
ATTRIBUTES				
LABEL$	IMP1	XMACRO0	CORPDATA	QUAL1
PCOMMS	IMP2	XMACRO1	CDETRAN	QUAL2
PREPORT	IMP3	XMACRO2	CORPTS	QUAL3
PCLASS	ENUFTIME	XMACRO3	COOTHR	QUAL4
PWHATIF	SDPOLICY	MODBLOC	XCDMOD	QUAL5
POPTIM	SDDOCO	MODDIAG	CDNEW	PROFMEMB
PFORCST	SDENF0	LOOKUPS	KEEP1	OMANAGR
SPUBLIC	SDENF1	IFS	KEEP2	OSCIENCE
SPRIVT	SDENF2	NESTEDIF	KEEP3	OTEACH
SPERSN	SDENF3	FORMCOM1	USERGRP	OACCNT
IAG	LIBRARY	FORMCOM2	GENDER	OIT
IMINE	THREED	FORMCOM3	AGE1	OCLERK
IMANUF	XSZ1	RUNBY1	AGE2	OOTHER
IELECT	XSZ2	RUNBY2	AGE3	STCONS
ICONST	XSZ3	RUNBY3	AGE4	STEXEC
ISELL	XSZ4	ENTSELF	EXPERT1	STDMAN
IFINCE	XSZ5	ENTCLRK	EXPERT2	STEMP
IBUSNS	XSZ6	ENTUSER	EXPERT3	STSELF
IPUBAD	ABSREL	PRIVATE	TRAIN1	STHELP
IEDUC	SPLITSCRN	OUTSELF	TRAIN2	
ICOMP	BORDERS	OUT1DEP	TRAIN3	
IOTHR	LINKSS	OUTMDEP	TRAIN4	
OS1	LINKDB	OUTEXORG	READ1	
OS2	LINKDDE	XFREQ1	READ2	
OS3	XGRAPH0	XFREQ2	READ3	
OS4	XGRAPH1	XFREQ3		
OS5	XGRAPH2	XFREQ4		
	XGRAPH3	XFREQ5		

Figure 7.1: Run recording sheet for Cluster Analysis of binary dichotomous variables.

TAXONOMY SYSTAT RUN ORDINAL VARIABLES				
DATE		TIME		NO:
IN FILE:		STANDARDISED CORRELATED		TRANSPOSED
OUT FILE:		LOGFILE		PRINTED
KMEANS		NUMBER		
JOIN		ROWS	COLUMNS	MATRIX
DISTANCE	PCT	GAMMA	PEARSON	EUCLIDEAN
LINKAGE	SINGLE MEDIAN	COMPLETE WARD	CENTROID	AVERAGE
ATTRIBUTES				

PWHATIF	LINKSS	USERGRP
POPTIM	LINKDB	GENDER
PFORCST	LINKDDE	AGE
PREST	XGRAPH	EXPERT
SPUBLIC	XMACRO	XTRAIN
SPRIVT	XLOGIC	READ
SPERSN	FORMCOMP	QUALIFY
ORGSIZE	RUNBY	PROFMEMB
IMPORTAN	ENTKNOW	STATUS
ENUFTIME	PRIVATE	STSELFEMP
SDPOLDC	OUTSCOPE	STCONS
SDEN ORC	XORDFREQ	ICOMP
LIBRARY	CDCHNG	OIT
THREED	CDNEW	PGROUP
XSIZE	KEPT	

Figure 7.2: Run recorder for cluster analysis of ordinal variables

APPENDIX C

SURVEY DATA

Table 25: **Part of Spreadsheet SIZE.SSF showing the calculation of 'useful size' and the variable XSIZE**

CASE	SIZE IN BYTES	CELL-FORM	CELL-DATA	% USE-FUL CELLS	USEFUL SIZE	XSIZE
71	9,668	2	4	100	9,668	2
35	90,357	1	4	100	90,357	3
78	100,000	1	1	40	40,000	3
24	2,048	1	1	40	819	1
56	33,000	1	2	60	19,800	3
57	9,000	3	1	80	7,200	2
62	30,000	1	1	40	12,000	3
69	36,864	3	3	100	36,864	3
30	4,096	1	1	40	1,638	1
89	4,000	1	3	80	3,200	1
23	34,304	1	1	40	13,722	3
20	26,624	3	1	80	21,299	3
55	137,216	1	1	40	54,886	3
76	370,688	1	1	40	148,275	4
90	23,000	1	2	60	13,800	3
102	6,084	1	1	40	2,434	1
21	6,024	2	2	80	4,819	1
58	800	1	2	60	480	1
54	32,142	2	3	100	32,142	3
107	197,000	1	4	100	197,000	4
53	495,664	1	1	40	198,266	4

Table 26: **Spreadsheet survey: Template: SIZE.SSF showing the average number of bytes occupied per cell for each case.**

CASE	PROGRAM	VERSION	SIZE	ROWS	COLS	WORKSHEETS	CELLS	BYTES / CELL
71	ABILITY	1.2	9,668	70	21	1	1,470	6.58
35	ASEASYAS	4	90,357	254	32	1	8,128	11.12
78	COMPUSHEET	CS+	100,000	163	46	1	7,498	13.34
24	ENABLE	2	2,048	30	60	1	1,800	1.14
56	ENABLE	2	33,000	148	22	1	3,256	10.14
57	ENABLE	2	9,000	50	13	1	650	13.85
62	ENABLE	2.14	30,000	26	20	1	520	57.69
69	ENABLE	2.14	36,864	107	16	1	1,712	21.53
30	ENABLE	2.2	4,096	25	7	1	175	23.41
89	ENABLE	3	4,000	25	9	1	225	17.78
23	ENABLE	3.57	34,304	82	32	1	2,624	13.07
20	ENABLE	OA	26,624	64	20	1	1,280	20.8
55	ENABLE	OA	137,216	57	23	6	7,866	17.44
76	ENABLE	OA	370,688	692	59	3	122,484	3.03
90	EXCEL		23,000	33,584	7	1	235,088	0.1
102	EXCEL		6,084	49	5	1	245	24.83
21	EXCEL	2	6,024	110	6	1	660	9.13
58	EXCEL	2	800	30	6	1	180	4.44
54	EXCEL	2.1	32,142	95	12	1	1,140	28.19
107	EXCEL	2.1	197,000	111	52	1	5,772	34.13
53	EXCEL	2.2	495,664	907	199	1	180,493	2.75
86	EXCEL	2.2	52,300	177	10	1	1,770	29.55

CASE	PROGRAM	VERSION	SIZE	ROWS	COLS	WORKSHEETS	CELLS	BYTES / CELL
95	EXCEL	2.2	333,000	1,404	62	1	87,048	3.83
13	EXCEL	3	49,428	150	16	1	2,400	20.6
94	EXCEL	3	100,000	300	29	1	8,700	11.49
4	EXCEL	3	17,000	44	15	1	660	25.76
6	EXCEL	3	44,091	424	10	1	4,240	10.4
10	EXCEL	3	200,000	48	28	8	10,752	18.6
19	EXCEL	3	5,343,956	57	6306	1	359,442	14.87
22	EXCEL	3	73,500	600	8	1	4,800	15.31
40	EXCEL	3	61,000	145	48	1	6,960	8.76
49	EXCEL	3	286,000	290	92	1	26,680	10.72
51	EXCEL	3	39,774	87	18	1	1,566	25.4
84	EXCEL	3	24,000	64	11	7	4,928	4.87
93	EXCEL	3	320,000	235	67	1	15,745	20.32
100	EXCEL	3	100,000	500	15	1	7,500	13.33
103	EXCEL	3	5,000	30	10	1	300	16.67
63	LOTUS		57,439	178	52	1	9,256	6.21
70	LOTUS		80,000	200	35	1	7,000	11.43
65	LOTUS	2	19,486	21	65	1	1,365	14.28
79	LOTUS	2	20,000	50	15	1	750	26.67
67	LOTUS	2	103,149	364	19	1	6,916	14.91
75	LOTUS	2	23,000	100	8	1	800	28.75
9	LOTUS	2.01	281,326	2,477	12	1	29,724	9.46
39	LOTUS	2.01	210,000	630	92	1	57,960	3.62
46	LOTUS	2.01	220,000	209	132	1	27,588	7.97
52	LOTUS	2.01	45,909	143	54	1	7,722	5.95
60	LOTUS	2.01	50,000	200	26	1	5,200	9.62
64	LOTUS	2.01	18,867	70	23	1	1,610	11.72
86	LOTUS	2.01	90,159	450	22	1	9,900	9.11
68	LOTUS	2.01	184,547	640	59	1	37,760	4.89
97	LOTUS	2.01	188,428	608	35	1	21,280	8.85

CASE	PROGRAM	VERSION	SIZE	ROWS	COLS	WORKSHEETS	CELLS	BYTES / CELL
8	LOTUS	2.1	46,000	100	18	1	1,800	25.56
26	LOTUS	2.2	251,084	109	24	1	2,616	95.98
27	LOTUS	2.2	24,790	63	7	1	441	56.21
37	LOTUS	2.2	250,000	456	95	1	43,320	5.77
47	LOTUS	2.2	321,985	1,533	34	1	52,122	6.18
73	LOTUS	2.2	20,637	85	18	1	1,530	13.49
81	LOTUS	2.2	40,000	250	80	1	20,000	2
14	LOTUS	3	30,000	204	6	1	1,224	24.51
17	LOTUS	3	721,534	270	32	13	112,320	6.42
15	LOTUS	3.1	200,000	400	20	12	96,000	2.08
25	LOTUS	3.1	450,000	60	16	5	4,800	93.75
38	LOTUS	3.1	842,317	116	52	52	313,664	2.69
41	LOTUS	3.1	400,000	150	30	14	63,000	6.35
42	LOTUS	3.1	9,353	34	12	1	408	22.92
43	LOTUS	3.1	4,200,000	4,500	14	8	504,000	8.33
44	LOTUS	3.1	371,770	153	22	15	50,490	7.36
50	LOTUS	3.1	242,000	2,128	54	1	114,912	2.11
74	LOTUS	3.1	87,926	470	67	5	157,450	0.56
96	LOTUS	3.1	19,916	45	25	2	2,250	8.85
98	LOTUS	3.1	160,000	150	27	7	28,350	5.64
80	LOTUSWORKS		3,415	31	9	1	279	12.24
61	MS WORKS	2	15,000	50	26	1	1,300	11.54
11	MS WORKS	2.00A	5,987	58	5	1	290	20.64
31	MS WORKS	2.00A	8,160	15	31	1	465	17.55
87	MS WORKS	2.00A	670,000	26	138	1	3,588	186.73
101	MS WORKS	2.00A	3,977	22	12	1	264	15.06

CASE	PROGRAM	VERSION	SIZE	ROWS	COLS	WORKSHEETS	CELLS	BYTES / CELL
5	MULTIPLAN	3	14,600	50	8	1	400	36.5
82	PRINTGRAPH		80,000	50	34	1	1,700	47.06
83	QUATTRO		240,000	1,200	12	1	14,400	16.67
106	QUATTRO		98,762	140	30	1	4,200	23.51
91	QUATTRO	1	18,000	150	14	1	2,100	8.57
2	QUATTRO	1	43,077	46	34	1	1,564	27.54
3	QUATTRO	1	436,000	799	99	1	79,101	5.51
18	QUATTRO	2	39,838	8,192	339	1	2,777,088	0.01
88	QUATTRO	2	18,505	43	54	1	2,322	7.97
1	QUATTRO	3	25,402	72	20	1	1,440	17.64
28	QUATTRO	3	20,000	200	8	1	1,600	12.5
92	QUATTRO	3	512,000	1,400	78	1	109,200	4.69
12	QUATTRO	3.01	115,630	133	24	1	3,192	36.22
16	QUATTRO	3.01	498,000	1,398	17	1	23,766	20.95
36	QUATTRO	3.01	11,904	41	14	1	574	20.74
99	QUATTRO	3.01	68,909	238	34	1	8,092	8.52
77	SUPERCALC	3	100,000	150	20	1	3,000	33.33
72	SUPERCALC	3	70,000	80	30	4	9,600	7.29
85	SUPERCALC	4	29,952	25	31	1	775	38.65
48	SUPERCALC	V 5	291,000	437	65	1	28,405	10.24
7	SYMPHONY	2	207,000	100	330	2	66,000	3.14
33	SYMPHONY	2.1	324,969	400	25	1	10,000	32.5
32	SYMPHONY	2.2	53,999	33	52	1	1,716	31.47

CASE	PROGRAM	VERSION	SIZE	ROWS	COLS	WORKSHEETS	CELLS	BYTES / CELL
105	TWIN	1	20,000	50	46	1	2,300	8.7
104	TWIN	3	4,000	60	15	1	900	4.44
45	UNIPLEX	7	10,000	50	15	1	750	13.33
34	UNIPLEX	V7	253,000	1,200	20	1	24,000	10.54
29	VP-PLANNER		44,000	50	50	1	2,500	17.6
59	VP-PLANNER		21,000	78	19	1	1,482	14.17

Table 27:

Frequencies of Values of variables in Binary Dichotomous data set

VARIABLE	0	1	TOTAL
PCOMMS	96	11	107
PREPORT	48	59	107
PCLASS	103	4	107
PWHATIF	99	8	107
POPTIM	101	6	107
PFORCST	88	19	107
SPUBLIC	73	34	107
SPRIVT	39	68	107
SPERSN	102	5	107
IAG	94	13	107
IMINE	81	26	107
IMANUF	102	5	107
IELECT	103	4	107
ICONST	105	2	107
ISELL	106	1	107
IFINCE	99	8	107
IBUSNS	91	16	107
IPUBAD	99	8	107
IEDUC	94	13	107
ICOMP	100	7	107
IOTHR	103	4	107
OS1	85	22	107
OS2	79	28	107
OS3	93	14	107
OS4	69	38	107

VARIABLE	0	1	TOTAL
OS5	102	5	107
IMP1	99	8	107
IMP2	50	57	107
IMP3	65	42	107
ENUFTIME	19	88	107
SDPOLICY	95	12	107
SDPOLICY	95	12	107
SDDOCO	103	4	107
SDENF0	107	0	107
SDENF1	98	9	107
SDENF2	103	4	107
SDENF3	106	1	107
LIBRARY	97	10	107
THREED	91	16	107
XSZ1	97	10	107
XSZ2	97	10	107
XSZ3	55	52	107
XSZ4	76	31	107
XSZ5	105	2	107
XSZ6	105	2	107
ABSREL	36	71	107
SPLITSCRN	80	27	107
BORDERS	54	53	107
LINKSS	68	39	107
LINKDB	83	24	107
LINKDDE	99	8	107
XGRAPH0	42	65	107
XGRAPH1	90	17	107
XGRAPH2	91	16	107

VARIABLE	0	1	TOTAL
XGRAPH3	98	9	107
XMACRO0	49	58	107
XMACRO1	87	20	107
XMACRO2	90	17	107
XMACRO3	95	12	107
MODBLOC	64	43	107
MODDIAG	93	14	107
LOOKUPS	77	30	107
IFS	56	51	107
NESTEDIF	77	30	107
FORMCOM1	61	46	107
FORMCOM2	60	47	107
FORMCOM3	93	14	107
RUNBY1	33	74	107
RUNBY2	83	24	107
RUNBY3	98	9	107
ENTSELF	31	76	107
ENTCLRK	96	11	107
ENTUSER	87	20	107
PRIVATE	72	35	107
OUTSELF	89	18	107
OUT1DEP	74	33	107
OUTMDEP	83	24	107
OUTEXORG	75	32	107
XFREQ1	101	6	107
XFREQ2	95	12	107
XFRE3	66	41	107
XFREQ4	76	31	107
XFREQ5	90	17	.07

VARIABLE	0	1	TOTAL
CORPDATA	42	65	107
CDETRAN	97	10	107
CDETRAN	97	10	107
CDRPTS	62	45	107
CDOTHR	97	10	107
XCDMOD	77	30	107
CDNEW	54	53	107
KEEP1	84	23	107
KEEP2	83	24	107
KEEP3	47	60	107
USERGRP	95	12	107
GENDER	16	91	107
AGE1	97	10	107
AGE2	72	35	107
AGE3	69	38	107
AGE4	83	24	107
EXPERT1	86	21	107
EXPERT2	36	71	107
EXPERT3	92	15	107
TRAIN1	86	21	107
TRAIN2	85	22	107
TRAIN3	98	9	107
TRAIN4	52	55	107
READ1	42	65	107
READ2	87	20	107
READ3	85	22	107
QUAL1	92	15	107
QUAL2	103	4	107
QUAL3	95	12	107

VARIABLE	0	1	TOTAL
QUAL4	64	43	107
QUAL5	74	33	107
PROFMEMB	56	51	107
OMANGR	80	27	107
OSCIENCE	78	29	107
OTEACH	95	12	107
OACCNT	83	24	107
OIT	98	9	107
OCLRK	105	2	107
OOTHER	103	4	107
STCONS	101	6	107
STEXEC	98	9	107
STDMAN	83	24	107
STEMP	54	53	107
STSELF	96	11	107
STHELP	103	4	107

APPENDIX D

OUTPUTS OF CLUSTER ANALYSES

EXPERIMENTAL RUNS TO DETERMINE SUITABLE PARAMETERS FOR USE IN PRODUCTION RUNS.

Table 28 in this Appendix gives details of all the cluster analysis runs performed. The first 84 runs were experimental in nature and were used to determine the most suitable paramaters for the production runs from which the taxonomy was devellopped.

Binary Dichotomous runs on mixed data

The seven runs in groups 1 to 7 analysed the binary dichotomous data-set with mixed attributes (i.e. application, developer and environmental attributes). The data-set was transposed and a correlation matrix was calculated using Jaccard's coefficient as a distance measure. The resulting matrix was input to the SYSTAT **JOIN** algorithm. Single, complete, centroid, average and Ward's linkage methods were experimented with. The data-set was too large to easily accommodate the statistical procedures available within the SYSTAT software, so the number of attributes used was decreased. Preference was given to those a priori attributes that were known prior to the development of the spreadsheet. Attributes measuring developer personal characteristics were removed. These runs demonstrated the software limitations and the necessity for restricting the number of variables used when clustering one hundred and six cases. The ordinal data-set had less variables than the binary dichotomous data-set and was used for the majority of the remaining runs.

Experimentation with clustering methods using ordinal variables.

The thirty-six runs in groups 8 to 12 clustered the ordinal data-set cases using developer attributes. Fifteen attributes were selected to measure the characteristics of the spreadsheet developer, e.g. qualifications, spreadsheet training and expertise. In group 8 runs, Euclidean distance was used both with average and Ward's linkage, and the results were compared to a **KMEANS** partitioning with ten clusters. **KMEANS** and **JOIN** using average linkage gave very similar results with only 13 out of 107 cases being placed in different clusters. The Pearson correlation coefficient for these two results was 0.93 showing a high positive correlation. Ward's linkage showed poorer agreement with the other two with a Pearson correlation coefficient of .589 when correlated with KMEANS. 39 out of 107 cases were allocated to different groups. Ward's linkage was considered unsuitable for further investigation.

Several clusters with more than three members were identified. One cluster consisted only of female developers and another of Academics acting as consultants. There were four groups with only one member. It was decided to continue with the ordinal data-set comparing KMEANS and JOIN algorithms.

Groups 9 and 10 runs investigated the use of the Goodman-Kruskal Gamma distance measure. This correlation measure was recommended for ordinal scales (Wilkinson, 1990, p. 58). Well separated clusters were obtained but their meaning was unclear and not so obvious as the clusters obtained using KMEANS and average linkage JOIN. The group 10 runs compared the KMEANS output for 10 clusters and JOIN for Gamma and Euclidean distance measures using average linkage on ranked and unranked data-sets. The results did not provide easily interpretable clusters.

The nineteen group 11 runs contrasted results received using Gamma, Kendall's Tau-b, Spearman Rho and Guttman Mu2 correlation coefficients with results obtained using the Euclidean distance coefficient. The attribute GENDER was discarded as this variable had been responsible for the formation of a group of female developers in previous runs. It was felt that a group based on gender would be unhelpful in developing a taxonomy designed for the control of spreadsheet development. The sector variables SPUBLIC and SPVRIVT (public and private sector) were also discarded for the same reason however SPERSN signifying personal or recreational development was retained.

Software constraints permitted the use of only ninety-nine cases when using correlation coefficients and the first ninety-nine were initially selected. Output using the Kendall Tau-b and Guttman Mu2 coefficients for ordinal data were compared with output using Euclidean and Gamma distance measures, and

KMEANS output for 7,8,9,10 and 11 clusters. Results showed a good match between Euclidean join of 8 clusters and KMEANS using 7 clusters with 11 mismatches out off 99 cases. MU2, Tau and Spearman Rho distance coefficients gave similar results to each other with 14 mismatches, however when they were compared with Euclidean JOIN using average linkage there were 30 mismatches.

Gamma distance measures disagreed with all others and some of the dendrogram had arms that did not join with the rest of the tree. It was decided to ignore Gamma coefficients. The clusters obtained using the other ordinal coefficients were not intuitive, so it was decided to discard them and continue the analysis using JOIN with Euclidean distance and average linkage and KMEANS. These two methods although based on different philosophies of clustering, one hierarchical and the other partitioned, gave results which were similar and furthermore easily interpretable and therefor useful.

Group 12 runs discarded GENDER but included both SPERSN and ORGSIZE reflecting the size of the organisation a developer worked for. Allowance was made, in some runs, for developers who either worked in the computer industry (ICOMP) or who classified themselves as computing professionals (OIT). All 107 variables remaining at this stage were included. Outputs of JOIN, using average linkage, and Euclidean distance were correlated with KMEANS for 6 and 7 clusters. Pearson correlation coefficients were used to compare the outputs of the clustering process. The JOIN had a .973 Pearson correlation with the KMEANS with 6 clusters and a .969 Pearson correlation with the KMEANS 7 cluster solution.Experimentation with clustering methods using binary dichotomous Variables

Nine group 13 and 14 runs repeated the analysis used with group 12 runs now using binary dichotomous variables and distance coefficients - PCT, Jaccard's and Anderburg's standardised S5. (Wilkinson, 1990) Most of the results were not encouraging and the software could not directly handle the larger data-sets required. This necessitated separate creation of a correlation matrix. The run using Jaccard's coefficient provided intuitive clusters:

- 11 employees either computer professionals or working in the computer industry. They had poor expertise but professional training, some worked in the personal or recreational sector some were self employed
- 48 developers with high expertise, working in larger organisations. All well qualified and trained often with professional qualifications.
- 16 developers with medium to low expertise. Younger or in smaller organisations. High qualifications but not really interested in spreadsheets.
- 5 computer consultants not particularly interested in spreadsheets
- 15 young less well qualified developers with average to low expertise.
- 3 non I.T. based executives. Older well qualified people with a low interest in spreadsheets.
- 7 non I.T. based executives with a high interest in spreadsheets
- 2 spreadsheet gurus. Professional D.P. spreadsheet consultants.

Experimentation with distance measures designed solely for ordinal data

To accommodate software constraints, a data-set containing only 99 cases was prepared for use in the thirty nine runs for groups 15 - 17. Eight cases were removed from the biggest clusters. The eight earliest joinings in the largest three groups were identified on the dendrogram. One of each pair was removed from the data-set. Coefficients recomended for use with ordinal data were tried i.e. Mu2, Rho, Tau and Gamma. (Wilkinson, 1990, p. 60) The analysis did not lead to intuitive cluster profiles and in some cases the tree dendrograms had arms that did not connect with the rest of the tree. The results were considered unsuitable for developing a taxonomy and it was decided to restrict further analysis to Kmeans clustering (partitioned) and Euclidean distance with average linkage joining (hierarchical).

Table 28 Cluster analysis runs and parameters

(R=ranked, S = Standadised, C = Correlated, T = Transposed)

Run	Scale	Attrib.	R	S	C	T	Method	No	Distance	Linkage	
1	bd	mixed			S3	y	join		Euclidean	average	
2		mixed				n	kmeans				too big for software
3		mixed				n	kmeans				too big for software
4		mixed				n	kmeans				too big for software
5	bd	apriori			S3	y	join		Euclidean	average	a priori attributes
6	bd	apriori			S3	y	join		Euclidean	complete	a priori attributes
7	bd	apriori			S3	y	join		Euclidean	Wards	a priori attributes
8a	ord	Dev		y			kmeans	10			including age and gender
8b	ord	Dev		y			join		Euclidean	average	including age and gender
8c	ord	Dev		y			join		Euclidean	Wards	including age and gender
9	ord	Dev					join		Gamma	average	including age and gender
10a	ord	Dev		y			kmeans	10			
10b	ord	Dev					join		Gamma	Average	including age and gender
10c	ord	Dev		y			join		Euclidean	average	including age and gender
11a	ord	Dev		y			join	8	Euclidean	average	no GENDER, SPUBLIC, SPRIVT
11b	ord	Dev		y			kmeans	8			
11c	ord	Dev		y			kmeans	9			
11d	ord	Dev		y			kmeans	10			
11e	ord	Dev		y			kmeans	11			
11f	ord	Dev					join		Gamma	average	
11g	ord	Dev			y	y	join		Tau	average	
11h	ord	Dev			y	y	join		MU2	average	
11i	ord	Dev			y	y	join Euclid.		Gamma	average	
11j	ord	Dev		y	y	y	join		Euclidean	average	
11k	ord	Dev			y	y	join gamma		Gamma	average	
11l	ord	Dev			y	y	join		Sp Rho	average	
11m	ord	Dev					kmeans	7			
11n	ord	Dev	y				kmeans	8			
11o	ord	Dev	y				join		Gamma	average	
11p	ord	Dev	y		y		join		Tau	average	
11q	ord	Dev	y		y		join		Mu2	average	
11r	ord	Dev	y		y		join		Gamma	average	
11s	ord	Dev	y		y		join		Rho	average	
12a	ord	Dev		y			join		Euclidean	average	+SPERSN+ORGSIZE no GENDER
12b	ord	Dev		y			kmeans	6			+SPERSN+ORGSIZE no GENDER
12c	ord	Dev		y			kmeans	7			+SPERSN+ORGSIZE no GENDER
12d	ord	Dev		y			join		Euclidean	average	+SPERSN+ORGSIZE no GENDER
12e	ord	Dev		y			kmeans	6			+SPERSN+ORGSIZE no GENDER
12f	ord	Dev		y			kmeans	7			+SPERSN+ORGSIZE no GENDER
12g	ord	Dev		y			join		Euclidean	average	+SPERSN+ORGSIZE no GENDER
12h	ord	Dev		y			kmeans	6			+SPERSN+ORGSIZE no GENDER
12i	ord	Dev		y			kmeans	7			+SPERSN+ORGSIZE no GENDER

<u>**Cluster Analysis Runs and Parameters**</u>

(R=ranked, S = Standadised, C = Correlated, T = Transposed)

Run	Scale	Attrib.	R	S	C	T	Method	No	Distance	Linkage	
13a	bd	Dev					join		PCT	average	developer variables
13b	bd	Dev					join		GAMMA	average	
13c	bd	Dev					join		Euclidean	average	
13d	bd	Dev					kmeans	5			
13e	bd	Dev					kmeans	6			
13f	bd	Dev					kmeans	7			
13g	bd	Dev					join		Jaccard	average	
13h	bd	Dev					join		Anderburg	average	
14	bd	Dev					join		Jaccard	average	Developer variables
15a	ord	Dev					join		Gamma	average	developer variables reduced data set
15b	ord	Dev					join		Euclidean	average	
15c	ord	Dev	y		y		join		Mu2	average	
15d	ord	Dev	y		y		join		Rho	average	
15e	ord	Dev	y		y		join		Tau	average	
15f	ord	Dev		y			kmeans	7			
15g	ord	Dev		y			kmeans	8			
15h	ord	Dev		y			kmeans	9			
15i	ord	Dev					join		Gamma	average	
15j	ord	Dev					join		Mu2	average	
15k	ord	Dev					join		Rho	average	
15l	ord	Dev					join		Tau	average	
15m	ord	Dev					join		S3 Jaccarc	average	
15n	ord	Dev					join		S5 Andert	average	
16a	ord	Dev		y			join		Gamma	average	developer variables
16b	ord	Dev	y				join		Gamma	average	
16c	ord	Dev					join		Euclidean	average	
17a	ord	Dev		y			join		Euclidean	average	reduced developer data set
17b	ord	Dev		y			kmeans	6			
17c	ord	Dev		y			kmeans	7			
17d	ord	Dev		y			kmeans	8			
17e	ord	Dev		y			kmeans	9			
17f	ord	Dev		y			kmeans	10			
17g	ord	Dev					join		Gamma	average	

Cluster Analysis Runs and Parameters

(R=ranked, S = Standadised, C = Correlated, T = Transposed)

Run	Scale	Attrib.	R	S	C	T	Method	No	Distance	Linkage	
17h	ord	Dev	y				join		Gamma	average	
17i	ord	Dev		y			join		Gamma	average	
17j	ord	Dev	y	y			join		Gamma	average	
17k	ord	Dev		y			join				repeat a
17l	ord	Dev		y			join				repeat c
17m	ord	Dev		y			join				repeat a
17n	ord	Dev		y			join				repeat c
17o	ord	Dev		y			matrix join				
18a	ord	Dev		y			join		Euclidean	average	weight EXPERT x 2
18b	ord	Dev		y			join		Euclidean	average	weight EXPERT x 3
18c	ord	Dev		y			join		Euclidean	average	weight XTRAIN x 2
18d	ord	Dev		y			join		Euclidean	average	weight XTRAIN x 2, EXPERT x 2
18e	ord	Dev		y			join		Euclidean	average	weight XTRAIN x 2, EXPERT x 3
18f	ord	Dev		y			kmeans	10			weight EXPERT x 2
18g	ord	Dev		y			kmeans	10			weight EXPERT x 3
18h	ord	Dev		y			kmeans	10			weight EXPERT x 3, XTRAIN x 2
18i	ord	Dev		y			join		Euclidean	average	wt EXPERTx3, XTRAINx2,MATRIX
20a	ord	Dev		y			join		Euclidean	average	no weighting
20b	ord	Dev		y			join		Euclidean	average	weight EXPERT x 3
20c	ord	Dev		y			join		Euclidean	average	weight EXPERT x 2
20d	ord	Dev		y			join		Euclidean	average	wt EXPERTx3,XTRAINx0,ICOMPx0
20e	ord	Dev		y			kmeans	12			wt EXPERTx3,XTRAINx0,ICOMPx0
20f	ord	Dev		y			kmeans	13			wt EXPERTx3,XTRAINx0,ICOMPx0
20g	ord	Dev		y			kmeans	14			wt EXPERTx3,XTRAINx0,ICOMPx0
20h	ord	Dev		y			kmeans	11			wt EXPERTx3,XTRAINx0,ICOMPx0
20i	ord	Dev		y			kmeans	10			wt EXPERTx3,XTRAINx0,ICOMPx0
20j	ord	Dev		y			kmeans	9			wt EXPERTx3,XTRAINx0,ICOMPx0
20k	ord	Dev		y			kmeans	8			wt EXPERTx3,XTRAINx0,ICOMPx0
20l	ord	Dev		y			kmeans	7			wt EXPERTx3,XTRAINx0,ICOMPx0
20m *	ord	Dev		y			join		Euclidean	average	wt EXPERTx3,XTRAINx0,ICOMPx0
20n	ord	Dev		y			kmeans	18			wt EXPERTx3,XTRAINx0,ICOMPx0
20o	ord	Dev		y			kmeans	21			wt EXPERTx3,XTRAINx0,ICOMPx0
20p	ord	Dev		y			kmeans	11			wt EXPERTx3,XTRAINx0,ICOMPx0
20q *	ord	Dev		y			kmeans	14			wt EXPERTx3,XTRAINx0,ICOMPx0
20r	ord	Dev		y			matrix join		Euclidean	average	wt EXPERTx3,XTRAINx0,ICOMPx0
21a	ord	non Dev		y			join		Euclidean	average	SPERSN in
21b	ord	non Dev		y			join		Euclidean	average	SPERSN out
21c	ord	non Dev		y			kmeans	8			
21d	ord	non Dev		y			kmeans	10			
21e	ord	non Dev		y			kmeans	13			

Cluster Analysis Runs and Parameters

(R=ranked, S = Standadised, C = Correlated, T = Transposed)

Run	Scale	Attrib.	R	S	C	T	Method	No	Distance	Linkage	
22a	ord	appl		y			join		Euclidean	average	SPERSN in
22b	ord	appl		y			join		Euclidean	average	SPERSN out
22c	ord	appl		y			kmeans	10			SPERSN out
23a	ord	appl		y			join		Euclidean	average	ENUFTIME out
23b	ord	appl		y			join		Euclidean	average	ENUFTIME in
23c	ord	appl		y			join		Euclidean	average	22 a with no case 15
23d	ord	appl		y			join		Euclidean	average	no PFORECAST
23e	ord	appl		y			join		Euclidean	average	add LINKED no LINKSS/DB/DDE
23f	ord	appl		y			join		Euclidean	average	weight IMPORTAN x 3
23g	ord	appl		y			join		Euclidean	average	weight SIZE x 3
23h	ord	appl		y			kmeans	10			
23i	ord	appl		y			kmeans	15			
23j	ord	appl		y			kmeans	18			
24a *	ord	appl		y			join		Euclidean	average	23e + ENTCLRK and ENTKNOW out
24b	ord	appl		y			join		Euclidean	average	24b + PFORECAST
24c	ord	appl		y			join		Euclidean	average	without cases 7, 95 and 19
24d	ord	appl		y			kmeans	10			as for 24a
24e	ord	appl		y			kmeans	18			as for 24a
24f	ord	appl		y			kmeans	20			as for 24a
24g *	ord	appl		y			kmeans	14			as for 24a
24h	ord	appl		y			kmeans	7			as for 24a
24i	ord	appl		y			kmeans	8			as for 24a, no case 19
24j *	ord	appl		y			kmeans	9			as for 24a
24k	ord	appl		y			join		Euclidean	average	as for 24a, cut to show 18 clusters
25a	ord	env		y			join		Euclidean	average	environment variavles
25b	ord	env		y			kmeans	4			
25c	ord	env		y			kmeans	5			
25d	ord	env		y			kmeans	6			
25e	ord	env		y			kmeans	7			
25f *	ord	env		y			join		Euclidean	average	+ SPERSN
25g *	ord	env		y			kmeans	7			
25h	ord	env		y			kmeans	8			
25i	ord	env		y			kmeans	9			

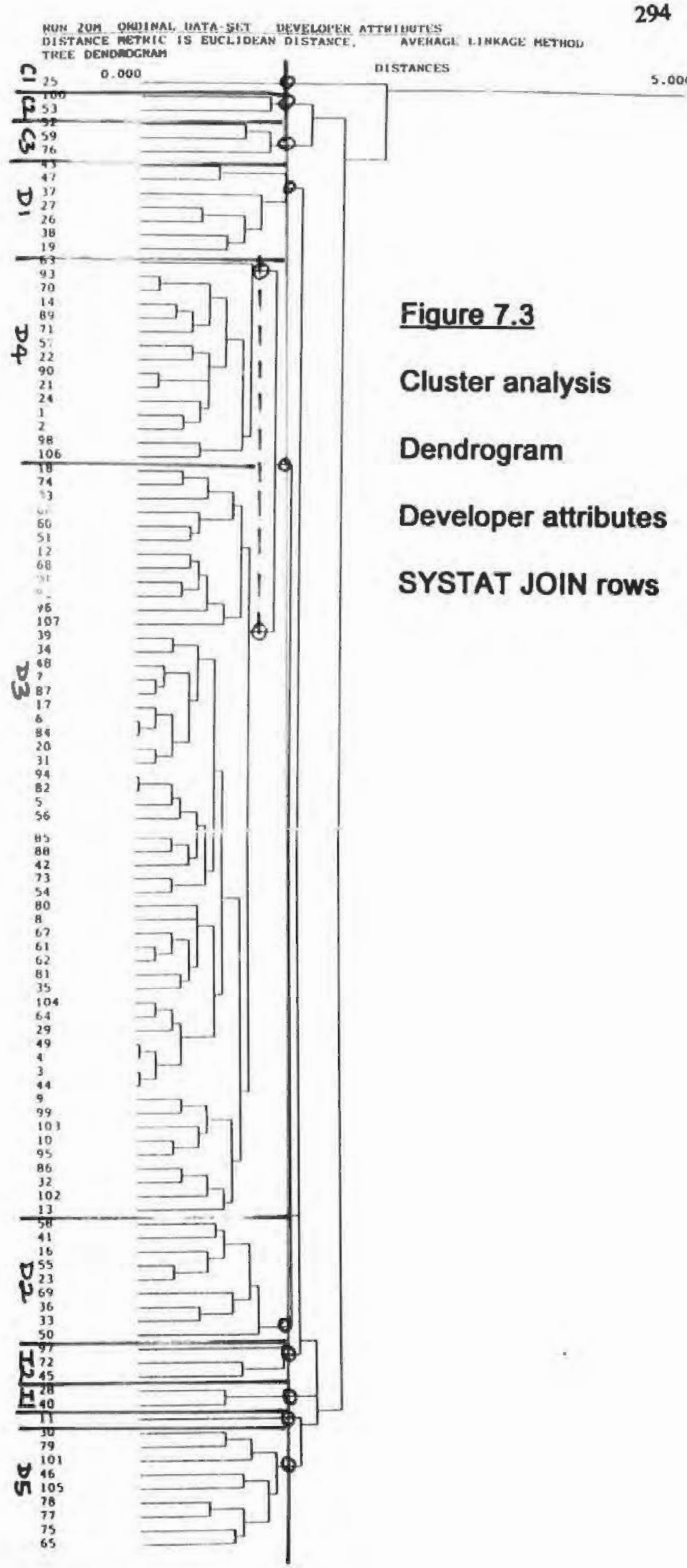


Figure 7.3

Cluster analysis

Dendrogram

Developer attributes

SYSTAT JOIN rows

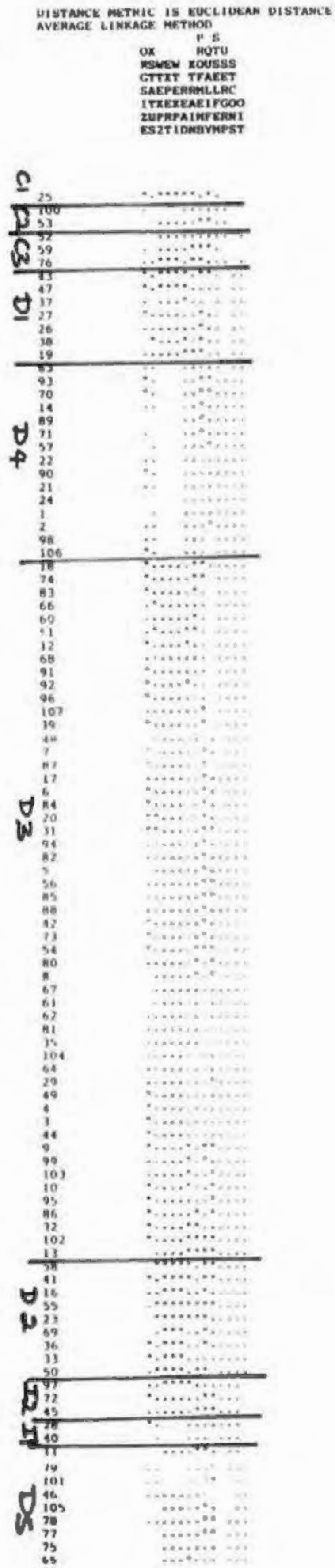


Figure 7.4

Cluster analysis

Dendrogram

Developer attributes

Shaded MATRIX plot

Table 29 Run 20q Kmeans analysis on ordinal Developer variables

KMEANS, SUMMARY STATISTICS FOR 14 CLUSTERS

VARIABLE	BETWEEN SS	DF	WITHIN SS	DF	F-RATIO	PROB
ORGSIZE	33.648	13	71.352	92	3.337	0.000
USERGRP	76.057	13	28.943	92	18.597	0.000
EXPERT	96.640	13	8.360	92	81.803	0.000
XTRAIN	20.290	13	84.710	92	1.695	0.075
READ	28.869	13	76.131	92	2.684	0.003
QUALIFY	12.622	13	92.378	92	0.967	0.489
PROFMEMB	20.976	13	84.024	92	1.767	0.060
XSTATUS	51.427	13	53.573	92	6.793	0.000
STSELFEM	99.203	13	5.797	92	121.109	0.000
STCONS	105.000	13	0.000	92	.	.
OIT	95.438	13	9.562	92	70.636	0.000
WTEXP1	96.640	13	8.360	92	81.803	0.000
WTEXP2	96.640	13	8.360	92	81.803	0.000

CLUSTER NUMBER: 1 D3 Knowledgeable Developers

MEMBERS

CASE	DISTANCE
3	0.50
4	0.50
5	0.65
6	0.52
7	0.51
8	0.78
9	0.74
10	0.72
12	0.80
13	0.91
17	0.49
18	0.59
20	0.67
29	0.49
31	0.73
32	0.72
34	0.59
35	0.47
39	0.72
42	0.45
44	0.50
48	0.43
49	0.50
51	0.96
54	0.66
55	0.60
60	0.78
61	0.64
62	0.49
64	0.47
66	0.79
67	0.62
68	0.61
73	0.58
74	0.78
80	0.64
81	0.43
82	0.48
83	0.77
84	0.52
85	0.57
86	0.63
87	0.51
88	0.40
91	0.78
92	0.80
94	0.48
95	0.56
96	0.71
99	0.64
102	0.79
103	0.75
104	0.54
107	0.73

STATISTICS

VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
ORGSIZE	-1.40	0.35	1.77	0.88
USERGRP	-0.36	-0.36	-0.36	0.00
EXPERT	0.12	0.12	0.12	0.00
XTRAIN	-0.87	0.12	1.56	1.00
READ	-0.74	0.01	1.72	0.96
QUALIFY	-2.03	-0.17	0.99	1.08
PROFMEMB	-1.03	-0.07	0.96	1.00
XSTATUS	-0.19	0.28	1.89	0.65
STSELFEM	-0.32	-0.32	-0.32	0.00
STCONS	-0.24	-0.24	-0.24	0.00
OIT	-0.30	-0.30	-0.30	0.00
WTEXP1	0.12	0.12	0.12	0.00
WTEXP2	0.12	0.12	0.12	0.00

CLUSTER NUMBER: 2 C1 Spreadsheet Expert and I.T. Consultant ("Guru")

MEMBERS

CASE	DISTANCE
25	0.00

STATISTICS

VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV
ORGSIZE	0.98	0.98	0.98	0.00
USERGRP	2.79	2.79	2.79	0.00
EXPERT	1.86	1.86	1.86	0.00
XTRAIN	1.56	1.56	1.56	0.00
READ	1.72	1.72	1.72	0.00
QUALIFY	0.99	0.99	0.99	0.00
PROFMEMB	-1.03	-1.03	-1.03	0.00
XSTATUS	-0.19	-0.19	-0.19	0.00
STSELFEM	-0.32	-0.32	-0.32	0.00
STCONS	4.06	4.06	4.06	0.00
OIT	3.27	3.27	3.27	0.00
WTEXP1	1.86	1.86	1.86	0.00
WTEXP2	1.86	1.86	1.86	0.00

CLUSTER NUMBER: 3 D4 Novice Developers

CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV
1	0.50	ORGSIZE	-1.40	-0.29	0.98	0.9
2	0.56	USERGRP	-0.36	-0.15	2.79	0.7
14	0.47	EXPERT	-1.63	-1.63	-1.63	0.0
21	0.32	XTRAIN	-0.87	-0.17	0.75	0.7
22	0.38	READ	-0.74	-0.50	0.49	0.4
24	0.50	QUALIFY	-2.03	0.08	0.99	0.9
57	0.51	PROFMEMB	-1.03	-0.37	0.96	0.9
63	0.84	XSTATUS	-1.22	-0.26	0.85	0.5
70	0.58	STSELFEM	-0.32	-0.32	-0.32	0.0
71	0.51	STCONS	-0.24	-0.24	-0.24	0.0
89	0.61	OIT	-0.30	-0.30	-0.30	0.0
90	0.45	WTEXP1	-1.63	-1.63	-1.63	0.0
93	0.61	WTEXP2	-1.63	-1.63	-1.63	0.0
98	0.64					
106	0.86					

CLUSTER NUMBER: 4 C3 Spreadsheet consultants , not I.T. Professionals

MEMBERS			STATISTICS			
CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
52	0.66	ORGSIZE	-1.40	-1.00	-0.61	0.4
53	0.92	USERGRP	-0.36	-0.36	-0.36	0.0
59	0.54	EXPERT	0.12	0.55	1.86	0.7
76	0.90	XTRAIN	-0.87	0.34	1.56	1.2
		READ	-0.74	-0.44	0.49	0.5
		QUALIFY	0.23	0.61	0.99	0.3
		PROFMEMB	-1.03	0.46	0.96	0.8
case 53 later assigned		XSTATUS	-2.26	-1.74	-0.19	0.9
to C2		STSELFEM	-0.32	-0.32	-0.32	0.0
		STCONS	4.06	4.06	4.06	0.0
		OIT	-0.30	0.59	3.27	1.5
		WTEXP1	0.12	0.55	1.86	0.7
		WTEXP2	0.12	0.55	1.86	0.7

CLUSTER NUMBER: 5 Not represented in the final taxonomy

MEMBERS			STATISTICS			
CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV
11	0.57	ORGSIZE	-1.40	-1.00	-0.61	0.4
45	0.57	USERGRP	-0.36	-0.36	-0.36	0.0
		EXPERT	0.12	0.12	0.12	0.0
case 11 later assigned		XTRAIN	1.56	1.56	1.56	0.0
to D5 self employed.		READ	-0.74	-0.74	-0.74	0.0
		QUALIFY	0.23	0.61	0.99	0.3
case 45 later assigned		PROFMEMB	0.96	0.96	0.96	0.0
to I1 IT employee		XSTATUS	-2.26	-1.22	-0.19	1.0
interested in		STSELFEM	-0.32	1.38	3.08	1.7
spreadsheets		STCONS	-0.24	-0.24	-0.24	0.0
		OIT	3.27	3.27	3.27	0.0
		WTEXP1	0.12	0.12	0.12	0.0
		WTEXP2	0.12	0.12	0.12	0.0

CLUSTER NUMBER: 6 Not represented in the Taxonomy

MEMBERS			STATISTICS			
CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV
97	0.00	ORGSIZE	0.19	0.19	0.19	0.0
		USERGRP	-0.36	-0.36	-0.36	0.0
		EXPERT	1.86	1.86	1.86	0.0
combined with cluster		XTRAIN	0.75	0.75	0.75	0.0
12. I1 I. employees		READ	1.72	1.72	1.72	0.
interested in spread-		QUALIFY	-0.52	-0.52	-0.52	0.
sheets		PROFMEMB	-1.03	-1.03	-1.03	0.00
		XSTATUS	-0.19	-0.19	-0.19	0.00
		STSELFEM	-0.32	-0.32	-0.32	0.00
		STCONS	-0.24	-0.24	-0.24	0.00
		OIT	3.27	3.27	3.27	0.00
		WTEXP1	1.86	1.86	1.86	0.00
		WTEXP2	1.86	1.86	1.86	0.00

CLUSTER NUMBER: 7 D5 Self-employed

MEMBERS			STATISTICS			
CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
30	0.75	ORGSIZE	1.40	-1.05	-0.61	0.39
46	0.92	USERGRP	-0.36	0.69	2.79	1.48
65	0.88	EXPERT	-1.63	-0.47	0.12	0.82
75	0.71	XTRAIN	-0.87	-0.69	0.75	0.51
77	0.58	READ	0.74	-0.4[illegible]	1.72	0.77
78	0.61	QUALIFY	-1.27	0.40	0.99	0.78
79	0.80	PROFMEMB	-1.03	0.07	0.96	0.99
101	0.93	XSTATUS	-2.76	-1.34	-0.19	1.03
105	0.75	STSELFEM	3.08	3.08	3.08	0.00
		STCONS	-0.24	-0.24	-0.24	0.00
		OIT	-0.30	-0.30	-0.30	0.00
		WTEXP1	-1.63	-0.47	0.12	0.82
		WTEXP2	-1.63	-0.47	0.12	0.82

CLUSTER NUMBER: 8 D2 Lay Experts

MEMBERS			STATISTICS			
CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV
16	0.19	ORGSIZE	-1.40	-0.25	0.98	1.00
23	0.64	USERGRP	-0.36	-0.36	-0.36	0.00
33	0.85	EXPERT	1.86	1.86	1.86	0.00
36	0.67	XTRAIN	-0.87	0.30	1.56	1.09
41	0.63	READ	-0.74	0.49	1.72	1.16
50	0.83	QUALIFY	-2.03	0.15	0.99	0.90
55	0.61	PROFMEMB	-1.03	0.74	0.96	0.6[illegible]
58	0.67	XSTATUS	-0.19	0.97	1.89	0.91
69	0.58	STSELFEM	-0.32	-0.32	-0.32	0.00
		STCONS	-0.24	-0.24	-0.24	0.00
		OIT	-0.30	-0.30	-0.30	0.00
		WTEXP1	1.86	1.86	1.86	0.00
		WTEXP2	1.86	1.86	1.86	0.00

CLUSTER NUMBER: 9 Not represented in the taxonomy

MEMBERS			STATISTICS			
CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
37	0.00	ORGSIZE	0.19	0.19	0.19	0.00
		USERGRP	2.79	2.79	2.79	0.00
later assigned to D1		EXPERT	0.12	0.12	0.12	0.00
user-group member		XTRAIN	-0.87	-0.87	-0.87	0.00
		READ	1.72	1.72	1.72	0.00
		QUALIFY	-2.03	-2.03	-2.03	0.00
		PROFMEMB	-1.03	-1.03	-1.03	0.00
		XSTATUS	-1.22	-1.22	-1.22	0.00
		STSELFEM	-0.32	-0.32	-0.32	0.00
		STCONS	-0.24	-0.24	-0.24	0.00
		OIT	-0.30	-0.30	-0.30	0.00
		WTEXP1	0.12	0.12	0.12	0.00
		WTEXP2	0.12	0.12	0.12	0.00

CLUSTER NUMBER: 10 D1 User group members

MEMBERS			STATISTICS			
CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
19	0.60	ORGSIZE	-1.40	-0.41	0.98	0.86
26	0.30	USERGRP	2.79	2.79	2.79	0.00
27	0.64	EXPERT	0.12	0.12	0.12	0.00
38	0.55	XTRAIN	-0.87	-0.26	1.56	1.05
		READ	-0.74	0.80	1.72	1.02
		QUALIFY	-0.52	0.05	0.23	0.33
		PROFMEMB	0.96	0.96	0.96	0.00
		XSTATUS	-0.19	0.59	1.89	0.86
		STSELFEM	-0.32	-0.32	-0.32	0.00
		STCONS	-0.24	-0.24	-0.24	0.00
		OIT	-0.30	-0.30	-0.30	0.00
		WTEXP1	0.12	0.12	0.12	0.00
		WTEXP2	0.12	0.12	0.12	0.00

CLUSTER NUMBER: 11 C2 I.T. consultants - not spreadsheet experts

MEMBERS			STATISTICS			
CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV
100	0.00	ORGSIZE	-0.61	-0.61	-0.61	0.0
		USERGRP	-0.36	-0.36	-0.36	0.0
		EXPERT	-1.63	-1.63	-1.63	0.0
		XTRAIN	-0.87	-0.87	-0.87	0.0
		READ	-0.74	-0.74	-0.74	0.0
		QUALIFY	0.23	0.23	0.23	0.0
		PROFMEMB	-1.03	-1.03	-1.03	0.0
		XSTATUS	-1.22	-1.22	-1.22	0.0
		STSELFEM	-0.32	-0.32	-0.32	0.0
		STCONS	4.06	4.06	4.06	0.0
		OIT	3.27	3.27	3.27	0.0
		WTEXP1	-1.63	-1.63	-1.63	0.0
		WTEXP2	-1.63	-1.63	-1.63	0.0

CLUSTER NUMBER: 12 I1 I.T. employees non consultants, interested in spreadsheets

MEMBERS			STATISTICS			
CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
72	0.00	ORGSIZE	0.98	0.98	0.98	0.00
		USERGRP	-0.36	-0.36	-0.36	0.00
		EXPERT	0.12	0.12	0.12	0.00
		XTRAIN	-0.87	-0.87	-0.87	0.00
		READ	0.49	0.49	0.49	0.00
		QUALIFY	0.99	0.99	0.99	0.00
		PROFMEMB	0.96	0.96	0.96	0.00
		XSTATUS	0.85	0.85	0.85	0.00
		STSELFEM	-0.32	-0.32	-0.32	0.00
		STCONS	-0.24	-0.24	-0.24	0.00
		OIT	3.27	3.27	3.27	0.00
		WTEXP1	0.12	0.12	0.12	0.00
		WTEXP2	0.12	0.12	0.12	0.00

CLUSTER NUMBER: 13 Not represented in the taxonomy

MEMBERS			STATISTICS			
CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
43	0.35	ORGSIZE	1.77	1.77	1.77	0.00
47	0.35	USERGRP	2.79	2.79	2.79	0.00
		EXPERT	1.86	1.86	1.86	0.00
Both cases were		XTRAIN	-0.87	-0.87	-0.87	0.
transfered to U1 user-		READ	1.72	1.72	1.72	0.00
group members		QUALIFY	-0.52	0.23	0.99	0.75
		PROFMEMB	-1.07	-0.04	0.96	1.00
		XSTATUS	0.85	0.85	0.85	0.00
		STSELFEM	-0.32	-0.32	-0.32	0.00
		STCONS	-0.24	-0.24	-0.24	0.00
		OIT	-0.30	-0.30	-0.30	0.00
		WTEXP1	1.86	1.86	1.86	0.00
		WTEXP2	1.86	1.86	1.86	0.00

CLUSTER NUMBER: 14 I2 I.T. employees disinterested in spreadsheets

MEMBERS			STATISTICS			
CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV
28	0.39	ORGSIZE	-1.40	-0.21	0.98	1.1
40	0.39	USERGRP	-0.36	-0.36	-0.36	0.0
		EXPERT	-1.63	-1.63	-1.63	0.0
		XTRAIN	-0.87	-0.87	-0.87	0.0
		READ	-0.74	-0.74	-0.74	0.0
		QUALIFY	-1.27	-0.52	0.23	0.7
		PROFMEMB	-1.03	-1.03	-1.03	0.0
		XSTATUS	-0.19	-0.19	-0.19	0.0
		STSELFEM	-0.32	-0.32	-0.32	0.0
		STCONS	-0.24	-0.24	-0.24	0.0
		OIT	3.27	3.27	3.27	0.0
		WTEXP1	-1.63	-1.63	-1.63	0.0
		WTEXP2	-1.63	-1.63	-1.63	0.0

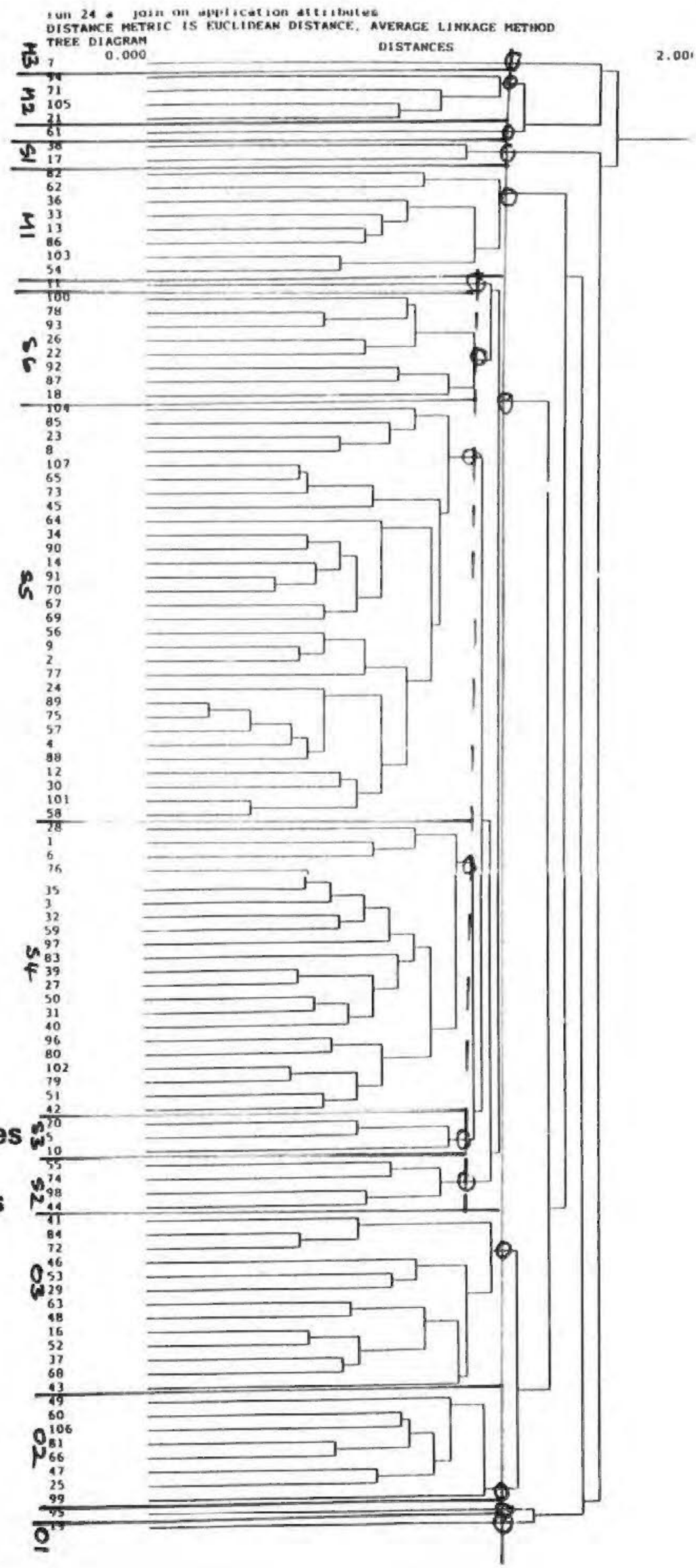


Figure 7.5

Cluster analysis

Dendrogram

Application attributes

SYSTAT JOIN rows

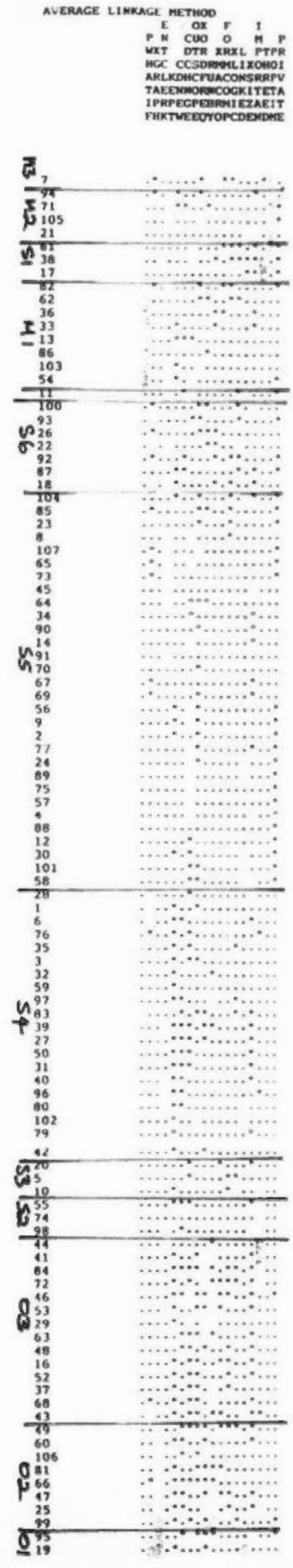


Figure 7.6

Cluster analysis

Dendrogram

Application attributes

Shaded MATRIX plot

Table 30 Run 24j Kmeans analysis on ordinal Application variables

SUMMARY STATISTICS FOR 9 CLUSTERS

VARIABLE	BETWEEN SS	DF	WITHIN SS	DF	F-RATIO	PROB
PWHATIF	105.000	8	0.000	97	.	.
POPTIM	105.000	8	0.000	97	.	.
IMPORTAN	28.145	8	76.855	97	4.440	0.000
THREED	45.489	8	59.511	97	9.268	0.000
XSIZE	40.105	8	64.895	97	7.493	0.000
LINKED	32.878	8	72.122	97	5.527	0.000
XGRAPH	30.858	8	74.142	97	5.046	0.000
XMACRO	58.335	8	46.665	97	15.157	0.000
XLOGIC	38.910	8	66.090	97	7.139	0.000
FORMCOMP	39.966	8	65.034	97	7.451	0.000
RUNBY	44.026	8	60.974	97	8.755	0.000
PRIVATE	15.136	8	89.864	97	2.042	0.049
OUTSCOPE	21.924	8	83.076	97	3.200	0.003
XORDFREQ	15.539	8	89.461	97	2.106	0.042
CDCHNGE	28.797	8	76.203	97	4.582	0.000
CDNEW	28.351	8	76.649	97	4.485	0.000
KEPT	11.380	8	93.620	97	1.474	0.177
ENTCLRK	41.341	8	63.659	97	7.874	0.000

CLUSTER NUMBER: 1 S5 - Non 3D, General
S4 - Non 3D, Corporate data creators
S2 3D. simple

MEMBERS

CASE	DISTANCE	CASE	DISTANCE
1	0.83	74	0.77
2	0.62	75	0.65
3	0.78	76	0.79
4	0.78	77	0.80
5	0.89	79	0.62
6	0.94	80	0.82
8	0.90	88	0.50
9	0.81	89	0.74
11	0.98	90	0.64
12	0.64	91	0.56
14	0.62	96	0.79
23	0.83	98	1.01
24	0.76	101	0.77
28	0.94	102	0.77
30	0.77	104	0.88
31	0.82		
32	0.75		
34	0.69		
35	0.77		
40	0.79		
42	0.62		
45	0.76		
51	0.84		
55	0.95		
56	0.77		
57	0.69		
58	0.89		
59	0.60		
64	0.80		
65	0.83		
67	0.72		
69	0.76		
70	0.49		
73	0.74		

STATISTICS

VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
PWHATIF	-0.28	-0.28	-0.28	0.0(
POPTIM	-0.24	-0.24	-0.24	0.0(
IMPORTAN	-2.16	-0.24	1.13	0.9(
THREED	-0.38	-0.16	2.28	0.6(
XSIZE	-2.10	-0.49	0.89	0.9:
LINKED	-0.80	-0.45	1.65	0.7(
XGRAPH	-0.70	-0.14	2.34	0.91
XMACRO	-0.78	-0.63	1.12	0.4(
XLOGIC	-0.88	-0.53	1.84	0.71
FORMCOMP	-1.01	-0.59	0.46	0.6(
RUNBY	-0.60	-0.51	1.00	0.3(
PRIVATE	-0.70	0.12	1.42	1.0:
OUTSCOPE	-1.52	-0.18	1.26	1.04
XORDFREQ	-2.21	-0.14	1.09	0.9(
CDCHNGE	-1.08	-0.30	1.38	0.9(
CDNEW	-0.98	-0.12	1.01	0.9(
KEPT	-1.64	0.01	0.81	1.01
ENTCLRK	-0.34	-0.34	-0.34	0.00

CLUSTER NUMBER: 2 M3 - Models, very complex

MEMBERS

CASE	DISTANCE
7	0.00

STATISTICS

VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
PWHATIF	-0.28	-0.28	-0.28	0.00
POPTIM	4.06	4.06	4.06	0.00
IMPORTAN	-0.51	-0.51	-0.51	0.00

VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
THREED	0.95	0.95	0.95	0.00
XSIZE	0.89	0.89	0.89	0.00
LINKED	-0.80	-0.80	-0.80	0.00
XGRAPH	1.33	1.33	1.33	0.00
XMACRO	2.08	2.08	2.08	0.00
XLOGIC	1.84	1.84	1.84	0.00
FORMCOMP	1.92	1.92	1.92	0.00
RUNBY	2.60	2.60	2.60	0.00
PRIVATE	-0.70	-0.70	-0.70	0.00
OUTSCOPE	0.33	0.33	0.33	0.00
XORDFREQ	1.09	1.09	1.09	0.00
CDCHNGE	0.15	0.15	0.15	0.00
CDNEW	-0.98	-0.98	-0.98	0.00
KEPT	-0.42	-0.42	-0.42	0.00
ENTCLRK	-0.34	-0.34	-0.34	0.00

CLUSTER NUMBER: 3 O3 - Data entry by user, Important spreadsheets
O2 - Data entry by data-entry clerk, Important
S3 - Non 3D, large and complex

MEMBERS

CASE	DISTANCE
10	1.07
16	0.68
18	1.00
20	0.88
25	0.98
27	0.58
29	0.94
37	0.73
39	0.65
41	1.03
43	1.00
46	0.92
47	0.82
48	0.64
49	1.01
50	0.76
52	0.70
53	0.78
60	0.91
63	0.80
66	0.90
68	0.67
72	0.90
81	0.81
83	0.86
84	0.87
87	0.83
97	0.78
99	1.11

STATISTICS

VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
PWHATIF	-0.28	-0.28	-0.28	0.00
POPTIM	-0.24	-0.24	-0.24	0.00
IMPORTAN	-0.51	0.74	1.13	0.70
THREED	-0.38	0.17	3.61	1.13
XSIZE	-0.10	0.55	2.88	0.66
LINKED	-0.80	0.55	2.88	0.93
XGRAPH	-0.70	-0.14	2.34	0.90
XMACRO	-0.78	0.96	2.08	1.00
XLOGIC	-0.80	0.78	1.84	0.87
FORMCOMP	-1.01	0.71	1.92	0.95
RUNBY	-0.60	0.72	2.60	1.20
PRIVATE	-0.70	-0.48	1.42	0.64
OUTSCOPE	-1.52	0.68	1.26	0.78
XORDFREQ	-1.11	0.37	1.09	0.78
CDCHNGE	-1.08	0.74	1.38	0.69
CDNEW	-0.98	0.67	1.01	0.75
KEPT	-1.64	0.30	0.81	0.82
ENTCLRK	-0.34	0.45	2.92	1.40

CLUSTER NUMBER: 4 M2 - Optimiser models

MEMBERS

CASE	DISTANCE
21	0.78
61	0.95
71	0.77
94	0.90
105	0.52

STATISTICS

VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
PWHATIF	-0.28	-0.28	-0.28	0.00
POPTIM	4.06	4.06	4.06	0.00
IMPORTAN	-2.16	-0.18	1.13	1.23
THREED	-0.38	-0.38	-0.38	0.00
XSIZE	-2.10	-0.50	0.89	1.02
LINKED	-0.80	-0.31	1.65	0.98
XGRAPH	-0.70	-0.09	1.33	0.81
XMACRO	-0.78	-0.78	-0.78	0.00
XLOGIC	-0.88	-0.33	1.84	1.09
FORMCOMP	-1.01	0.46	1.92	0.93
RUNBY	-0.60	-0.28	1.00	0.64
PRIVATE	-0.70	0.57	1.42	1.04
OUTSCOPE	-1.52	-0.59	0.33	0.59
XORDFREQ	-2.21	-0.89	1.09	1.08
CDCHNGE	-1.08	-0.34	1.38	0.96
CDNEW	-0.98	-0.18	1.01	0.98
KEPT	-1.64	-1.15	-0.42	0.60
ENTCLRK	-0.34	-0.34	-0.34	0.00

CLUSTER NUMBER: 5 S1 - 3D complex

MEMBERS			STATISTICS			
CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
17	0.55	PWHATIF	-0.28	-0.28	-0.28	0.00
38	0.78	POPTIM	-0.24	-0.24	-0.24	0.00
44	0.69	IMPORTAN	-0.51	0.59	1.13	0.78
		THREED	3.61	3.61	3.61	0.00
case 44 to S2		XSIZE	0.89	1.56	1.89	0.47
		LINKED	-0.80	0.43	1.65	1.00
		XGRAPH	-0.70	-0.36	0.31	0.48
		XMACRO	0.17	0.81	1.12	0.45
		XLOGIC	-0.20	0.48	1.16	0.55
		FORMCOMP	-1.01	-0.03	0.46	0.69
		RUNBY	-0.60	-0.60	-0.60	0.00
		PRIVATE	-0.70	0.71	1.42	1.00
		OUTSCOPE	-1.52	-0.59	0.33	0.76
		XORDFREQ	-0.01	-0.01	-0.01	0.00
		CDCHNGE	-1.08	-0.67	0.15	0.58
		CDNEW	-0.98	-0.98	-0.98	0.00
		KEPT	-1.64	-0.42	0.81	1.00
		ENTCLRK	-0.34	0.75	2.92	1.54

CLUSTER NUMBER: 6 Not reresented in the taxonomy

MEMBERS			STATISTICS			
CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV
95	0.50	PWHATIF	-0.28	-0.28	-0.28	0.0
106	0.50	POPTIM	-0.24	-0.24	-0.24	0.0
		IMPORTAN	-2.16	-1.34	-0.51	0.8
Case 95 to O1		THREED	-0.38	-0.38	-0.38	0.0
case 106 to O2		XSIZE	-0.10	0.39	0.89	0.5
		LINKED	-0.80	-0.18	0.43	0.6
		XGRAPH	-0.70	-0.70	-0.70	0.0
		XMACRO	0.17	0.65	1.12	0.4
		XLOGIC	-0.88	-0.54	-0.20	0.3
		FORMCOMP	-1.01	-0.28	0.46	0.7
		RUNBY	1.00	1.80	2.60	0.8
		PRIVATE	-0.70	0.36	1.42	1.0
		OUTSCOPE	-1.52	-1.06	-0.59	0.4
		XORDFREQ	1.09	1.09	1.09	0.0
		CDCHNGE	1.38	1.38	1.38	0.0
		CDNEW	-0.98	-0.98	-0.98	0.0
		KEPT	-0.42	0.20	0.81	0.6
		ENTCLRK	2.92	2.92	2.92	0.0

CLUSTER NUMBER: 7 M1 - "what if" models

MEMBERS			STATISTICS			
CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
13	0.72	PWHATIF	3.48	3.48	3.48	0.0
33	0.68	POPTIM	-0.24	-0.24	-0.24	0.0
36	0.78	IMPORTAN	-0.51	-0.10	1.13	0.7
54	0.78	THREED	-0.38	-0.38	-0.38	0.0
62	0.84	XSIZE	-1.10	-0.23	0.89	0.6
82	0.92	LINKED	-0.80	-0.34	1.65	0.8
86	0.54	XGRAPH	-0.70	-0.32	1.33	0.7
103	0.82	XMACRO	-0.78	-0.19	1.12	0.8
		XLOGIC	-0.88	0.40	1.84	0.7
		FORMCOMP	-1.01	0.46	1.92	1.04
		RUNBY	-0.60	-0.20	1.00	0.6
		PRIVATE	-0.70	-0.17	1.42	0.9
		OUTSCOPE	-1.52	-0.36	1.26	0.7
		XORDFREQ	-2.21	-0.56	1.09	1.1
		CDCHNGE	-1.08	-0.31	1.38	0.8
		CDNEW	-0.98	0.02	1.01	1.0
		KEPT	-1.64	-0.26	0.81	1.14
		ENTCLRK	-0.34	-0.34	-0.34	0.0

CLUSTER NUMBER: 8 O1 - Data entry by clerk, unimportant spreadsheets

MEMBERS		STATISTICS				
CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
19	0.00	PWHATIF	-0.28	-0.28	-0.28	0.00
		POPTIM	-0.24	-0.24	-0.24	0.00
		IMPORTAN	-0.51	-0.51	-0.51	0.00
		THREED	-0.38	-0.38	-0.38	0.00
		XSIZE	2.88	2.88	2.88	0.00
		LINKED	0.43	0.43	0.43	0.00
		XGRAPH	2.34	2.34	2.34	0.00
		XMACRO	-0.78	-0.78	-0.78	0.00
		XLOGIC	-0.88	-0.88	-0.88	0.00
		FORMCOMP	-1.01	-1.01	-1.01	0.00
		RUNBY	1.00	1.00	1.00	0.00
		PRIVATE	-0.70	-0.70	-0.70	0.00
		OUTSCOPE	0.33	0.33	0.33	0.00
		XORDFREQ	-0.01	-0.01	-0.01	0.00
		CDCHNGE	0.15	0.15	0.15	0.00
		CDNEW	1.01	1.01	1.01	0.00
		KEPT	0.81	0.81	0.81	0.00
		ENTCLRK	2.92	2.92	2.92	0.00

CLUSTER NUMBER: 9 S6 - Specialised graphical spreadsheets

MEMBERS		STATISTICS				
CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
22	0.74	PWHATIF	-0.28	-0.28	-0.28	0.00
26	0.55	POPTIM	-0.24	-0.24	-0.24	0.00
78	0.47	IMPORTAN	-2.16	-0.72	-0.51	0.54
85	0.70	THREED	-0.38	-0.38	-0.38	0.00
92	0.78	XSIZE	-0.10	0.39	0.89	0.50
93	0.64	LINKED	-0.80	1.19	2.88	1.09
100	0.78	XGRAPH	1.33	1.58	2.34	0.44
107	0.82	XMACRO	-0.78	0.41	1.12	0.63
		XLOGIC	-0.88	0.06	1.84	0.96
		FORMCOMP	-1.01	0.27	0.46	0.49
		RUNBY	-0.60	0.20	1.00	0.80
		PRIVATE	-0.70	0.62	1.42	1.02
		OUTSCOPE	-0.59	-0.25	1.26	0.64
		XORDFREQ	-2.21	0.26	1.09	1.20
		CDCHNGE	-1.08	-0.46	1.38	0.87
		CDNEW	-0.98	-0.98	-0.98	0.00
		KEPT	-1.64	-0.11	0.81	1.07
		ENTCLRK	-0.34	-0.34	-0.34	0.00

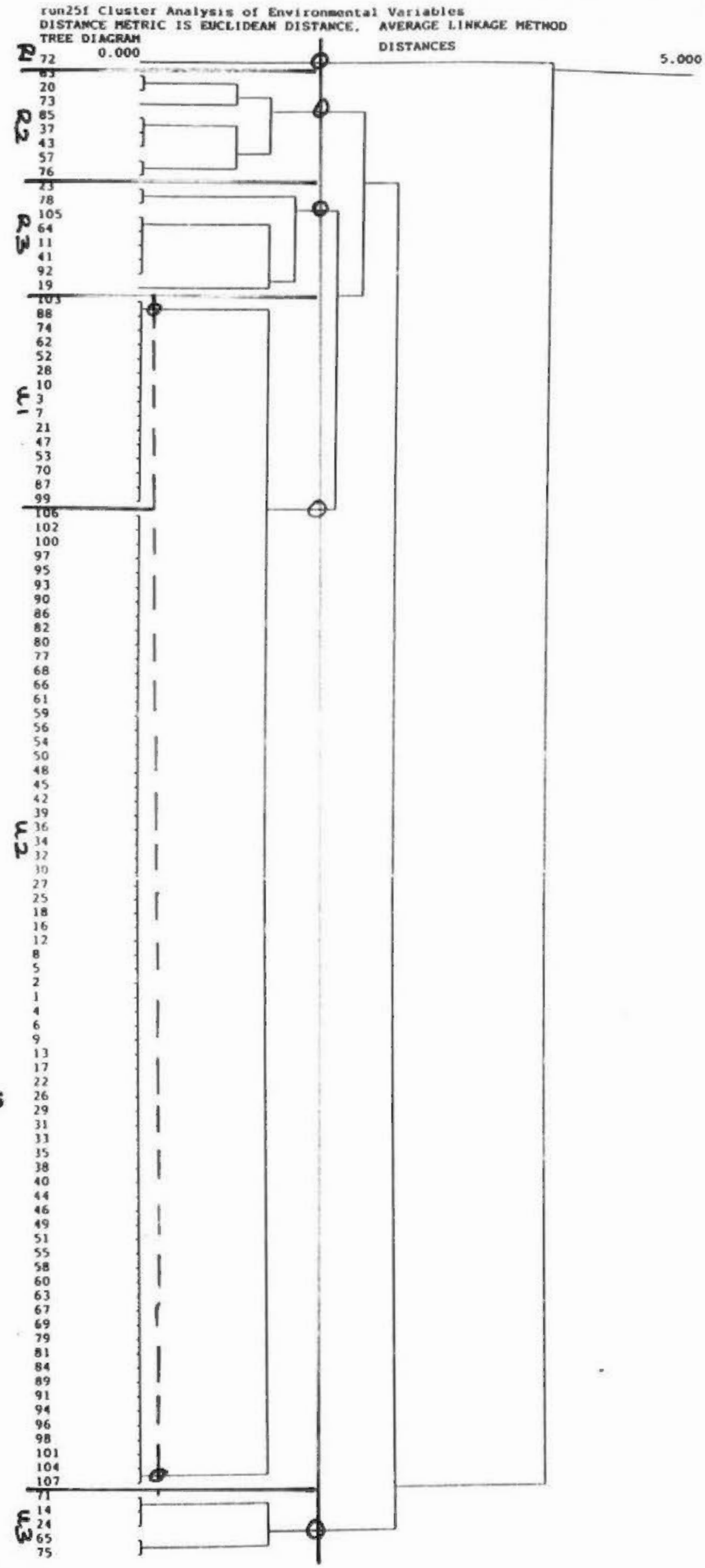


Figure 7.7

Cluster analysis

Dendrogram

Environment attributes

SYSTAT JOIN rows

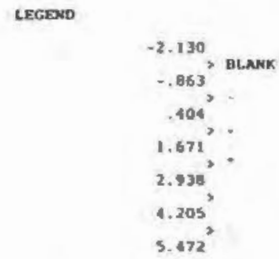


Figure 7.8

Cluster analysis

Dendrogram

Environment attributes

Shaded MATRIX plot

Table 31 **Run 25g Kmeans analysis on ordinal Environmental variables**

SUMMARY STATISTICS FOR 7 CLUSTERS

VARIABLE	BETWEEN SS	DF	WITHIN SS	DF	F-RATIO	PROB
SPERSN	105.000	6	0.000	99	.	.
ENUFTIME	96.920	6	8.080	99	197.922	0.000
SDPOLDC	88.385	6	16.615	99	87.774	0.000
SDENFORC	92.267	6	12.733	99	119.567	0.000
LIBRARY	105.000	6	0.000	99	.	.

CLUSTER NUMBER: 1 U2 - Uncontrolled development

MEMBERS / STATISTICS

CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
1	0.00	SPERSN	-0.22	-0.22	-0.22	0.00
2	0.00	ENUFTIME	0.47	0.47	0.47	0.00
4	0.00	SDPOLDC	-0.32	-0.32	-0.32	0.00
5	0.00	SDENFORC	-0.31	-0.31	-0.31	0.00
6	0.00	LIBRARY	-0.30	-0.30	-0.30	0.00

CASE	DISTANCE	CASE	DISTANCE
8	0.00		
9	0.00	56	0.00
12	0.00	58	0.00
13	0.00	59	0.00
16	0.00	60	0.00
17	0.00	61	0.00
18	0.00	63	0.00
22	0.00	66	0.00
25	0.00	67	0.00
26	0.00	68	0.00
27	0.00	69	0.00
29	0.00	77	0.00
30	0.00	79	0.00
31	0.00	80	0.00
32	0.00	81	0.00
33	0.00	82	0.00
34	0.00	84	0.00
35	0.00	86	0.00
36	0.00	89	0.00
38	0.00	90	0.00
39	0.00	91	0.00
40	0.00	93	0.00
42	0.00	94	0.00
44	0.00	95	0.00
45	0.00	96	0.00
46	0.00	97	0.00
48	0.00	98	0.00
49	0.00	100	0.00
50	0.00	101	0.00
51	0.00	102	0.00
54	0.00	104	0.00
55	0.00	106	0.00
		107	0.00

CLUSTER NUMBER: 2 R2 - Loose control

MEMBERS		STATISTICS				
CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
20	0.76	SPERSN	-0.22	-0.22	-0.22	0.00
37	0.57	ENUFTIME	0.47	0.47	0.47	0.00
43	0.57	SDPOLDC	1.93	2.77	4.17	1.09
57	0.57	SDENFORC	1.62	2.58	3.55	0.96
73	0.76	LIBRARY	-0.30	-0.30	-0.30	0.00
76	0.57					
83	0.76					
85	0.57					

CLUSTER NUMBER: 3 U3 Personal or recreational development

MEMBERS		STATISTICS				
CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
14	0.46	SPERSN	4.47	4.47	4.47	0.00
24	0.46	ENUFTIME	-2.13	-0.57	0.47	1.27
65	0.70	SDPOLDC	-0.32	-0.32	-0.32	0.00
71	0.46	SDENFORC	-0.31	-0.31	-0.31	0.00
75	0.70	LIBRARY	-0.30	-0.30	-0.30	0.00

CLUSTER NUMBER: 4 R1 - Tight control

MEMBERS		STATISTICS				
CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV.
72	0.00	SPERSN	-0.22	-0.22	-0.22	0.00
		ENUFTIME	-2.13	-2.13	-2.13	0.00
		SDPOLDC	4.17	4.17	4.17	0.00
		SDENFORC	5.47	5.47	5.47	0.00
		LIBRARY	3.27	3.27	3.27	0.00

CLUSTER NUMBER: 5 R3 - Spreadsheet library available

CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV
11	0.38	SPERSN	-0.22	-0.22	-0.22	0.0
23	0.94	ENUFTIME	0.47	0.47	0.47	0.0
41	0.38	SDPOLDC	-0.32	0.32	1.93	1.0
64	0.38	SDENFORC	-0.31	0.24	1.62	0.8
78	0.94	LIBRARY	3.27	3.27	3.27	0.0
92	0.38					
105	0.38					

CLUSTER NUMBER: 6 Not in the taxonomy

CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV
19	0.00	SPERSN	-0.22	-0.22	-0.22	0.0
		ENUFTIME	-2.13	-2.13	-2.13	0.0
included with R3		SDPOLDC	-0.32	-0.32	-0.32	0.0
		SDENFORC	-0.31	-0.31	-0.31	0.0
		LIBRARY	3.27	3.27	3.27	0.0

CLUSTER NUMBER: 7 U1 - Rushed development

CASE	DISTANCE	VARIABLE	MINIMUM	MEAN	MAXIMUM	ST.DEV
3	0.00	SPERSN	-0.22	-0.22	-0.22	0.0
7	0.00	ENUFTIME	-2.13	-2.13	-2.13	0.0
10	0.00	SDPOLDC	-0.32	-0.32	-0.32	0.0
21	0.00	SDENFORC	-0.31	-0.31	-0.31	0.0
28	0.00	LIBRARY	-0.30	-0.30	-0.30	0.0
47	0.00					
52	0.00					
53	0.00					
62	0.00					
70	0.00					
74	0.00					
87	0.00					
88	0.00					
99	0.00					
103	0.00					

APPENDIX E

GENDER BASED ANALYSES

CHI SQUARE TESTS ON DEVELOPER GENDER

Gender and Measures of Status and Training

Table 32

Spreadsheet Survey. Developer gender and employment status

	unpaid helper	employee	consultant, executive or self employed	total
women	2	9	5	16
men	3	50	37	90
total	5	59	42	106

The frequencies in table 32 were used to test the hypothesis:

H_0: There is no difference in the employment status of women and men spreadsheet developers.

χ^2 calculated was 2.755 (χ^2 critical = 5.99147, α = .05, 2 d.f.), so H_0 could not be rejected. There is no association between developer gender and employment status.

<u>Table 33</u>

Spreadsheet Survey. Developer gender and employer organisation size.

	single person	one dept	many depts one site	many sites	total
women	3	5	3	5	16
men	19	23	11	37	90
total	22	28	14	37	106

The frequencies in table 33 were used to test the hypothesis:

> H_0: There is no difference in the size of the organisations where men and women spreadsheet developers are employed.

χ^2 calculated was 0.975 (χ^2 critical = 7.84173, α = .05, 3 d.f.), so H_0 could not be rejected. There is no association between developer gender and size of the organisation for which a developer works.

<u>Table 34</u>

Spreadsheet Survey. Developer gender and qualification.

	other	degree	post grad	total
women	3	6	7	16
men	28	37	35	90
total	31	43	32	106

The frequencies in table 34 were used to test the hypothesis:

> H_0: There is no difference in the qualifications of women and men spreadsheet developers.

χ^2 calculated was 1.901 (χ^2 critical = 5.99147, α = .05, 2 d.f.), so H_0 could not be rejected. There is no association between gender and the educational qualifications of spreadsheet developers.

Table 35

Spreadsheet Survey. Developer gender and training.

	self trained	trained by work-mates	attended a course	prof. DP person	total
women	8	2	3	3	16
men	47	7	18	18	90
total	55	9	21	21	106

The frequencies in table 35 were used to test the hypothesis:

> H_0: There is no difference in the training of women and men spreadsheet developers.

χ^2 calculated was 0.391 (χ^2 critical = 7.81473, α = .05, 3 d.f.), so H_0 could not be rejected. There is no association between the gender and the training of spreadsheet developers.

Gender and Task Importance

Table 36

Spreadsheet Survey. Developer gender and spreadsheet importance.

	unimportant	moderate importance	major importance	total
women	2	9	5	16
men	6	48	36	90
total	8	57	41	106

The frequencies in table 36 were used to test the hypothesis:

> H_0: There is no difference in the importance of spreadsheets developed by women or by men.

χ^2 calculated was 0.903 (χ^2 critical = 5.99147, α = .05, 2 d.f.), so H_0 could not be rejected. There is no association between developer gender and the importance of a spreadsheet,

Table 37

Spreadsheet Survey. Developer gender and range of spreadsheet distribution

	self	one dept	many depts	ex organisation	total
women	2	6	2	6	16
men	16	27	22	25	90
total	18	33	24	31	106

The frequencies in table 37 were used to test the hypothesis:

> H_0: There is no difference in the range of distribution of spreadsheets developed by men or women.

χ^2 calculated was 1.763 (χ^2 critical =7.81473, α = .05,3 d.f.), so H_0 could not be rejected. There is no association between developer gender and the range of distribution of a spreadsheet

Table 38

Spreadsheet Survey. Developer gender and the development of spreadsheets which create corporate data.

	does not create corporate data	creates corporate data	total
women	8	8	16
men	46	44	90
total	54	52	106

The frequencies in table 38 were used to test the hypothesis:

> H_0: There is no difference in the frequency of creating corporate data in spreadsheets developed by women or by men.

χ^2 calculated was 0.007(χ^2 critical = 3.84146, α = .05 1 d.f.), so H_0 could not be rejected. There is no association between the gender of a spreadsheet developer and the frequency of developing spreadsheets where new corporate data is created.

Table 39

Spreadsheet Survey: Developer gender and the creation of spreadsheets which update corporate data

	no corporate data	read only	update allowed	total
women	5	5	6	16
men	37	30	23	90
total	42	35	29	106

The frequencies in table 39 were used to test the hypothesis:

H_0: There is no difference in the frequency of changing corporate data in spreadsheets developed by women or by men.

χ^2 calculated was 1.060 (χ^2 critical = 5.99147, α = .05, 2 d.f.), so H_0 could not be rejected. There is no association between the gender of the developer and the frequency of developing spreadsheets which alter corporate data.

Gender and Spreadsheet Technical Complexity

Table 40

Spreadsheet Survey: Developer gender and spreadsheet link complexity

	no links	links to other spreadsheets	links to other objects	total
women	11	3	2	16
men	47	26	17	90
total	58	29	19	106

The frequencies in table 40 were used to test the hypothesis:

> H_0: There is no difference in the link complexity of spreadsheets developed by women or men.

χ^2 calculated was 1.498 (χ^2 critical = 5.99147, α = .05, 2 d.f.), so H_0 could not be rejected. There is no association between developer gender and spreadsheet link complexity.

Table 41

Spreadsheet Survey. Developer gender and the use of graphics

	none	simple	intermediate	complex	total
women	13	2	0	1	16
men	52	15	16	7	90
total	65	17	16	8	106

The frequencies in table 41 were used to test the hypothesis:

> H_0: There is no difference in the frequency with which graphics are used in spreadsheets developed by women or by men.

χ^2 calculated was 4.254 (χ^2 critical = 7.81473, α = .05, 3 d.f.), so H_0 could not be rejected. There is no association between gender and the frequency with which graphics are used in spreadsheets.

Table 42

Spreadsheet Survey. Developer gender and the use of macros

	no macros	simple macros	complex macros	total
women	10	4	2	16
men	48	16	26	90
total	58	20	28	106

The frequencies in table 42 were used to test the hypothesis:

H_0: There is no difference in the frequency with which macros are used in spreadsheets developed by women or by men.

χ^2 calculated was 1.966 (χ^2 critical = 5.99147, α = .05, 2 d.f.), so H_0 could not be rejected. There is no association between developer gender and use of macros in spreadsheets.

Table 43

Spreadsheet Survey. Developer gender and spreadsheet size

	XSIZE = 1	XSIZE = 2	XSIZE = 3	XSIZE > 3	total
women	4	4	6	2	16
men	6	6	45	33	90
total	10	10	51	35	106

The frequencies in table 43 were used to test the hypothesis:

> H_0: There is no difference in the size of spreadsheets developed by women or by men.

χ^2 calculated was 12.524 (χ^2 critical =7.81473, α = .05, 3 d.f.), so H_0 was rejected. There is an association between gender and spreadsheet size. Men tend to develop larger spreadsheets than women do.

Table 44

Spreadsheet Survey. Developer gender and spreadsheet logical complexity

	xlogic =0	xlogic =1	xlogic = 2	total
women	11	3	2	16
men	39	10	41	90
total	50	13	43	106

The frequencies in table 44 were used to test the hypothesis:

> H_0: There is no difference in the logical complexity of spreadsheets developed by women or by men.

χ^2 calculated was 6.166 (χ^2 critical = 5.99147, α = .05, 2 d.f.), so H_0 was rejected. There is an association between gender and logical complexity of spreadsheets with men designing more complex spreadsheets.

Table 45

Spreadsheet Survey. Developer gender and spreadsheet formula complexity

	simple formula	complex formula	total
women	11	5	16
men	35	55	90
total	46	60	106

The frequencies in table 45 were used to test the hypothesis:

> H_0: There is no difference in the complexity of the formulas in spreadsheets developed by women or men.

χ^2 calculated was 4.931 (χ^2 critical = 3.84146, $\alpha = .05$,1 d.f.), so H_0 was rejected. There is an association between developer gender and formula complexity with men using more complex formulas in spreadsheets.

APPENDIX F

AUSTRALIAN CENSUS STATISTICS

Table 46. Preston and Australian workforce employment category statistics from 1986 census.

	Unpaid helper	Employer	Self Employed	Unem-ployed	Not in work-force	Wage or Salary	Total
Bunbury	83	605	772	1,062	7,433	7,775	17,730
Capel	43	178	314	133	900	1,152	2,720
Collie	24	108	135	353	2,790	3,207	6,617
Dardanup	39	157	268	138	1,039	1,359	2,990
Donny-brook	52	176	390	214	898	870	2,600
Harvey	93	322	520	359	2,783	2,976	7,053
Preston	324	1,546	2,399	2,259	15,843	17,339	39,710
Australia	60,690	400,159	651,234	663,148	4,788,648	5,401,432	11,965,311

Table 47. **Preston and Australian workforce educational staistics from 1986 census.**

	Degree	Diploma	Trade	Other	Not qualified	Not stated	Total
Bunbury	496	537	2,039	1,906	11,191	1,559	17,728
Capel	112	120	309	289	1,696	200	2,726
Collie	147	160	804	706	4,267	532	6.62
Dardanup	58	88	370	339	1,910	217	2,982
Donny-brook	66	91	227	291	1,737	190	2,602
Harvey	202	215	794	722	4,604	603	7,140
Preston	1,081	1,211	4,543	4,253	25,405	3,301	39,794
Australia	603,449	419,652	1,172,694	1,414,329	7,200,776	1,154,411	11,965,311

APPENDIX G

SOFTWARE USED

SOFTWARE USED IN THE PREPARATION OF THIS THESIS

The working environment for this thesis used consisted of an IBM PS/2 SX running DOS 3.3 and Microsoft WINDOWS 3.0 and Hewlett Packard Laserjet III and Cannon Bubble jet "Squirt" printers.

- The thesis document was prepared using Lotus Samna Ami Professional version 2.0 with font enhancement provided by Adobe Systems's Inc. Adobe Type Manager
- The graphs were prepared using Samna Ami Pro., SYSTAT Inc.'s SYGRAPH and Microsoft EXCEL for Windows
- Other graphics prepared using Microsoft Windows Paintbrush, Microsoft Powerpoint for Windows and Samna Ami Pro.
- Data collection instruments prepared using Microsoft Word for Windows.
- Data storage, validation and transformations using Enable Software Inc.'s ENABLE OA, database, SQL and spreadsheet modules and Microsoft Excel for Windows.
- Statistical analyses using SYSTAT Inc.'s SYSTAT.
- Literature abstracts managed using Enable Software Inc.'s ENABLE OA database and word processing modules.