SOFTWARE PSYCHOLOGY AND THE COMPUTERISATION OF THE WEIGHTED APPLICATION BLANK

A thesis presented in partial fulfillment of
the requirements for the degree of
Master of Arts in Psychology
at Massey University.

Josephine Ann Smith

1986

ABSTRACT.

This study investigated the use of a Weighted Application Blank (WAB) for selecting candidates likely to pass the first year of a comprehensive nursing course. A subject pool of 415 comprehensive nursing course applicants was drawn from 1980 to 1985 first year Polytechnic classes. A discriminant analysis on the application form responses made by these subjects was performed. Computer software was then developed incorporating results from Human Factors research. The software aimed to computerise the WAB method of classifying applicants following principles of software psychology. A group of 50 computer naive subjects participated in an experimental evaluation of the software. Five subjects took part in initial pilot study trials of the software. The remaining 45 subjects' were divided into three equally sized groups. The subjects task was to enter eight sets of nursing course application form data. The "computerised" group received instructions on how to do this from the screen, the "written" group from a manual and the "verbal" group verbally from the experimenter. Time taken to complete the task and the number of errors made were recorded. Three ANOVAs were performed to establish if group exerted an influence on trial times or error rates. In addition, applicants were required to complete two questionnaires. The first prior to the experimental trials and the second following them. Results indicated that group influenced time taken on the task ($F(1,294) = 7.43$, $p<.001$). Group did not exert an influence on errors made on each question

$(F(32,672) = 1.022,\ p>.05)$. The interaction between errors made on each application form and group was significant $(F(14,294) = 2.809, p<.001)$ however the main effect for group of this comparison was insignificant $(F(2,294) = 0.045,\ p>.05)$. Responses to the questionnaires were evaluated and an assessment was made of the responses. It was concluded that the fields of human-computer interface design and personnel selection had been successfully combined. Leading to the expectation that an area of great research potential had been opened up.

ACKNOWLEDGEMENTS

At the outset of my University career my mother was sure I would return home within a month having decided that University was just not my "scene". Six years later I have at last concluded my studies, this thesis being the final "chapter". Needless to say there are a fair number of persons to whom I owe my thanks. To the following people however I am particularly indebted.

Mike Smith my thesis supervisor and at various stages lecturer, gets to head the list. It was he after all who waded through the many drafts of this research and who always managed to keep me at my work. No mean feat given that much of the writing was done during the summer months. He also kept me up to date with Karls latest exploits.

Shona Butterfield, Head of Nursing Studies at Manawatu Polytechnic at the time the research was conducted. Her support and encouragement throughout the project is sincerely appreciated.

Dave George for his enthusiastic willingness to provide ideas and assistance to me at various times throughout the research. He too managed to "dob" Karl in a few times.

Alison Stewart for assisting with the administration of the experimental tasks. She was as amused as I was by Karls problems with the micros.

My graduate classmates also provided support and entertainment (not always intentionally eh Karl?). In particular

Sue Henderson - with whom I spent countless weekends at Massey working on our respective theses.

Sharon Rippin - another weekend worker who also warned whenever it was not the "best" time to go and see Mike.

Karl Pajo - a weekend worker in spirit rather than practice. We had many arguments and despite what he may think he seldom won.

My flatmates, Sandy, Clare and Hami, also warrant a mention. After all, they lived with me throughout the writing of the thesis. I'm sure there were times they wondered who it was they saw for half an hour each evening. I'm positive they wondered if I'd ever get it finished.

Finally, my family. Thanks to my parents for their constant offers of help. If I could have found a way to employ them I would have. Also thank you to L..... S... (the family cat) for sending me letters recounting both her and the families exploits.

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

TABLES

FIGURES

CHAPTER ONE

GENERAL INTRODUCTION

Overview.

The recent phenomena of widespread computer use means that many occupations now routinely incorporate computers. Tasks as diverse as the monitoring of a chemical plant and intermediate school education are two examples. It seems a small step therefore to suggest that the area of personnel selection should also be looking at the computerisation of some of its functions.

How this computerisation is carried out is something that few researchers have directly addressed. Techniques for personnel selection have long been discussed, as have guidelines for designing systems for interactive use, but how to design software to aid in the selection of people is a new area.

Personnel psychology is concerned with fitting people to jobs. Ideally then, organisations wishing to fill vacancies will use a technique, promoted by personnel psychologists, in the hope of eventually hiring an individual capable of performing the job. In practice selection does not often follow this ideal. Seldom is an Occupational Psychologist employed, instead reliance is placed on techniques found to exhibit dubious reliability and validity. Interviewing (Arvey, 1977) and testing (Muchinsky & Tuttle, 1979) are particular favourites, work samples (Robertson & Kandola, 1982) and trainability tests (Downs, 1977) are two alternatives. Use of work samples and trainability tests is often limited however due to the expense of administering them.

One promising method of selection is the Weighted Application Blank (WAB). The WAB offers several features other techniques lack. Most important is its low cost and ease of use. The WAB is based on information gleaned from the application form an organisation uses for recruitment. Often therefore the introduction of WAB processes mean that little additional effort or money must be allocated by the organisation. Using application forms filled out by past applicants it is possible to identify characteristics that distinguish between successful and unsuccessful job performance (England, 1971).

Human Factors engineering, also known as Ergonomics, is concerned with "fitting the job to the person". Rouse (1979) defines the field as addressing issues related to the design and evaluation of the interface between people and machines.

Too often Human Factors practitioners are involved in solving problems with existing person-machine systems. It would seem to make more sense for systems to originally be designed using Human Factors principles. The person-machine system to be examined here is the human-computer interface. This interface presents something of a challenge to Ergonomists, as the individuals for which the computer interface must be designed are extremely diverse in their expectations of and expertise in computer use. From information presented it is hoped it will be possible to design software appropriate for use by novice computer users.

Considering the large numbers of applications many organisations sift through each time they wish to fill a vacancy, it seems a logical step that some parts of this process become computerised. The present study aims to computerise a specific selection method. The measured success of this adaptation will indicate whether this approach is appropriate. Further, it is intended that the software eventually designed will be usable by anyone, this includes individuals with no computer experience. Having set the goal of developing software for novice computer users it becomes necessary to acknowledge the particular needs of this group. Human Factors guidelines should help in fulfilling this goal.

The present study must address issues and describe the main techniques of personnel selection. It must also discuss the popular methods of computer software design. The marrying of these two areas of Occupational Psychology must then be evaluated through a study of both the predictive success of the selection method chosen and the most easily used version of the selection software.

CHAPTER TWO

REVIEW OF THE SELECTION LITERATURE.

Popular Selection Techniques

In an article tracing the ancient history of testing, DuBois (1970) describes selection, with the use of tests, of government officials in China around 2200 B.C.. The problem of fitting people to jobs has apparently been of concern for quite some time.

Growing interest in selection procedures was in evidence earlier this century, Hollingworth in 1923 evaluated some of the more "interesting" methods. The graphological assessment of handwritten application forms, perusal of unstandardised letters of recommendation, chaotic interviews and application of physiognomy techniques to submitted photographs seemed the order of the day. Needless to say the predictive efficiency of these methods was no more than two or three percent above chance.

A number of improvements since the 1920's have helped establish personnel selection as an important task in many organisations. The two world wars played major roles in stimulating growth in the personnel selection field, with techniques developed for use during these times, still in use today. Testing, methods of job specification, job knowledge tests, officer rating forms and training programs were just a few of the developments attributable to the wars (Uhlaner, 1976).

Personnel selection revolves around the belief that "not all individuals are equally well suited for all jobs" (Landy and Trumbo, 1980). This being so, factors that distinguish between those suited and those not suited to a job need to be found. Once identified (usually through job analysis), elements able to differentiate between the two groups can be combined into a test or sample of behaviour. This can then be used to help identify promising job candidates before they are placed in a job.

Job analysis is an important aspect in the development of selection techniques. It aims to describe the parts of a job that distinguish it from other jobs. In essence job performance is broken down into small measurable units thus facilitating the design of tests (Guion, 1961). Blum and Naylor (1968) describe nine different methods of analysing a job. Amongst these are the questionnaire method - a worker is asked to respond to questions about a job in writing, the individual interview - a person is questioned in detail about the activities involved in the job, and the observation interview - where the individual is asked about their job as they perform it. In research situations, job analysis often provides the information base from which the criterion measure for the study is drawn.

The selection of a criterion variable is instrumental in determining the direction or emphasis of research. Thus data collected is devoted to the description of the criterion. In selection research the criterion chosen will often be length of tenure. High criterion (probably high tenure) and low criterion (probably short tenure) groupings are sought with subjects examined for attributes that can distinguish them as being a member of one group or the other (England, 1971). Whatever criterion is decided on the entire study is dependent on its adequacy and accuracy.

If success in a job is the criterion, what methods are available to help ensure that people selected for a position will succeed?

Testing is one of the more established techniques. Tests developed earlier this century were primarily designed for the measurement of intelligence. Binet and Simon (1905) and Terman (1917) were the early workers in this area. Soon, those concerned with the selection of people for jobs picked up on the use of tests as one way of differentiating between applicants (Muensterberg, 1914).

The First World War saw the first extensive use of aptitude testing. The emphasis was still on the measurement of intelligence, however the numbers that could be tested at one time increased from one to up to five hundred (Hull, 1928). The Second World War saw a revitalisation of the testing process with some movement into specific aptitude testing occurring (for example, with tests for mechanical aptitude and eye-hand co-ordination).

Testing came in for some criticism during the 1960's, with psychologists questioning the almost indiscriminate use of tests in situations where their applicability had not been established. A call was made therefore for the job-relatedness of tests to be examined (Gross, 1962; Whyte, 1956). Fair testing and employment practices became part of many countries legislation and testing had to abide by these legal requirements.

The large number of tests available today, lets employers select tools appropriate for specific employment situations. Categories of tests include power tests, speed tests, group tests, individual tests, pencil and paper tests, performance aptitude tests and achievement tests. Buros (1978), in a frequently updated Mental Measurements Yearbook, documents not only the available tests but also provides, where possible, research data on the reliability and validity of the tests.

Research on testing is one way test users are able to discriminate between good and bad tests. Muchinsky and Tuttle (1979), review the empirical literature relating to test score predictors. The research considered in this study had research with tenure as the criterion, and tests of personality, interests, intelligence and aptitude as predictors. Diverse results were obtained in each testing area with positive, zero and negative correlations between predictor and criterion variables apparent. The authors had to conclude that situational factors were influencing the findings. It can be concluded therefore that tests need to be validated for the specific use to which they are to be put.

Realistic tests of tasks similar to those performed on the job represent one category of the vast testing area that has experienced increasing popularity in recent years (Schmidt, Grenthal, Hunter, Berner and Seaton, 1977; Mount, Muchinsky and Hanser 1977; Robertson and Kandola, 1982). In an effort to reliably represent a job to an applicant and to gain an indication of their likely performance on the job, work samples have been developed. Work samples gauge an applicants performance on a task similar to the one they will be required to undertake on the job. A minimum of instruction is given and behaviours exhibited during task completion are noted. The candidate is assessed according to their behaviour during the test as well as the output from the test, be it a sewing machine sample (Downs, 1977) or responses to problems in an in-basket test (Lopez, 1966). Work samples have been used for some time now with varying results. Asher and Sciarrino (1974) review validity research conducted on the work sample. The results are generally encouraging, validity coefficients higher than .3 being obtained in approximately 70% of the studies. A 1968 study by the

Jewish Employment and Vocational service of Philidelphia revealed that work sample testing resulted in nearly twice as many successful job placements (40%) than did counselling only situations (23%).

A more recent extension of the work sample idea has occurred with the development of trainability tests (Downs, 1977). Trainability tests are similar to work samples in that some set task is performed. In this case however, the actual trainability of the applicant is being assessed. The individual is permitted to ask as many questions as they require to successfully complete the job assigned. With the aim being to evaluate the persons likely success during training for a job.

In the case of both work samples and trainability tests key elements of the intended job are tested. Both therefore require that resources be used to complete the test. Both also need to be administered, in ideal situations, on a one to one basis, although groups of up to four persons have been tested successfully (Robertson and Mindel, 1980).

The interview is arguably the most frequently used selection technique. Actual methods of interviewing vary from an unstructured, non-directive approach to highly structured panel interviews where a predefined set of questions is asked.

Research on the interview has consistently concluded that the reliability and validity of the technique is insufficient to warrant its use as a selection tool (for example, Arvey, 1977; Wright, 1969; Landy, 1976). Yet organisations continue to make use of it. Latham, Saar, Pursell and Campion (1980) pursued the use of situational testing combined with interviewing in an effort to improve the validity of the interview. Thus a structured interview based on systematic job analyses was developed by the authors. In an interview setting, job candidates were asked how they would behave, given certain situations. Their responses were rated against those of previous good and poor employees with selection made in light of the comparisons. A predictive validity of .33 was obtained. While most research questions the use of the interview, given the low reliability and validity scores obtained, the technique may serve some purpose. Landy (1985) suggests the interview be used for 1) communicating information about jobs, 2) clarifying applicants questions, 3) obtaining information from candidates, and 4) maintaining good public relations. In combination with other selection techniques therefore the interview may be serviceable.

The interview has been incorporated into what is now one of the most comprehensive selection packages, the Assessment Centre. Such centres make use of a variety of popular selection tools plus some specifically designed group dynamics exercises. Job applicants can therefore spend a number of days at the centre performing tasks as diverse as paper and pencil tests and leaderless group discussions. Finkle (1976) presented a comprehensive review of the development of assessment centres. Early data on the predictive validity of the technique was seen as promising. However concern has been expressed about this selection method. Dunnette and Borman (1979) question the quality control of assessment procedures. Further reservations were voiced regarding the reliability and validity of the ratings made of assessees (Hinrichs and Haanpera, 1976; Klimoski and Strickland, 1977). Primarily disquiet surrounds the fact that instead of future performance, assessment actually concentrates on identifying those individuals who fit the company "mould".

The above selection techniques are costly in both time and money.

Combine this fact with their unsatisfactory predictive efficiencies and the need for alternative selection methods becomes clear.

The Weighted Application Blank.

Owens (1976) in a comprehensive discussion of background data and it's use as a personnel selection tool, views biographical information as providing a picture of where an individual "has been" i.e. a record of the individuals life to date. Owens argues via the concept of behavioural consistency (Wernimont and Campbell, 1968) that a persons past experiences (and their responses to them) can aid greatly in predicting their likely directions in the future. To lend support to his assertions Owens quotes Galton (1902) "The future of each man is mainly a direct consequence of the past.....It is therefore, of high importance when planning for the future to keep the past under frequent review".

England in 1961 made the first concerted attempt at defining and describing the Weighted Application Blank (WAB) as a selection tool. This early publication was updated in 1971.

The assumptions behind WAB development were defined by England as being -

1) that past behaviours can help in the prediction of future behaviour.
2) that certain aspects of an individuals background can be related to their successful performance in a job.
3) that there is a need to be able to determine which aspects of a persons background are important in the performance of a certain job.

4) that it is possible to combine certain aspects of a persons background for the prediction of likely performance in a given occupation. Each application form item can therefore be assigned a numerical weight corresponding with its predictive power.

The WAB aims to establish which application form items represent attributes important to successful performance and provides a systematic method for assessing if candidates possess them.

Typically a WAB will combine two types of information. First, demographic or situational information will be requested. For example, age, sex, number of dependants and educational qualifications. Such items are easy to verify and simple to answer. Secondly, items referring to preferences or attitudes will be asked. In this case the questions are not as straight forward. For example, applicants might be asked to give their reasons for applying for a position, or to give an opinion on an issue relevant to the work they will be doing. These items require some evaluation by the applicant before they can be answered.

During the past decade or so, items defined as 'biodata' (an abbreviation for biographical data) have been incorporated into the WAB. Biodata items include questions such as "Were your parents happy with your school work?" or "Were you viewed positively by your high school teachers?". Clearly biodata items include information somewhat removed from the job itself. That a relationship exists between responses to biodata items and, for example, success on the job, or long term tenure is evident from research results (Freeberg, 1967; Wells and Muchinsky 1985; Neiner and Owens 1985).

Research investigating biodata and related data types (for example, biodemographic data; Federico, Federico and Lundquist, 1976) will be considered alongside studies of more uncomplicated life history items. All refer to the past for the source of their information. Indeed Guion (1967) makes the point that any item type can be tried when developing a WAB, only one point needs to be kept in mind and that is that responses can be classified in some way. As part of the process of developing a WAB, empirical analysis will result in the thinning out of the number of items included or actually weighted in the final application blank used.

Methods used in weighting the WAB

In general terms the development of weights for the application blank is achieved with a systematic method of determining which application blank responses were given more frequently by applicants who proved to be desirable employees. At the same time the responses needed to be given less frequently by applicants deemed to be less desirable.

Until recently no research comparing the predictive powers of the different WAB methodologies had been undertaken. An as yet unpublished article by Smith and George (1985) addresses the question for what seems the first time. These authors compare differential weighting techniques (horizontal and correlational analysis) with multivariate weighting techniques. Of the two approaches, differential techniques have tended to be the more popular in the past. In a selected bibliography presented by George (1983), 23 of the 31 studies cited used one of the differential methods.

The horizontal percent method takes the responses of members in each group and compares them to each item. The groups are designated as past successful and unsuccessful using a previously selected criterion. Whenever a difference in item response for the two groups is noted, a weight is assigned to that item which reflects the positive/negative relationship the item has to the criterion. A new applicant has their score calculated by summing the item weight by item response values.

The correlational analysis method uses the correlation co-efficient, calculated between each item and the criterion to weight the item. An individuals score is calculated by summing the item weight by item response values.

Multivariate weighting techniques go further than differential weighting techniques in that item to item relationships as well as item to criterion relationships are investigated. The two most popular multivariate methods used are regression analysis and discriminant analysis. When used with the WAB each yields a slightly different result. Regression analysis identifies the unique contribution each application form item makes to the prediction of the criterion. That is, each independent variable that explains a statistically significant portion of the dependent variable is identified. Future applicants have their responses 'run through' (Lawrence, Salsburg, Dawson and Fasman, 1982) the regression equation. As a result an individuals likely performance with respect to the criterion can be gauged. In the Lawrence et al (1982) study the average tenure of managers at the time of hire was predicted. By using past managers responses on a WAB as the independent variables and the length of tenure as the criterion, a regression equation was calculated. Subsequent applicants had their WAB item responses run through this regression equation and an indication of their likely tenure was obtained.

Discriminant analysis, while following similar statistical steps, effects a different description of the criterion. The same steps as regression analysis are followed in that a criterion is selected. A common example is success in a job. WAB item responses are collated for people who were both successful and unsuccessful on the job over a previous time span. The variables are linearly combined to ensure that the criterion groups are as statistically distinct as possible. The analysis results in a discriminant function being calculated. Some of the original items will have been dropped from the analysis as they would be found to contribute insignificant amounts of information to the description of the criterion. The responses of future applicants to the items found important in describing the criterion will be entered into the discriminant function equation, the score obtained will classify the new applicant into either the successful or unsuccessful criterion group.

The weighted application form compared to other selection techniques.

Smith (1978) compared the predictive power of a WAB with that of a selection interview. For a sample of 107 recently hired factory workers, interviewer predictions correctly identified 20% of those who eventually left within three months. The WAB was able to successfully predict in 80% of the cases, that an employee would leave within three months. For a company concerned about the cost of a high turnover rate, these results suggest that sole reliance on the interview when selecting employees was not the most effective use of resources. Smith, in his final conclusions, suggested that the use of a WAB combined with an information giving interview would seem a sensible and cost cutting solution.

Dunnette and Borman (1979) attempted a comprehensive summary of the modern era of personnel selection. While not assessing biodata and/or WAB's in comparison to other techniques some important points are made by these authors. After reviewing issues significant to selection practices, such as validity, statistical considerations and job performance measurement, these authors discussed the research results from several different selection practices. One method covered was that of biographical information. Dunnette and Borman support WAB's in their prediction of job performance but point out that they have been criticised for failing to establish the nature of predictor-criterion relationships. Work by Owens (1976) and Schoenfeldt (1974) is seen as refuting the criticism by viewing biographical information as a record of an individuals life thus supporting the assumption that past behaviour is the best predictor of future behaviour.

Reilly and Chao (1982) conducted a direct comparison of eight selection procedures seen as alternatives to testing. Alternatives were sought following suggestions that tests may result in an adverse impact on racial and ethnic minorities. When the comparisons were complete, only biodata was found to be as valid or more valid than tests and to result in minimal adverse impact. The seven other techniques (peer evaluation, interviews, self-assessments, reference checks, academic achievement, expert judgement and projective techniques) were unable to match either test or biodata methods in terms of their validities and their levels of adverse impact.

Advantages and Disadvantages of WAB Use.

From the preceding discussion it can be seen that the use of the WAB includes both advantages and disadvantages. On the negative side is the requirement for large (100 plus) subject pools when developing and cross validating a WAB. Perhaps this is not such a problem when it is considered that those using or wishing to use WAB's for selection purposes have mostly been large organisations experiencing high turnover. Where this is the case a large subject sample is relatively simple to obtain.

The success of the WAB in predicting performance has also been challenged (Schwab and Oliver, 1974). These authors report that most studies yielding negative results are "buried" by their investigators and that journal editors reject a disproportionate number of articles reporting non-significant findings. However this argument must be equally applicable to any personnel selection research or indeed any research in psychology.

WAB methods have also been criticised for failing to explain the relationship between weighted variables and the criterion. Clearly it is important to establish why some variables are useful in predicting, for example, tenure and others are not. The recent use of multivariate statistical methods has gone some way in reducing this problem (George, 1983). In particular discriminant analysis demonstrates predictive efficiency comparable to more common differential weighting techniques. At the same time discriminant analysis offers descriptive capabilities not offered by alternative weighting methods.

On the positive side the WAB offers a low cost, simple to use aid to the selection process. Indeed most organisations would find that their existing application form(s) required little adaptation. Reilly and Chao (1982) describe how the use of application forms by many organisations means little effort or cost is involved in first, expanding the forms already used so that they include biodata items and, second, performing an analysis of the data to yield weighted items. Employers have however expressed some concern about the accuracy of application form responses. Cascio (1975) investigated the relationship between reported and verified responses to 17 biographical information blank items. His findings indicated that the overall responses were strikingly accurate. Further, the information supplied on a WAB need not only be used in the calculation of WAB scores. Such data can be used in initial screening of applicants (i.e. determining that minimum hiring requirements are met) as well as providing supplementary information prior to the interview.

Proposed use of the weighted application blank.

George (1983) evaluated two WAB weighting techniques using as his sample intake students for a comprehensive nursing course. The present study builds upon these results. WAB's have been used successfully in a variety of situations. In the present case a Polytechnic run nursing course is experiencing an unacceptably high attrition rate. Improvement of the courses selection procedures was seen as a way to reduce the number of students finding themselves unable to complete the first year. Limited financial resources, large numbers of applicants to be screened and accessible application form data, meant the WAB was viewed as the most suitable alternative. As in any selection situation it is never intended that the WAB scores obtained be the sole selection determinants.

In the present study therefore, a WAB will be used to help predict a nursing course applicants likely success on the course after one year of study. For the final decision other methods will also be used. Predictions will be based on the characteristics of the previous five years applicants. Information from these applicants records will be weighted using discriminant analysis. This method has been selected as it not only identifies variables that are significantly related to the criterion but also provides useful descriptive information.

CHAPTER THREE

REVIEW OF HUMAN-COMPUTER INTERFACE LITERATURE.

During the past ten years an increase in interest and effort in computerising personnel systems has occurred (Short, 1979; Walker, 1980). This is due in part, to the growing sophistication of the personnel function, the relative decrease in the cost of systems and the need for fast and up to date personnel information.

Generally the elements of a personnel department which are computerised have been employee databases and payroll schemes (I.P.M., 1982). There is however no reason the function could not be extended to the selection of new employees. Computers seem particularly suited to the provision of a fast, easily updated service of applicant screening. The WAB as outlined above lends itself particularly well to this. Not only is the data from application blanks simple to enter and manipulate, with appropriate software the calculation of applicant scores for comparison purposes is easily accomplished.

The adaption of a selection process from a manually performed to a computerised operation requires some planning. The field of Human Factors (Ergonomics) should play an important role.

The Human-Computer Interface

The aim in writing software is always to break the design task into easily handled modules (Kemp, 1982). For a WAB, results from initial weighting techniques will indicate which application form items are required and will provide values with which the item responses can be weighted. Any software would thus need to request the responses to the relevant items, weight them and finally indicate whether an applicant shows positive or negative attributes when compared with previous applicants.

The composition of computer users as a group is becoming more and more diverse. No longer is computer use the sole preserve of computer professionals, instead individuals from all walks of life are coming to computers with the intention of speeding up and simplifying their work. These users are often naive about computing: the computer is essentially a mystery to them.

The Role of Human Factors in the Design of Software.

Human factors (Ergonomics) is the scientific study of how to design work areas for use by those assigned to work within them. It is the study of person-machine relationships. This concern to improve the work place for safe, pleasant use can be applied to the general design of computer software. Such a requirement is not however easy to implement.

Shneiderman (1979), describes just how difficult the implementation of Human Factors guidelines can be. The diversity of computer users is a major problem area. Regular users rapidly progress beyond their initial fascination with a system, coming to demand more and more of it. Systems therefore need to accommodate the various knowledge levels of a wide variety of users. They need to be easy to learn as well as providing help to those unfamiliar with them. How to arrive at the best possible solution for software design can be assisted by following a Human Factors approach.

Novice Users: A Special Group.

A result of the drop in the cost of computer hardware and increases in software availability and computer power is that a microcomputer is now more easily obtained by a wider group of people. Many users are therefore unlikely to have experience or training in computer technology. Researchers working in the area of human-computer interaction have referred to this group of individuals as "casual" users (Martin, 1973), "general" users (Miller and Thomas, 1977) or "naive"users (Kennedy, 1975). No distinction between these various definitions has been made so they will be taken to be synonyms and used interchangeably throughout the ensuing discussion. As a group, these users present some perplexing problems for software designers.

Eason (1976) provides a useful definition of the naive user. Principally they are people with no expertise in computer technology but who use a computer system as an aid to performing a task. This has been taken to mean that they have received no formal computer system training and probably do not desire such training. No matter how technically sophisticated a programme is, if it does not perform the tasks the user requires it to perform, the system as a whole will be evaluated in a negative light. Also, naive users generally require and desire only minimal levels of computer technology knowledge and skill, they will therefore seek to minimise the time and effort devoted to using the computer system.

The final general attribute Eason presents describes the apparent "at risk" status of naive computer users. As a results of naive users limited knowledge, unexpected responses that have occurred because of a system malfunction or a user error can be experienced as traumatic and may colour the users attitude to the system for some time.

In summary, novice users need to minimise the effort of learning and using a computer system. Their wish for "ease of use" coupled with an "at risk" status engenders problems for software development. In other words novice users need a system that is easy to use and learn, that meets their task needs and that also protects them from the system.

Conceptual/Cognitive models of the Naive User.

Naive users are discerning users. If a system is difficult to use it will not be used. Thus designers must cater to a group of clients who will quickly reveal dissatisfaction with a programme through discontinued use of it. The designers are thus being forced to take note of the user population. More effort is now being expended in establishing some basic facts about naive users.

Too often software designers write programmes that are based on their own intuitions of what a novice user needs. It is proposed here, that as a first step, an understanding of the thought processes of the naive user is crucial in the design of software.

In interactions with computers, a user "interface" exists between the designers conceptual model and the users cognitive model of the system. The user gains their impression or cognitive model of the system via the interface. Saja (1985), in a discussion of the human computer interface, reports how a cognitive model is developed by an individual interacting with another person or with a machine. Building on work by Winograd in 1977, Saja describes how the users model of the discourse is developed during the interaction. A user will therefore constantly be searching for information about the computer so they can reach an internal understanding of the system. This cognitive model is subsequently utilised to produce and understand statements in the interaction, it may also be called upon in the future when users find themselves in similar situations. The interface takes on some importance for the programme designer therefore, in that it must be presented in such a way that the user gets the impression of the system intended by its designers. Where the conceptual and cognitive models match up, the cognitive model will be a representation of the systems capabilities and limitations. If the designers conceptual model and the users cognitive model do not coincide the result will be frustration for the user who is expecting the system to perform in a way it has not been designed to do. Frustration has been found to reduce user performance (Shneiderman, 1979) and hence both the human and computer resource will be under utilised.

Recognising the importance of users having conceptual models of the systems they use, Jagodzinski (1983) saw naive users as a group requiring a particularly clear understanding of the system they use. This concern for novice users arises from Jagodzinskis' understanding that their lack of computer technology knowledge could lead to early disdain of an ambiguous system.

Having acknowledged the importance of the naive users need to develop a sound cognitive model of a systems capabilities several authors (eg. Shneiderman, 1980; James, 1981) describe ways a designer can go about incorporating features in their software to ensure that this will take place.

Interface Design and Presentation.

Just how to introduce a user to a new computer system and then train them in the use of that system are topics long debated by system designers (Mayer,1967; Nickerson, 1969; Kennedy, 1975; DuBolay and O'Shea, 1981). Both these problems are addressed by research investigating the best methods of computer system representation.

Too often, according to Jagodzinski, systems designers rely on their own experience and intuition when writing a programme, Mumford (1980) cautions against such an approach. Instead the system developers are advised to follow guidelines established on theory rather than instinct. In this way the assumptions behind a particular mode of practice will not remain hidden. Understanding and knowledge of why certain approaches work will conceivably lead to fewer errors occurring during the design process.

Just how the system should be, and indeed can be represented to novice users is also addressed by Green (1980). This author asserts that novice users need to be provided with an overview of the task the programme is performing. The aim being to ensure that users do not "think" they know why the programme works, but that they "know" they know why it works.

Saja (1985), also describes how a novice user needs to form a cognitive model of the system before actually interacting with it. Otherwise they are likely to attempt to fit a previous cognitive model to the situation as they lack a relevant one. An essential part of the human-computer interface is therefore an initial off-line written introduction to the programme. The aim is for this off-line portion to provide a valid, broad foundation for the users cognitive model of the system. Both the off-line and on-line parts of a programme represent the interface for that software. After continued interaction with both portions of the software, the users cognitive model should be reinforced and extended. The user's model evolves during their interactions with the system. A system designer needs to recognise the value of the off-line/on-line interface. By using it a designer can help to ensure that a prospective user of a system will generate a true model of the system, its capabilities and its limitations.

A programme incorporating a unified, off-line and on-line interface, will conceivably be simple to modify should the need arise. In changing any aspect of such an interface, the designer has the relatively simple task of tracing through the programme identifying problem areas.

Another advantage of an interface made up of off-line/on-line portions, presented as a unit, is the opportunity for programme consistency. Consistency aids in the acquisition of an accurate cognitive model, leading to a user successfully utilising a programme. Inconsistency on the other hand, can severely shake a users faith in a system. The user will lose confidence in their cognitive model not knowing any longer what a system can or cannot do.

Other approaches to interface design and presentation have entirely different foci from that expounded by Saja (1985). Indeed, James (1981), found that users did not usually read through comprehensive documentation when it was provided as part of the interface. Rather, they expected the information required to operate the system to be available to them on the screen. Mayer (1967) proposed the concept of the automated training system, where the aim was to develop training tutorials that were self-instructional and which presented material in small incremental steps, requiring frequent responses from the trainee. In a discussion of the benefits of such a training system, Mayer cited the results of a study where the subjects were found to learn the system by actually using the system. The positive results of learning to use a system successfully worked as a motivator to the users to continue in their familiarisation with the system. Kennedy (1975), also suggests that the training of novice computer users be provided by the computer itself. He asserted that it is imperative that the trainee not be antagonised by the system to be learnt, principally because users will probably be put off using a system that they do not like. However Kennedy did state that describing the systems structure and describing terminology clearly should also improve the performance of naive computer users. Presumably Kennedy has recognised the importance of a valid and reliable representation of the computer interface.

From the above there appear to be two distinct schools of thought when it comes to the best way to design an interface for a novice computer user. On the one hand there are those advocating recognition of cognitive models as developed by users when interacting with computer systems. For example Saja (1985) supports the use of interfaces that rely on both an off-line and on-line section. The initial introduction to the system provided is seen as being the foundation of an accurate representation of the programme. On the other hand, there are those wishing to see interface presentation designed to be performed by the system itself. Such a method promotes confidence in and familiarity with the system in question.

The interface presented to users is an important aspect of software design and there are specific factors that must be addressed to ensure that the interface eventually presented is as friendly as possible.

Specific Design Requirements of Software Intended for Novice Users.

Most of the problems experienced by system designers when implementing software could be avoided had these computer professionals consulted with those their work was designed to serve. Several authors have prescribed goals for the system designer to work by (for example, Foley and Wallace, 1974; Engel and Granda, 1975; Wasserman, 1973). Researchers such as Hansen (1971) or Pew and Rollins (1975) specifically state that the user population must be known. Indeed Shneiderman (1979), states that "know the user" should be the motto of every software designer. It is only through enhanced knowledge of the user that their true attitudes and anxieties will be understood. A designer who can take the gist of the research findings and combine them into a workable interface, thereby meeting the needs of the novice users, will be producing software that can be used by a broad group of individuals.

Psychological Issues In Software Design.

Research has shown that user attitudes can dramatically effect learning and performance with interactive systems. This has been touched on above. Walther and O'Neil in a 1974 study of user performance found that novice users with negative attitudes towards computers learned an editing task more slowly and made more errors than those with positive attitudes. One reason given for the decrease in performance was that anxiety leads to a reduction in short term memory capacity (a trait that is essential in interactions with a computer).

Walther and O'Neil go on to describe situations that could conceivably lead to anxiety. Thus, feelings of insecurity about using a computer arising from insufficient orientation to the system, presentation of overwhelming amounts of detail that has to be assimilated or pressure to work too rapidly for the users level of familiarity with the system, were factors identified as possibly provoking anxiety. Clearly mild pressure can serve as a motivator for many individuals, if this pressure is permitted to become too strong the resultant anxiety is likely to interfere with the competent completion of the job. Eason and Damodaran (1981), lent support to the importance of user attitudes when interacting with the computer. Negative attitudes were again seen as detrimental to performance, while positive attitudes, especially in the early phases of interaction were seen as likely to instill in the user the confidence to deal effectively with any small problems they encountered. By implication then, systems designers would do well to create computer software that promotes positive attitudes.

Shneiderman (1980), identifies the need to control the environment as another psychological trait of computer users. The strength of this desire will vary from person to person with many individuals not showing much concern for their fate. Novice terminal users will be quite content to initially follow the instructions issued by the software they are utilising. They accept the computer as the controlling agent in the interaction. With frequent use of a programme the user will quickly become frustrated by a lack of perceived control. The computer becomes a tool which the user will resent if it starts to send messages that indicate the computer is in charge. This change in emphasis is a design challenge for software developers. The programme needs to be able to adapt to the individuals level of expertise, dropping the "leader" role as soon as the individual is competent in the basic workings of the programme.

Desire for closure is another aspect in a persons make-up that software designers would do well to recognise. Closure describes the relief felt when a task is completed successfully. Computer users are likely to seek closure in their interactions with the computer. An attempt therefore needs to be made to design a programme in sections. As the user finishes each section they will be rewarded with the relief felt at achieving closure.

Lastly, James (1981) identified a need for privacy amongst novice users. Adult users became inhibited and self-conscious when onlookers were present. Indeed these users came to use the computer only when they were unlikely to be observed by others. Clearly such a situation is unacceptable to both the user and employer, indicating that some attention to ensuring a user can have privacy is needed.

Design Principles Aiming to Maximise Human-Computer Interaction.

Mayer (1967) asserted that part of the novices difficulty in working with many systems was that their design made them unapproachable. Software has been developed in such a way that it, and computers in general, are shrouded in a mysterious complex of jargon and sophisticated displays. Because of this criticism early in the development of software, Shneiderman (1979) recommends that effort be given to the design of software that helps the user feel at ease. This needs to be done however without being patronising or too obvious.

A suggested step along the way to improving the approachability of interactive systems is the use of natural language (Petrick, 1976). Briefly, the aim of natural language is to construct software that can accept instructions in English. An attempt is made to provide several synonyms for each word so as to eradicate the need to memorise a set of acceptable commands. Hill (1972) seriously questions the wisdom of such a move however. His concern stems from an awareness that the English language can be extremely ambiguous. It is pointed out that with a natural language system the impressive speed, storage and accuracy of computers will be bypassed. Instead long and tedious clarification procedures would have to be designed in an effort to ensure that a users meaning was fully understood by the system.

One aspect of software design where the use of language is important however is in the provision of acceptable error messages. Shneiderman (1980), is probably the main proponent of work in this area. He discusses how the error checking and handling components of software often present the programmer with their major design task. Naive users require a greater programming effort than any other group. For this group, entries made that do not conform to those expected must be rapidly identified and diagnostic messages provided. The user will then be able to easily locate and correct their mistakes. Messages should be brief, positively worded and constructive. Shneiderman suggests that instead of condemning the user with messages such as 'ILLEGAL SYNTAX', it makes more sense to attempt to indicate where the error is and how it can be fixed.

The optimum time to issue an error message is also an area of some debate. Segal (1975) found that performance improved when messages were issued immediately rather than after a RETURN sequence or even later. Disruption to thought processes caused by this fast identification of an incorrect entry did not appear detrimental to task completion. An advantage of such an approach was the simple replacement of the incorrect character. Having identified that an error has been made, the problem of providing the user with the right sort of information occurs. The expertise of the user must in this case be kept in mind. An experienced, frequent user will probably be able to find and correct an error when they receive an indication that a mistake has been made. Thus a locked keyboard or special character is probably enough to trigger a response in them. Users making occasional use of the computer probably require a brief note at the top or bottom the screen to remind them of the proper syntax of an entry. Novice users however are unfamiliar with the semantics of a system and are therefore likely to require more than brief prompting on the necessary syntax. Instead full explanations of the options available or the set of possible commands are needed. It is unlikely that even the expert user will retain a working knowledge of every portion of the system. To cater for the inevitable lapses in memory, HELP facilities are usually incorporated into the design of interactive software.

Probably the best approach to the provision of HELP facilities is to allow the user to decide when help is required. Eason and Damodaran (1978) found that as users become more familiar with a programme their needs in terms of which interface mode they require, changes. Thus they may no longer desire a verbose, self-explanatory level of instruction. By implication the software must be able to accommodate the various levels of expertise likely to be found among the software's projected user population. This is an important point in that one of the major characteristics of the novice user population is its widely differing competence levels. It has been suggested that the user be able to request various amounts of HELP information. For example, the typing of one '?' would yield a one or two word prompt on the appropriate responses, two '??', a brief reminder and three '???', a full description of the type of information required. Tagg (1981), picked up on this idea and postulated that HELP facilities would be more effective if they took the users level of expertise into consideration. In the example above therefore, the novice user would probably require a full description of the information required each time they make a mistake. More experienced users will probably resent the time taken to output a lengthy description accompanying a typographical error.

A similar problem to that involved in choosing an error message format occurs when selecting an appropriate interface mode. Martin (1973) identified three potential formats, menu selection, fill-in-the-blank or parametric. Menu selection is the most straight forward of these, in that a set of numbered choices are provided from which the user is expected to select one. (See Figure 1).

```
READY FOR HIGHEST SCHOOL QUALIFICATION:
1) SCHOOL CERTIFICATE
2) SIXTH FORM CERTIFICATE
3) UNIVERSITY ENTRANCE
4) BURSARY

?_
```

FIGURE 1: Menu selection example.

Little or no training is required and the user receives an idea of the range of system features. Further, the limited number of options available in any one menu frame means the chance of error is minimised.

Fill-in-the-blank interface modes require that the user is familiar with the range of responses acceptable to the system (See Figure 2). The user is presented with a brief prompt, after which they need to enter the requested information. Some training may be required, however experience with the format over a relatively short period of time should result in users becoming proficient. Clues about the type of information needed can be provided, for example 'CASH DESIRED ($10 TO $200 IN $10 UNITS)'. Should users not be able to recall

```
READY FOR

YEAR AT UNIVERSITY ? second

NUMBER OF PASSES   ? 8

MAJOR              ?_
```

FIGURE 2: Fill-in-the-blank example.

the exact formatting conventions they must be able to request more information. Finally, a parametric interface is designed for those frequently using the system. A detailed knowledge of acceptable input formats is essential. (See Figure 3). Such a system is extremely fast and user satisfaction is high due to the amount of control over the system that exists. Again it is essential that the user be able to request more information should they require it. With careful design a system could cater to a variety of users with diverse levels of computing experience. Shneiderman (1980) laments the dearth of relevant research in this area. He suggests the need for research to clarify which combinations of applications are suitable for the various sets of potential users.

```
ACCOUNT   ENTER/SEATON,PASSWORD/ROBYN,FUNDS/500
          ENTER/FRANKS,PASSWORD/JEFF,FUNDS/200
          ENTER/GRAHAM,PASSWORD/JILL,FUNDS/700
          ENTER/BROOKS,PASSWORD/PAUL,FUNDS/600
          ENTER/?_
```

FIGURE 3: Parametric mode example.

External Factors Likely to Affect Computer Use by Naive Users.

Eason (1976) is concerned with the influence of external factors that are likely to upset the smooth utilisation of a computer system by naive users. Personal factors such as those outlined above were seen as important, but so too were specific task factors and job/role factors. Task factors imply the requirements of a users job, in that the tasks the user is expected to complete exert a strong influence on the kind of service the user needs from the software. Job/role factors also effect the acceptability of a programme. Software requiring a substantial commitment in terms of time to learn the system will not be acceptable to any one who works in a time constrained job, the benefits are not consistent with the costs.

Within the task related factors, Eason identified information needs, task structure and frequency of use as important to the naive user in the completion of a computer task. The specific job related factors seen as important were the relationship of the job to the system, the status of the user and the presence of a technical language in the users job.

Software Design and Testing.

Unfortunately the design and introduction of software in recent years has not been as thorough as it could have been. Gould and Lewis (1985), present an approach to system design that embraces three important principles. These are

1) an early focus on the user population (through recognition of some of the above factors).
2) empirical measurement of the initial prototypes.
3) use of an iterative test-retest cycle aiming to ensure that programme changes are tested out before being accepted into the design.

These three steps warrant elaboration. The early focus on the intended user population is one way designers can be certain that they are developing a programme that will meet the needs of its users. The cognitive, behavioural, anthropometric and attitudinal characteristics of the users as well as the nature of the work to be performed should all be noted. Empirical measurement of the software early in its development can be achieved by having a sample of users implement it. These subjects should be encouraged to make comments, their reactions and performance can be recorded and analysed. In this way problems with the software are identified early and are problems based on the reactions of actual users rather than the suspicions of system developers. Iterative design, as stated above, helps make certain that any changes made are meaningful, in that they precipitate an improvement in the performance of the software.

Gould and Lewis (1985) go on to discuss the actual methods employed by designers. Despite stating that this three tier approach to system design was obvious and made sense, five groups of systems designers and planners failed to in fact implement such an approach within their work. The 447 subjects were asked to provide the sequence of five or so steps that they would typically follow when developing and evaluating a new computer system. Each persons responses were graded by three judges, a point was earned for any mention, however remote, of the three steps outlined above. Only 16% of the respondents included all three steps in their methods, 24% mentioned two of the steps, 35% mentioned one step and 26% did not mention any of the steps. Gould and Lewis concluded that even after stating that the design principles presented seemed straight forward and logical, designers rarely applied them in their work. A number of reasons were posed as ways to explain this anomaly. Included amongst these were that the principles were not worth following, that user diversity was overestimated and users did not really know what was required. Gould and Lewis refute the many criticisms pointing out, for example, that designers cannot judge the user population if they do not have ongoing contact with it. Having made this point these authors reiterate the importance of following the three principles. If designers truly wish to serve, or indeed, be able to compete to serve, the interests of a growing pool of new computer users they will have to recognise the requirements of that group.

Another aspect of interface design was addressed by Mozeico in a 1982 study. In response to the perceived diversity of user experience, Mozeico suggests that the human computer interface should be designed to accommodate several user learning stages. A first time user is seen as benefiting from software that is able to present information in easily digested blocks. It is suggested that the software be partitioned in a way that reflects the various learning stages of users. The user is therefore spared the frustration of trying to deal with an overloaded short-term memory. Treu (1977) identified four learning stages in a study of users working with an interactive graphics package. A users needs at any one of these stages are different to their needs at any other stage. Mozeico found that Treu's four stages did not cater for users who were not motivated to progress from one learning level to the next. By adding one stage of his own therefore, an ideal of five learning stages was arrived at. The results of subsequent research to establish the effectiveness of software designed to cater to users with varying amounts of experience were encouraging. Not enough data was collected however for them to be conclusive.

As is the case with many of the articles published that address issues in the design of the human-computer interface, Yestingsmeier (1984) presents several important ideas but has not backed up his assertions with research evidence. In this case however, the ideas seem particularly relevant to the present research. Yestingsmeier points out that it is becoming more and more widely accepted that Human Factors considerations need to be kept in mind when designing interactive computer systems. Three phases for the design and implementation of software systems are suggested. The three phases incorporate some of the principles stipulated by Gould and Lewis (1985), however the emphasis is slightly different. Yestingsmeier's three phases include 1) a study phase, 2) a design phase and 3) a development/operation phase.

The study phase involves consulting with the end user to ensure that the problems of their job are understood by the system designer. The design problem can then confidently be defined. Having reached an initial proposal of features for the final design, the designer again consults with the end users. In this way designers can verify that they have understood exactly what these end users require.

The second phase is the design phase where system designers are encouraged to recognise the various levels of competence apparent in the user group. A list of 15 design objectives is presented, among these are suggestions for the inclusion of "HELP" facilities, meaningful feedback, easily understood error messages and uncluttered screen presentation. The next step within this phase is the establishment of testing guidelines. That is, a method of gauging whether the programme is both syntactically and semantically correct. Throughout the second phase the system designer is encouraged to remain in contact with the end user group. Considering the amount of time it can take to complete the initial design, a programmer must be aware of any changes in the intended software setting.

Finally the third phase is the development/operation phase. At this stage in the softwares development, final testing and documentation of the system takes place. Documentation is seen as essentially making or breaking a system. A system where time is wasted searching through inadequate or poorly written documentation will quickly be spurned by the user population. Thorough testing of each section of the programme should occur followed by rigorous testing of the system as whole. Once both the on-line and off-line components are readied for final use, intended users can be introduced to the system and early training performed. It is doubtful that the system will achieve full approval even at this late stage, so any code and document changes should be made.

The training of naive users mentioned by Yestingsmeier has been researched by Kennedy (1975). In this study the best method of presenting a new system to naive users was addressed. Each subject's attitude to computers was assessed. Those found to be antagonistic or indifferent towards computers were put in one group, those receptive to computers in another. Each of these two groups was again divided into two groups. One group were given manuals that explained the system to be learnt, the other group were expected to use the system itself to solve any problems they encountered. The relative success of these two methods of naive user training were then tested using a data entry task. It was found that user attitude had only a small affect on the users performance. The point was made however that attitude did seem to be important to the subjects initial acceptance of the system. For this reason therefore it was important that the method of training did not antagonise the user.

Which of the two methods of system training (use of a manual or online) was the better remained a point of debate with there being no significant difference between those subjects given a manual and those who were not. It was noted however that of those given the manual only three subjects had read it before the trial. It was these three subjects who were ranked in the first three positions. The point was made that while the sample was too small for the result to be considered seriously, preparation appears to make a positive impact on performance of a task. Thus during preparation for a task it seems imperative that major terms are defined and the capabilities and limitations of the system outlined.

Another interesting outcome of this research was the finding that interaction with the machine provided many subjects with self-teaching opportunities. Where self-teaching consisted of using the systems guidance and error indication facilities. With these features at hand, Kennedy describes how they combine to help build naive users confidence in their abilities. The reason, the person/computer relationship is transformed from one where the machine guides the user, to one where the user guides the machine. Prother, Berry and Bermuday (1972) were able to show that the opportunity for users to learn by trial and error with feedback from the computer gives significantly improved performance than prompting or demonstrations.

Reliability and Validity of Software.

Any user is entitled to expect that the software they use has been tested by its developers to a stage where the possibility of malfunction is as near to zero as possible. A programme where breakdowns are found to occur inconveniently frequently can be expected to be spurned by its targeted users.

Software designers are therefore obliged to ensure that their developments are usable. Indeed, software reliability should be such that the software becomes indispensable to the user population. It would be unrealistic to call for perfection. The rapidity with which things change makes a perfect system impossible to attain. Llewelyn and Wickens (1968) suggest that designers aim to have the system usable by "average" users. Unfortunately the "average" user is not defined by these authors making catering to the average user difficult to do. If however "majority of users" becomes the target group, a realistic goal is set.

Designers must be constantly aware of the need for reliability. The increasing growth in computer use by many diverse groups of users coupled with bigger, more complex systems has led to the design of many unreliable systems. In recent times between one third and one half of the effort that goes into the development of software, has been spent on testing and debugging. Considering this huge commitment of resources to software development, unreliability is an expense a developer can ill-afford.

A number of methods for producing more reliable software have been outlined by Kopetz (1976). The discovery of a design methodology aiming to eradicate activities such as testing, debugging and run-time error treatment is one proposal. Clearly such a software development technique would save enormous amounts of time, money and effort. It is however doubtful that a design methodology itself would result in suitably reliable software. Thus, secondly, there is a place for testing and debugging. It is assumed that extremely thorough testing and debugging can lead to reliable software. Currently most effort in software design is expended on the testing and debugging of software yet only a fraction of all input cases can in fact be executed. A probable consequence of this is that reliable systems remain as elusive as ever. Finally, with the inclusion of redundancy, the reliability of a system can also be increased. Redundancy is necessary for the detection and correction of faults. Where a system contains more resources than are required to fulfil the systems task, it is said to contain redundancy. During the design phase completely different alternatives for a portion of the task may be identified. A system can employ active redundancy, where redundant components take an active part in normal system operation, or standby redundancy, where redundant system parts are only switched on when the active parts of the system drop out due to a failure. This last method of reliability improvement is distinctly different to the previous two methods, in that it is assumed that a system will contain errors.

In the present research the issue of redundancy is not as important as it might be in a situation where large and complex systems are being used. It is proposed therefore that two of the three methods outlined by Kopetz, will form a basic structure for the development of software in the present project.

The validity of the programme is as important as its reliability. Validation in this context refers to the effective testing of programmes. Traditionally software validation has been an exercise based on intuition, occurring at the end of the overall software design process. In recent times it has become evident that the cost of an error found during the closing phases of software design is far greater than an error found early in the programmes development (Howden, 1981). Fujii (1981) describes a software development process in which validation is part of each phase of the process and not carried out after everything else has been done (See Figure 4).

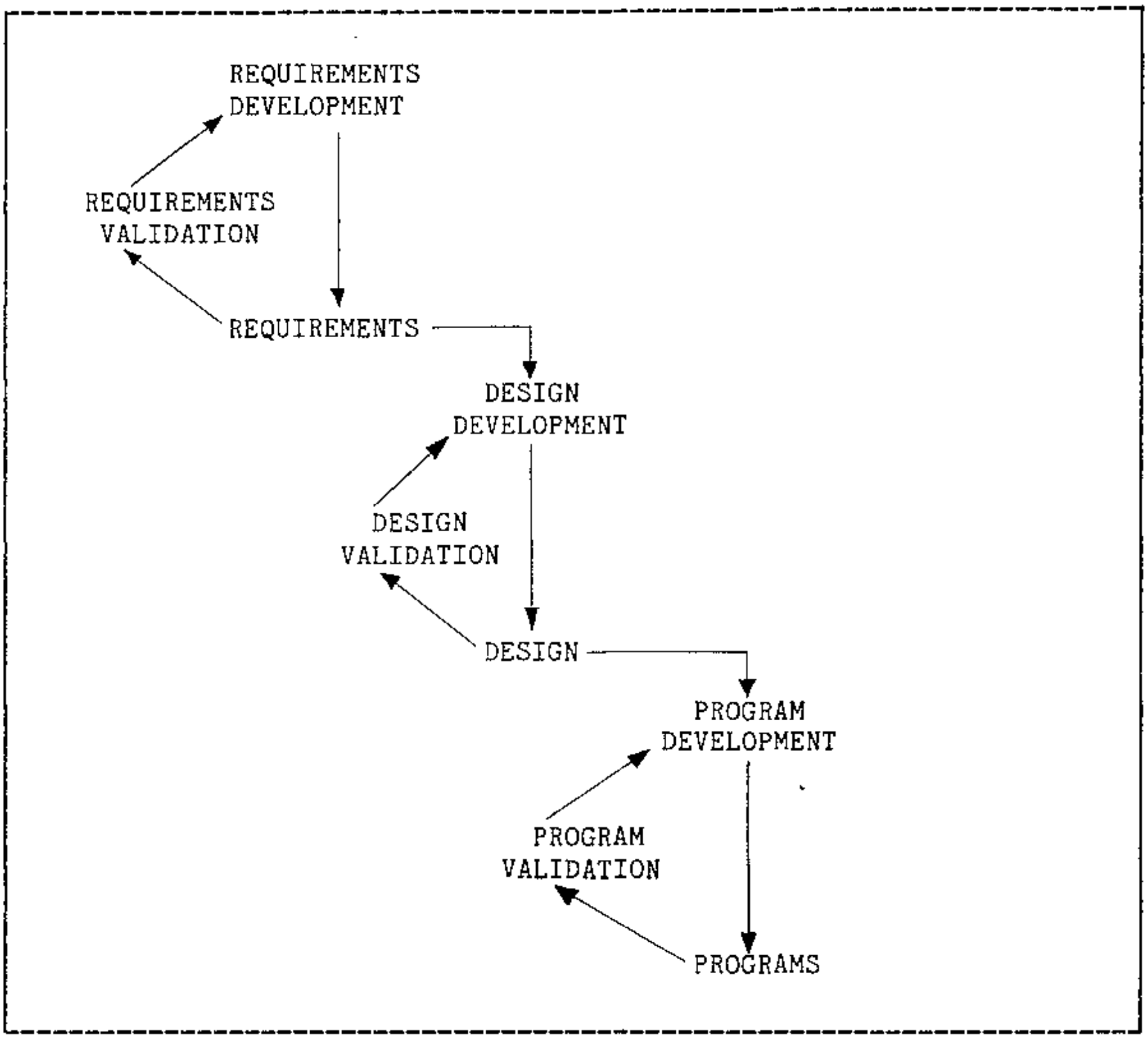


FIGURE 4: The integration of validation into the software development process (Fujii, 1977).

Thus validation occurs during establishment of what is required of the system, the design of the system and the coding of the design of the system.

Howden (1981) describes how two types of validation activities can be carried out, static and dynamic analysis. Static analysis involves the analysis of the system through referral to the system documents, actual execution of the system is not therefore necessary. Important properties such as consistency and completeness are evaluated by checking the "requirement" and "design" documents. Dynamic analysis does require that the system be executed. The most important dynamic analysis technique is program testing. Again the requirement and design documents are of importance, as it is from these that the set of test data is drawn.

If serious in their intention to provide users with reliable programmes, programmers need to incorporate validation procedures into the overall design. The programme must run smoothly and the documentation accompanying it must therefore be understandable by all users.

CHAPTER FOUR

AIMS, OBJECTIVES AND HYPOTHESES.

Literature Review Summary

From the various reviews of personnel selection practices (Dunnette and Borman, 1979; Ash and Kroeker, 1975) it becomes clear that few techniques exhibit predictive validities commensurate with the frequency with which they are used. A theme that consistently arises is that the most successful methods are situational (Landy and Trumbo, 1980). That is, where time and effort are expended on discovering which elements in a job are measurable and, which elements also differentiate between candidates likely or unlikely to succeed in a job.

The WAB was identified as one technique that yields as good if not better predictive information when compared with other methods (Reilly and Chao, 1982). It is this approach that is to be researched in the present study. While the problem is not one of selection for a job and prediction of tenure, it is very similar. Selection of candidates for a comprehensive nursing course is required, with a need to ensure that candidates will pass the first year of the course successfully.

The polytechnic for which the software is to be developed utilises a number of micro-computers. The eventual entry of all student information into these computers suggested that a computerised selection tool would be the most successful. The WAB procedure appears to lend itself particularly well to such an approach, being made up of a number of steps easily translated into modules for programming.

In any programming effort it is essential to follow human factors guidelines. Authors have found to their chagrin, that through failing to consult with the user pool for which their software was designed needless delays and revisions to their work resulted.

Many guidelines for software development have been presented (Shneiderman, 1980; Segal, 1975; Martin, 1973) following research in what is a growing field. The programme developed needs to incorporate these findings into its final form to ensure the people it is intended to serve find it usable.

Other issues to be investigated include the identification of induction and training methods found most satisfactory by users, recognition of different user learning stages (Mozeico, 1982), the various presentation formats (Kennedy, 1975) and on-line/off-line documentation preferences (Prother, Berry and Bermuday, 1972). The assessment of these and users written reactions to the software need to be sought.

Objectives.

The present research can be split into two quite neat sections. The literature reviewed evidences this.

From the information on selection methods it appears that

1) WAB's represent a simple, low cost method of selection. In many situations the information is already available, from for example, existing application forms. (Smith, 1978).

2) The validity of biodata (and by implication WAB's) is superior to all other selection methods tested. These other techniques include interviews, self assessments, peer evaluations, tests and references. (Reilly and Chao, 1982)

3) Multivariate weighting techniques have been found to be at least as valid as more popular differential weighting methods. Further, multivariate techniques provide added information about the relationship of items to the criterion. (George, 1983)

Typically the WAB is constructed after detailed study of the job for which a vacancy exists. Thus a job analysis would be performed to establish what skills and/or traits distinguish those performing the job from those employed elsewhere. The results of such analyses provide an initial indication of what items need to be included within the WAB (England, 1971).

Owing to the nature of the present study it was not possible to perform such an analysis of the work to be performed. First, as no direct contact with students in the course took place, an observation of their performance was not possible. It was only negotiated with the Nursing Course Head of Department to have access to the files of the students, not the students themselves. Still, some insight into the content of the course was gained through conversation with course tutors. A further constraint on the ability to replicate the suggested steps was the reliance on the past five years application forms. Unfortunately the need to have the work done rapidly and with sufficient numbers to make the results worthwhile meant existing information had to be used.

The major findings relating to software development and testing were as follows

1) It is imperative to recognise the attributes and needs of the

targetted population. (Shneiderman, 1979)
In the present study novice users are targetted. As the literature reveals, this group has been the focus of some detailed research and is seen as the user of tomorrow. (Eason, 1976; James, 1981; Dubolay and O'Shea, 1981)

2) Attention should be given to assuring that the cognitive model developed by users is one that promotes full understanding of a systems capabilities and limitations. (Jagodzinski, 1983)
This may require that both off-line and on-line documentation of the system be provided. (Saja, 1985)

3) Specific suggestions for interface features were made (Shneiderman, 1980). The impact of these features on user performance has seldom been assessed. A call was made for researchers to address this issue. (Kennedy, 1975)

4) Guidelines for software design and testing were given by several authors (Gould and Lewis, 1985; Mozeico, 1982; Yestingsmeier, 1984). Again a call was made for software developers to adhere to the steps described in an effort to standardise software development. Such standardisation would also lead, by implication, to improved reliability.

5) The need for valid and reliable software is also seen as important to software development. As in the case of all tools used in psychology, high validity and reliability should be pursued.

Aims

1) To find which items in a currently used application form provide information that can distinguish between those who are likely to succeed in a comprehensive nursing course after one year of study and those who are not.

2) To be able to predict which applicants in a sample of applicants are likely to succeed in a nursing course. Predictions will be made using WAB techniques.

3) To develop software that can use the results of WAB analyses to quickly assign future applicants to "likely to succeed/likely to fail in the nursing course after one year" groupings.

4) To write the software in a way that acknowledges the importance of the human-computer interface. Thus suggested design features will be incorporated into the software.

5) To develop software following methods that promote high levels of reliability and validity.

6) To identify the interface mode that results in the best performance by naive users.

Hypotheses

1) That information in an application form when combined using discriminant analysis methods can distinguish between individuals who will succeed and those who will not succeed after one year of study in a comprehensive nursing course.

2) That <u>on-line</u> software designed to incorporate human factors features, for example, provision of help and error messages, will result in more superior performance on a data entry task by novice users than will written or verbal help.
Performance was to be assessed via both time taken and error rate for the entry of each applicant form.

3) That the items selected for use in the programme would incur similar numbers of errors. Thus no one item would be any more difficult (or easy) than any other item. In this way the programme will have met the requirement of user friendliness in that the user task is straightforward and clearly understood.

CHAPTER FIVE

METHOD.

Overview and Design

The present study was conducted in two phases; the first analysed six years of applicant information to find which variables distinguished between candidates likely to succeed and candidates likely to fail the first year of a comprehensive nursing course. The second sought to discover the effects of mode of information presentation on subjective estimates and speed of implementation of the programme.

The first phase involved discriminant analysis of coded application form data, while the second led to three oneway ANOVA's of mixed design being performed as well as an analysis of descriptive questionnaire data.

Subjects

Two pools of subjects were used in the present study, see Table 1.

TABLE 1

Summary of Subject Detail.

	Total Number	Age Range	Mean Age	Number of Females	Number of Males
Applicant Differentiation	415	16-47	19 yrs 3 mths	393	22
Software Study	45	20-45	30 yrs 6 mths	41	4

The application forms of four hundred and fifteen successful comprehensive nursing course candidates were used in the development and testing of the WAB. The sample consisted of 22 men and 393 women with an age range of 16-47, the mean age being 19 years and 3 months. Not all forms submitted were complete, where information was missing, an attempt was made, by letter, to obtain the missing information. Despite these efforts 98 of the original 415 subjects could not be used in the analysis. Further, a subsample of 80 of the 317 subjects with complete data sets, was not used within the analysis but kept as a holdout sample. These subjects were used to test the weightings from the discriminant analyses. Holdout samples are essential in research of this type (England, 1971). Through being independent of the test, they provide a gauge of the true effectiveness of the WAB.

The second group of 50 subjects was used in the software development phase of the research. Subjects were a mixture of polytechnic tutors, students and office workers. Five of these subjects provided initial feedback on the software in its development and pre-test stages. The remaining 45 subjects took part in the experiment itself. Ages ranged between 20 and 45 with the mean age being 30 years and 6 months. Forty-one of the subjects were female and four were male. All subjects were computer novices. Novices being defined as people who had not used a computer over extended periods of time. Twenty-eight, however had some minimal contact with a computer, though none for more than a few hours.

The pool of 45 test subjects was divided into three groups of 15 subjects. Individuals were assigned to groups randomly.

Statistical Methods

Discriminant Analysis

Linear discriminant functions aim to predict a categorical criterion variable from a continuous set of predictor variables (Tatsuoka, 1971).

By taking several variables and combining them, discriminant analysis seeks to obtain a single dimension which separates subjects into two distinct groups. This is done by forming one or more linear combinations of variables,

$$\text{i.e. } D_i = d_{i1} Z_1 + d_{i2} Z_2 + \ldots + d_{ip} Z_p$$

where D is the score on discriminant function i, the d's are weighting coefficients and the z's are standardised values of the p discriminating variables used in the analysis (Cooley and Lohnes, 1971). The maximum number of functions that can be derived is either one less than the number of criterion groups, or equal to the number of discriminating variables. In the present study only one discriminant function will be derived.

Each individuals predicted score is derived from a weighted linear composite of the predicted variables. "To obtain categorical predictions, the beta weights used to develop the weighted linear composite are applied to the mean predictor scores of each of the criterion groups. This procedure yields a weighted linear composite of means for a group, or a group variate score, which is called a centroid" (Weiss, 1976). Each criterion groups centroid is calculated and can be plotted on a straight line (discriminant). An individuals discriminant score is then calculated and compared with the centroids of the criterion groups. A candidate is then classified as a predicted member of the criterion group whose centroid is closest to their discriminant score.

Predicted and actual criterion group memberships are compared in a classification table, with an overall "hit-rate" expressed. That is, the proportion of actual criterion group members who are correctly predicted as being members of that group. In the present study then, it is possible to ascertain the number of students predicted to succeed on the nursing course who actually did succeed, and the number of students predicted to fail who actually failed.

The present study incorporates several nominal variables. While there is debate on whether the technique should include the use of such data, Lachenbruch (1975) points out that the robustness of discriminant analysis means the technique can easily accommodate this data type.

In interpreting discriminant analysis outputs, the following features need to be considered.

Beta weights, or more correctly standardised discriminant function coefficients, serve to identify the variables that contribute most to the differentiation of the two groups. Large beta weights indicate variables that distinguish the criterion groups well. Thus these variables will also be likely to have the largest mean differences between criterion groups on the predictor variables. Where this is not the case, two predictor variables are usually highly inter-correlated with each other.

When the sign of the standardised discriminant function coefficients is ignored it is possible to work out the relative contribution of the variable to the function. The sign denotes whether the contribution is a positive or a negative one. The functions can also be "named" by identifying the dominant characteristic measured by variables associated with it.

Eigenvalues and their associated canonical correlations are output for each function and denote the relative ability of functions to separate the groups. Wilks Lambda values (and associated chi-square tests) indicate the amount of discriminating power that exists in the variables used before each function is derived. The larger the lambda value, the less discriminating power is available. The chi-square value indicates whether significant amounts of information remain to be explained. An immediate indication of the efficiency of predictor variables if given therefore, as the first Wilks Lambda value output corresponds to the amount of discriminating power that exists in predictor variables before any has been removed in the first function.

The group centroids also provide information on the success of the discriminant function(s). By comparing the group means for each function it is possible to tell how far apart the groups are along that dimension.

The classification of cases is achieved through using a series of classification functions, one for each group. The classification of cases can be adjusted to reflect the relative group sizes. In the present study, membership in the successful group is far more likely than membership in the unsuccessful group. This will be accommodated for in the analysis.

Where large numbers of predictor variables are to be used in the prediction of a criteria, the researcher is able to select the most useful of these variables via a stepwise method. Stepwise procedures begin by selecting the single, best discriminating variable and entering it into the discriminant equation. Subsequent discriminating variables best able to improve the value of the discrimination criterion when combined with a previous variable or variables are then chosen.

Both forward and backward stepwise methods are possible. A backwards stepwise procedure is employed when most of the predictor variables involved in the analysis contribute significantly to discrimination. At each step, variables that add little to the discriminant equation are removed. Usually a minimum F probability value is stipulated, variables with probabilities above this value are discarded, largest first.

ANOVA

In very general terms, this method is designed to establish if one group of subjects differs significantly from another group. Analysis of variance is usually appropriate when the groups of observations are created by using a nominal level variable as the independent variable of the study.

In the present case then, the trial group represents the nominal variable, this variable has three categories implying three groups of observations. The task is to determine whether these three groups differ significantly in the average level of performance on the dependent variable. Number of errors and time taken to complete a task will be used as dependent variables in the various ANOVA's to be calculated.

The analyses to be performed will all be oneway, where significant effects exist, the results will be decomposed in order to identify their source.

Materials

Introductory Questionnaire.

Morrison (1983) conducted research into the attitudes to computers of 412 students at an Australian University. Following factor analysis of the results he found that attitudes fell into four distinct categories - negative attitudes, awesome (1), awesome (2), and an application attitude.

The first factor, negative attitude, describes feelings such as, computers will result in unemployment and will dehumanise some tasks. Fears were also harboured concerning computer reliability and power over the lives of individuals. The second and third factors, awesome

(1) and awesome (2) respectively, both reflected the view that computers are awesome machines. In awesome (2), the awesome machine impression resulted from an inadequate understanding of computers evidenced by astonishment. The last factor, "application", seemed to reflect concern about the real positive and negative implications of widespread computer applications.

An open ended introductory questionnaire aiming to gauge subjects attitudes to computers was developed for this study (see Appendix 1). The information collected sought to source a users feelings toward and their expectations of the computer.

The Software.

The programme developed for the research aimed to computerise the classification of individuals according to the attributes of previous applicants found successful and unsuccessful in a Comprehensive Nursing Course. See Appendix 2 for a complete printout of the programme. WAB techniques of weighting candidate attributes (described earlier) were to be used to achieve this aim. Multivariate procedures were used in this classification process.

Taking the standardised discriminant function coefficients output from the analysis, two scores were calculated for each candidate. The greater of the two classified the applicant into either the successful or unsuccessful group. The programme developed first calculates these scores and then compares them. The result is a classification of subjects into likely to succeed and likely to fail groups.

To perform these tasks the programme needs to obtain the application form responses to items found to discriminate between the two criterion groups. It was intended that the information be entered directly from the application forms sent in by candidates. The programme therefore had to ask the same questions as appear in the application form, with the user filling in the appropriate answer. For an overview of the tasks performed by the software see Figure 5.

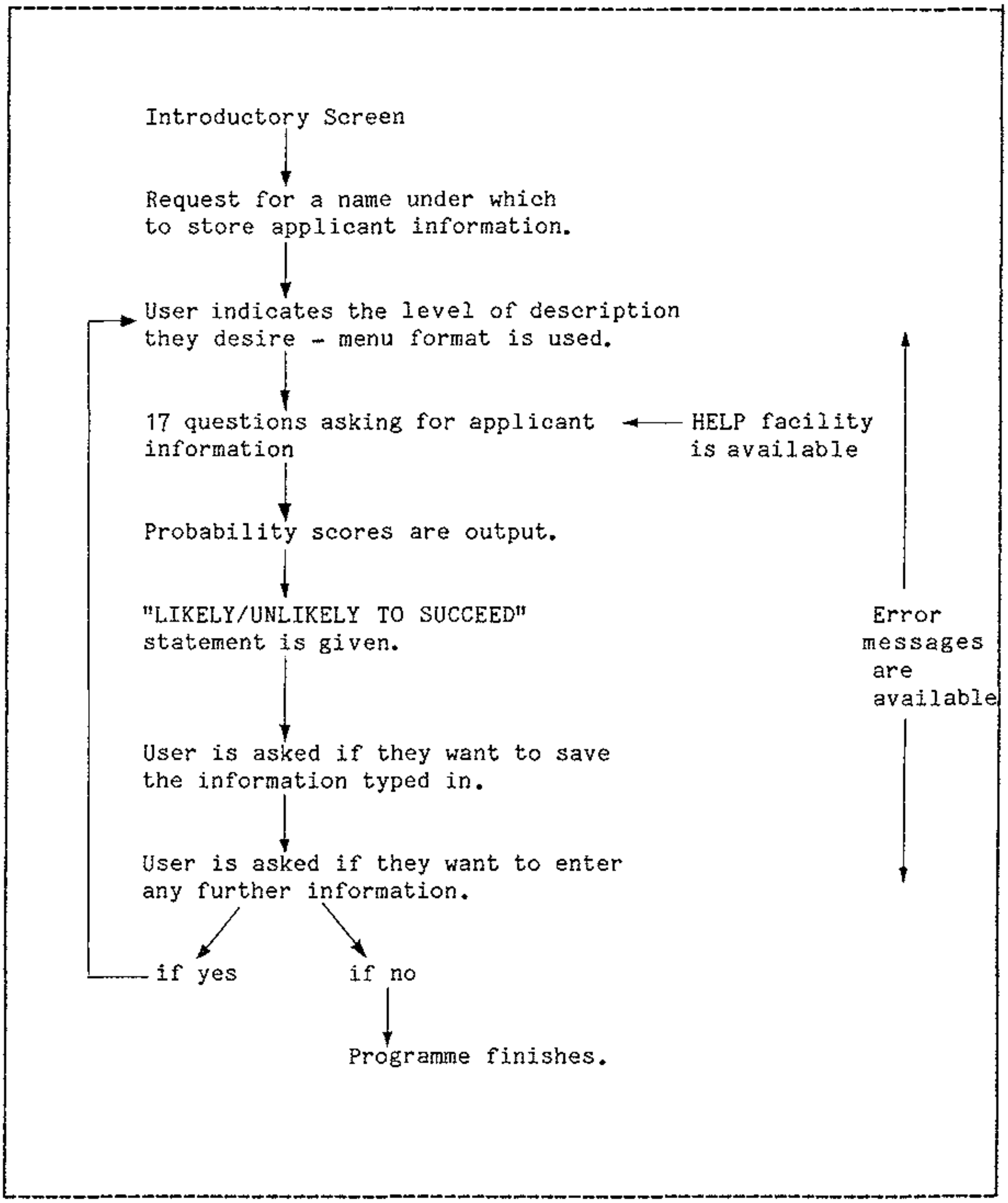


FIGURE 5: Overview of the steps the software is designed to follow.

Following the introductory screen, the programme made its first request of the user, refer to Figure 6. The file created via this command stored the applicant data entered by the user.

```
Welcome to the Manawatu Polytechnic Comprehensive Nursing Course's
          "Applicant Success Prediction" program.

The first thing you need to do is enter ONE WORD under which the
program can store the applicant information you are about to enter.

* Make a note of the word you choose. *

* DON'T use a word you or someone else has used before. *

Please type ONE WORD under which the applicant information
can be stored ...
```

FIGURE 6: Introductory Screen of Software.

Next the user was presented with a menu from which they were able to select how much information was presented to them on screen (see Figure 7). Three levels of choice were available. The first represented the introductory level and was the level at which first time users of the programme were encouraged to start. For each of the questions asked full descriptions of the type and form of the information required were given. (The questions correspond to items found to contribute significant amounts of information for the classification of subjects into successful or unsuccessful groups.) The second level provided briefer sentence descriptions of the questions. The more concise form meant the time taken to work through the programme was reduced. Finally the third level provided one or two word prompts of the desired information. See Figures 8, 9 and 10 for a comparison of these formats.

Please type the number of the desired option in the space provided. If you have not used this program before it is suggested that you choose number 1.

1. This is the full description. First a summary of the question is given, followed by a full description and then a brief prompt after which you need to enter the required data.

2. This is a summary description. A summary question is given followed by the brief prompt after which you need to enter the data.

3. This option includes only the brief prompt. It is designed for those familiar with the program.

FIGURE 7: Information Selection Menu.

While it was expected that use of the programme would lead to familiarity with the answer type required for each question, it seemed unlikely that total retention of these details would occur. Help facilities were therefore available in the second and third levels of choice.

As stated earlier, "HELP" facilities are an integral part of interface design. It was decided that help should be available during both the sentence and one or two word prompt levels of the on-line programme. Generally the issuing of a help command results in more detail on the type of answer a particular question requires. Having provided the information the programme should continue with the level of description initially selected by the user.

17. Type the number of jobs or type of work the applicant WOULD NOT LIKE TO DO. The types of work need to be completely different.

---

17. Work not liked ...

FIGURE 8: Example of a Full Description.

If you want more information on how to answer this question, type - 99.

How many different jobs would the applicant NOT like to do?

17. Work not liked ...

FIGURE 9: Example of a Brief Description.

```
If you want more information on how to
answer this question, type - 99.
--------------------------------------------------------

 17. Work not liked ...
```

FIGURE 10: Example of a Prompt.

For each question asked the appropriate help command was also displayed on the screen for the users convenience. For questions where 'y'/'n' responses were needed, the user typed "HELP" when unsure of what was wanted. While for questions where a numeric entry was required the user entered "99" to obtain the extra information.

After selecting the level of descriptive information they desired users worked through 17 questions, entering information directly from the application form in most instances. Where this was not appropriate, for example where applicants were required to explain themselves using several reasons, the user had to decide whether reasons matched those identified in the programme or were different. Refer to Figure 11 for an illustration.

A users capacity for information retention was not expected to be infallible and neither was their ability to avoid making mistakes. The provision of error detection methods was therefore seen as essential. Errors occurred whenever a response was entered that the programme did not expect or could not recognise.

Error messages were designed to be informative and helpful. The risk of discouraging the user needed to be avoided, so terse, abrupt messages, (eg. "ERROR - ILLEGAL SYNTAX") were seen as inappropriate (Shneiderman, 1980). When error messages were issued, an indication was given that an error had been made as well as how the error could be remedied. The messages were timed to be issued immediately an incorrect response was entered into the programme.

```
Note that if this question is answered 'no' or 'n',
questions 9-13 inclusive will be skipped and also
answered 'no' or 'n'.

 8. In this case the work needs to be for a period longer
    than 4 weeks.
    The following are some examples of the type of job
    the applicant might have held :
            - general nursing student/nurse.
            - nurse aid.
            - enrolled nurse.
            - psychiatric nurse.
            - ancilliary staff.
           - community work (of a nursing nature).

   Type 'yes' or 'y' if the applicant has worked in a
   nursing related field for longer than 4 weeks.
   Type 'no' or 'n' if the applicant has not worked in
   a nursing related field or has worked but for a period
   of less than 4 weeks.
--------------------------------------------------------------

8. Worked in nursing ...
```

FIGURE 11: An Example of a "Decision" Question.

The screen following the seventeenth question provided probabilities of the individuals likely success or failure on the course (see Figure 12). The information was repeated in the next screen in a written form with the word "likely" or "unlikely" highlighted, see Figure 13.

```
The probability of this applicant failing year 1 of the course is: 0.04

The probability of this applicant passing year 1 of the course is: 0.96

Press RETURN to continue ...
```

FIGURE 12: Probability Score Screen.

```
From the information that has been entered this applicant is

       LIKELY

to succeed after one year of study.
NOTE however that this is a TENTATIVE prediction based on
the attributes of past applicants. You need to refer to any
other sources of information you have before making a final
decision.


Press RETURN to continue ...
```

FIGURE 13: Written Description of Applicants Likely Performance.

In the event of mistakes, the opportunity to re-enter an entire applicant data set was provided by asking the user if they wanted to "save" the last applicants information (see Figure 14). By responding "no" to the programme's question the applicants data was not saved. The user, in the next loop through the programme could enter the corrected data set.

```
You have finished this session. Do you want to save the
applicant information you have just typed in?
Type 'yes' (or 'y') or 'no' (or 'n') ...n


The information worked on in this session will NOT be saved.


Press RETURN to continue ...
```

FIGURE 14: An Example of an Interaction To Save the Information.

The last screen of the software represents the chance to restart the programme (see Figure 15). Users were asked if they wished to enter further applicant data or if they wished to terminate the session with the computer.

```
Do you want to continue entering information in to the program?

    Type 'yes' (or 'y') if you do want to add more information.

    Type 'no' (or 'n') if you have had enough.

Are you continuing ...
```

FIGURE 15: The Final Question Asked.

Where users still have application form data to enter, they would restart the programme, otherwise a session ends at this point.

Besides the obvious features such as help facilities and error messages several other interface attributes were included in the software.

The desire for closure in interactions with a computer was recognised and incorporated into the programmes design. This was achieved by developing the software so it would progress through a loop, conclude the loop and subsequently restart from the beginning. The obvious return to a starting point was included to reinforce the feeling that a section had been completed. The loop corresponded with the entry of one applicants information. Refer to Figure 5. The feelings of relief characterising closure were expected to occur at the conclusion of each of these entries.

Control was a second attribute seen as important to interface design. The provision of different levels of descriptive information was seen as one way of ensuring that the user was able to decide the course the interaction was to take. Further, the opportunity to conclude a session with the computer at the end of each application form entry, meant users were not locked into the system.

Privacy was also an attribute identified as important to novice users (James, 1981). The self contained nature of the software meant outside intervention was not required. Thus users could initiate the programme, work through it (following instructions initially), and complete the task without any outside help.

Following the assertions of Petrick (1976) the viability of using natural language dialogue within the programme was investigated. As the software design process progressed it became clear that long, user generated instructions would not be necessary for the programme's operation. For the most part users were required to enter single character responses ('y' for 'yes and 'n' for 'no') and RETURN key sequences. Such entries were more in line with a code identified as being not always easy for the novice to use (Petrick, 1976). The logic of this code however was very clear. Where 'y'/'n' responses were not appropriate, a value of some sort was required in all but one instance. The exception was for an item requesting an applicants name.

While the human side of the human-computer dialogue did not require the use of natural language phrases, the instructions, error messages and help facilities did need to present clear, easily understood information.

Error messages were presented as sentence reminders of the type of entry that was acceptable to the programme. The use of code and/or cryptic prompts was avoided. Instead an emphasis on ease of understanding for novice users was the focus.

Full, English descriptions were seen as appropriate primarily as novice computer users were to be working with the system. Yet it was also seen as useful for users to be able to turn off these full descriptions when they felt familiar with the procedures the software followed. The speed of the programme would in this way be enhanced. This aspect of the interface was selected for study in the present research.

To elaborate, the ability to turn off parts of the information in a programme is important if frustration amongst the systems users is to be avoided. Eason and Damodaran (1978) support this notion but go on to suggest that users be able to again request this information if they forget or become unsure of what the software requires. The software developed for this study incorporates the facility of turning off a detailed description of the information required when answering each question. Where briefer forms are selected it is possible though, through typing the appropriate help response, to obtain the full description for the question.

Off-line Manual

The descriptions presented in the manual were of two kinds, identical in wording to the descriptions found in the programme (see Appendix 3). The first, filling several pages, was a full description of each question. It was expected that subjects would use these full descriptions when entering the first two or three applicant data sets. Having gained some familiarity with the requirements of the programme, it was assumed that subjects would turn to briefer, one sentence descriptions of each question. If for no other reason than these briefer descriptions were all on one page, so removing the need to turn through several pages to find the relevant description.

Final Questionnaire

Primarily this questionnaire sought subject impressions of the software they had used (see Appendix 4). User feedback being an important step in the development of user friendly programmes (Kreusi, 1983). In addition however it was expected that the open-ended nature of the questions would encourage users to be honest and yield information on changed attitude towards the computer. Subject responses to the introductory and final questionnaires were therefore to be compared.

Procedure

WAB Weighting

Following approval to use the application forms of nursing course applicants from 1980-1985 inclusive, some verification of form content across these years was needed. After comparing the elements of each years forms, questions that consistently appeared in all years were selected for coding. A coding form was thus constructed and then used to transfer information from the application blanks into numeric equivalents ready for analysis. (See Appendix 5).

When all 415 sets of subject data had been prepared the information was entered into Massey University's Prime 750 computer. The SPSSX package was used for the various analyses performed. Initially a check of the completeness and correctness of the data was made. Cross tabulations and frequency listings were therefore obtained. At this point a decision was made to exclude ethnic background and gender information from the analysis. While the WAB has been found to exhibit minimal adverse impact (Reilly and Chao, 1982), New Zealand law regarding non-discrimination on the basis of race and gender precluded the inclusion of these variables (Human Rights Commission Act, 1977). Discriminant analysis using a backward stepwise procedure was employed in an effort to ascertain which combination of variables was likely to discriminate most clearly between the successful and unsuccessful groups.

Software Experiment.

Pilot trials with five subjects established the structure and task sequence of the experimental trials. These subjects were invited to work through the task, giving opinions and suggestions as they progressed. In this way the method of timing each trial was established as was the presentation and format of trial application forms.

The 45 subjects in the experiment were randomly assigned to one of three groups. All three groups received the same instructions, but the version of the programme they were to use differed.

Each subject was first asked to fill in an introductory questionnaire (see Appendix 1). Having done this, a brief description of the trial was given. This included the assurance that it was not the subject's performance as such that was being assessed but rather the relative ease with which they were able to use the software, thus it was stressed that the programme was under trial and not them.

Next a written introduction to computers and the purpose of the study was provided (see Appendix 6). Subjects were asked to read through this information and then wait for the next set of instructions. Within this introduction the general aims of the study were made clear as were the steps necessary for the subject to initiate the trial. It was required that they be able to turn on a micro computer and subsequently load a floppy diskette, then choose whichever version of the programme they were assigned to use. The introductory notes provided step by step instructions on just how to this could be achieved.

The eventual aim was to create off-line documentation that would enable novice users to operate the computer with no outside aid. At this stage it was seen as important to ascertain which aspects of the off-line and on-line procedures were easily understood and which were difficult or confusing. Where ambiguities existed there was the opportunity to rewrite those portions, so fostering a more user friendly system. Having signalled to the experimenter that they had read the information, subjects were given the opportunity to ask questions about any aspect of the notes they did not fully understand.

When both the experimenter and subject were satisfied all the information was understood a set of eight application forms (see Appendix 7 for an example) was handed to the subject. Eight forms were used, as preliminary trials with five subjects revealed that in order to ensure that the total trial time did not exceed 45 minutes a maximum of eight forms could be processed. Again any questions the subject asked were answered.

Up to this point all three trial groups had exactly the same introduction to the experiment. At this stage the experimenter verbally emphasised for each subject the trial type they were in. The initial trials with 5 subjects indicated the need for this verbal emphasis.

Subjects started a trial by indicating, with a menu, the mode of the programme they were to use. The user selected the number (one, two or three) corresponding with their trial type (see Figure 16).

```
Please type the number of the session type you are
participating in.

 1. Computerised
 2. Written
 3. Verbal

 Session NUMBER is ...
```

FIGURE 16: Trial Type Selection Menu.

The first interface was one where the individual interacted with the computer at all times. This onscreen/on-line version of the programme was designed to remove any need for outside or off-line help.

The programme was designed to present the user with a choice of the amount of help they required with any aspect of the data entry task.

In order to test the software, a second version of the programme was employed. This version was not intended to stand alone. Deliberately designed to contain the bare minimum of information, users were forced to make use of an external information source. It retained the basic structure of the programme but provided only prompts for the 17 questions. Help messages were also unavailable (compare Figure 17 with Figure 5). Thus, supplementary information, either verbal or written in form, was required to accompany it.

The alternate programme version was used by subjects in the remaining two groups.

Subjects in the second or "written" group were given a manual containing descriptions of the information required by the software (see Appendix 3). It was intended then that they use this manual during the trial, thus working by themselves. (Refer to the Materials Section).

Finally, subjects in the "verbal" group were told by the experimenter that they were to work through the task following the instructions on the screen. When they reached a point from which they could not progress, subjects were told they could ask the experimenter for help. Thus a conventional training situation was set up. The help they were given related primarily to describing the type of information the programme was requesting. The information was not volunteered, instead subjects were expected to ask for it whenever they were unsure of the data to be entered. Further, subjects were able to request full and abbreviated descriptions, the choice being left to them. The

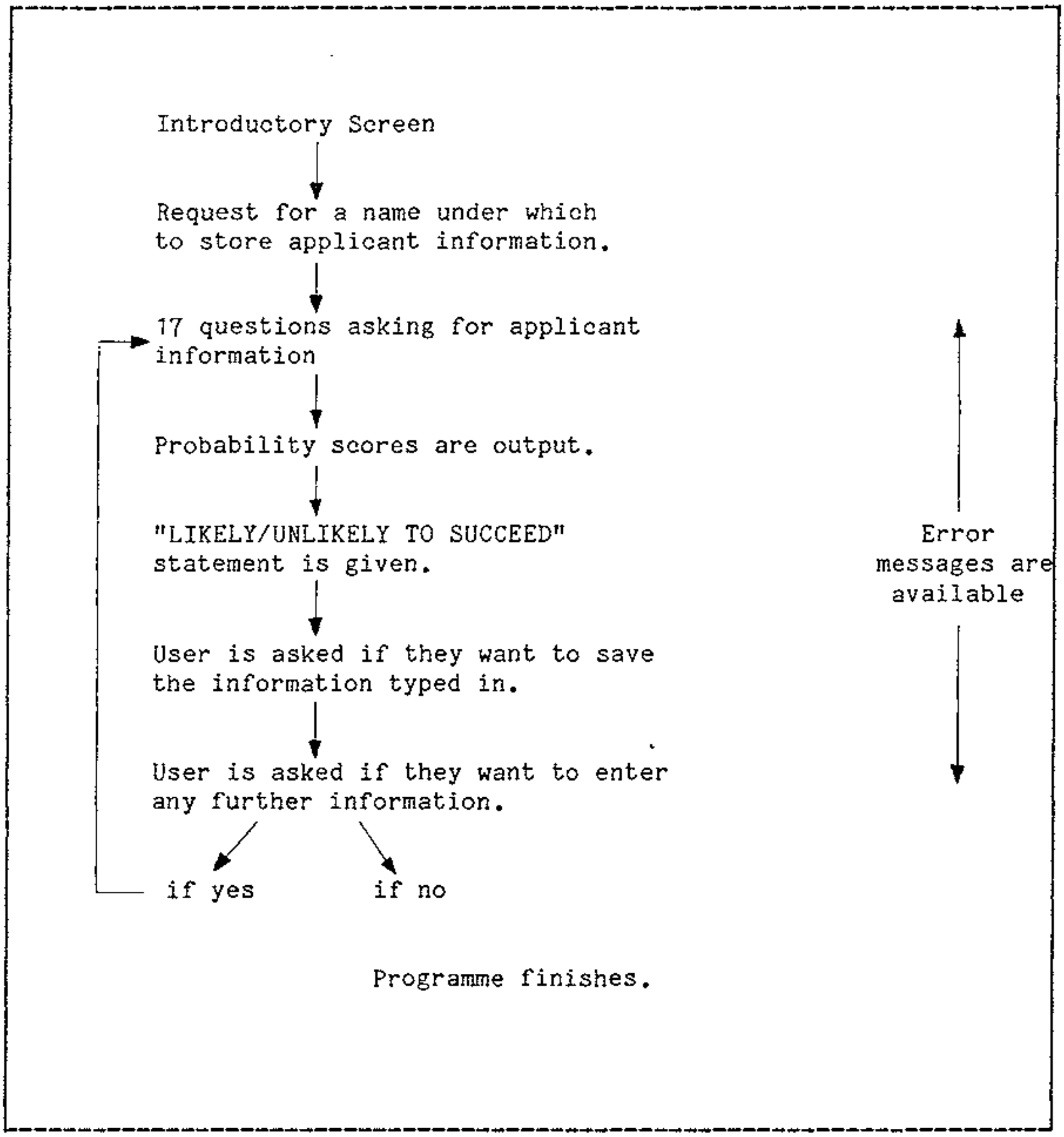


FIGURE 17: Overview of the Steps the Second Version of the Software is Designed to Follow.

descriptions were identical to the descriptions provided to the written group via a written form and the computerised group via the screen.

Having issued the various instructions and materials to the subjects the trial began. Subjects switched on the computer and loaded the programme disk then followed the instructions on screen. The trial consisted of entering eight sets of applicant data. Subjects were timed during the trial. A total time was recorded from the point of choosing the programme version corresponding with their trial type, to the displaying of the final programme message. The time taken to enter each applicant data set was also recorded. In this case timing commenced with the display of the first question and ended following response to "Do you wish to enter more applicant information?". The information each subject entered from the forms was stored on disk so it could be assessed for accuracy. In this way it would be possible to establish that the instructions had been understood and followed.

With completion of the trial, subjects were requested to fill in the final questionnaire (see Appendix 4). Having done this, subjects were thanked for their co-operation in the experiment. If result summaries were desired, it was made clear that the experimenter was happy to provide these when they became available.

CHAPTER SIX

RESULTS.

Discriminant Analysis

A backwards stepwise hierarchical discriminant analysis was performed using 36 application form data items as predictors of membership of two groups. The analysis reported in the results represents the culmination of an exhaustive series of analyses that sought to establish the optimum classification function. Predictor variables covered individual history aspects such as education and qualifications as well as reasons candidates had for applying to the course. Groups were either likely to succeed or unlikely to succeed after one year of study.

Of the original 415 cases, 98 were excluded as criterion variable information was unavailable (these were applicants for the 1985 course), a further 50 were excluded as discriminating variable data was missing. The sample of 267 subjects that remained was randomly divided into two groups. A subgroup of 80 subjects was selected as a holdout sample, with the remaining 187 cases being used in the analysis.

A significant amount of discriminating power in the original variables is evidenced by Wilks' lambda (Wilks' lambda = 0.7796, $X^2(16) = 44.07$, $p<.001$), see Table 2. The 16 variables selected following the analysis produced a moderate degree of separation between the groups as indicated by a canonical correlation of 0.4695 for the first and only discriminant function. The calculation of a squared canonical correlation value yields information of the proportion of variance in

TABLE 2

Summary of Canonical Discriminant Function

EIGENVALUE	PERCENT OF VARIANCE	CANONICAL CORRELATION	WILKS' LAMBDA	CHI-SQUARE	D.F.	SIGNIF.
0.28275	100%	0.4694913	0.7795779	44.073	16	0.0002

the discriminant function explained by the groups. In this case 22.04%, representing a moderate correlation, of the variance in the function was explained.

The sixteen variables found to discriminate between the criterion groups and their relative contributions to the variance of the function are presented in Table 3.

TABLE 3

Standardised Discriminant Function Coefficients and their Relative Contributions.

VARIABLE	COEFFICIENT	RELATIVE CONTRIBUTION
DEPENDANTS BETWEEN 6 AND 13	-0.95128	13.18 %
AGE	0.76526	10.60 %
HAS APPLICANT WORKED IN NURSING	0.67994	9.42 %
WORKED AS COMMUNITY NURSE	-0.58743	8.14 %
WORKED AS ENROLLED NURSE	-0.58162	8.06 %
NUMBER OF TIMES U.E. ATTEMPTED	0.53967	7.48 %
ANY NURSING QUALIFICATIONS	0.47862	6.63 %
HAS APPLICANT ANY DEPENDANTS	0.38705	5.36 %
NUMBER OF U.E. SUBJECTS PASSED	-0.37698	5.22 %
WORKED AS A NURSE AID	-0.33138	4.59 %
NUMBER OF JOBS NOT LIKED	0.32950	4.57 %
SOME EXPERIENCE NURSING IS GIVEN AS A REASON FOR BEING A NURSE	0.31177	4.32 %
WORKED AS NURSING STUDENT	-0.25551	3.54 %
"OTHER" REASONS FOR NURSING	0.24584	3.41 %
EVER BEEN IN HOSPITAL	-0.22207	3.08 %
WORKED AS PSYCHIATRIC NURSE	-0.17213	2.39 %

The classification of subjects into group, led to 87.94% being correctly assigned. See Table 4.

TABLE 4

Classification results for cases selected for use in the analysis.

		NO.OF CASES	PREDICTED GROUP 0	PREDICTED GROUP 1
ACTUAL GROUP	FAILED YR 1 (GROUP 0)	28	11 39.3%	17 60.7%
	PASSED YR 1 (GROUP 1)	171	7 4.1%	164 95.9%
	PERCENT OF "GROUPED" CASES CORRECTLY CLASSIFIED:			87.94%

A total of 199 cases were used in the analysis. Eleven (39.3%) of the 28 unsuccessful candidates were correctly classified, while 164 (95.9%) of the successful candidates were correctly classified.

Overall, 86.42% of the hold-out sample were correctly classified by the discriminant function calculated (see Table 5). Of the nine unsuccessful candidates in this group, none (0.0%) were correctly classified as being a member of the unsuccessful group. Seventy (97.2%) of the 72 candidates who were successful on the course were identified as being such.

The sixteen variables selected as contributing to the differentiation of the two groups and their classification coefficients are presented in Table 6.

TABLE 5

Classification results for cases not selected for use in the analyses.

		NO.OF CASES	PREDICTED GROUP 0	PREDICTED GROUP 1
ACTUAL GROUP	FAILED YR 1 (GROUP 0)	9	0 0.0%	9 100.0%
	PASSED YR 1 (GROUP 1)	73	3 4.11%	70 95.89%

PERCENT OF "GROUPED" CASES CORRECTLY CLASSIFIED: 86.42%

TABLE 6

Classification Function Coefficients.

VARIABLE	FAILED YEAR 1	PASSED YEAR 1
Age	3.252303	2.963068
No. of U.E. subjects passed	1.899426	2.244800
No. of times U.E. attempted	13.819460	11.918920
Any nursing qualifications	3.005411	-0.365703
Experience is reason for nursing	-0.012252	-1.790690
Other reasons for nursing	2.660632	1.708202
Has applicant wked in nursing	3.788690	1.644330
Worked as general nurse	-11.283700	-9.002857
Worked as nurse aid	-1.949925	-0.666090
Worked as enrolled nurse	-0.190565	4.649414
Worked as psychiatric nurse	-3.380067	-0.746158
Worked as community nurse	-1.098124	1.226727
Ever stayed in hospital	-1.628914	-0.928658
Number of tasks disliked	2.451078	1.881007
Has the applicant any dependants	-17.926990	-20.422460
Any dependants btw. 6 and 13 yrs	-4.625068	-1.227493
(Constant)	-47.567020	-38.084330

ANOVA

The experimental results were analysed using analysis of variance (ANOVA) techniques.

The first ANOVA sought to establish that trial group affected the time taken to enter eight sets of applicant information. Time refers to time per question. Time per question values were used in the comparison as the number of times a question appeared across application forms varied. Table 7 presents the results.

The interaction between trial group and application form is significant ($F(1, 294) = 7.436$, $p<.001$). Indicating that time was affected by group and application form, see Figure 18. The main

TABLE 7

Results of an ANOVA comparing time per question for each Subject across Group and Applicant Form.

Source	S. S.	D. F.	M. S.	F	Prob
BTWN SUBJECTS					
GROUP	646.6250	2	323.3125	8.693	< 0.001
ERROR BTWN	1562.1406	42	37.1938		
WITHIN SUBJECTS					
APP. FORM	10065.5313	7	1437.9329	252.998	< 0.001
GROUPxAPP.FORM	591.6875	14	42.2634	7.436	< 0.001
ERROR 2	1670.9687	294	5.6836		
SIMPLE MAIN EFFECTS BTWN. GROUP (WRITTEN, COMPUTERISED, VERBAL) MEANS					
COMPARISON 1					
WRIT. & COMPUT.	142.6658	1	142.6658	3.836	non sig.
ERROR	1562.1406	42	37.1938		
COMPARISON 2					
COMP. & VERB.	181.3847	1	181.3847	4.8767	< 0.05
ERROR	1562.1406	42	37.1938		
COMPARISON 3					
VERB. & WRIT.	645.7795	1	645.7795	17.3626	< 0.001
ERROR	1562.1406	42	37.1938		

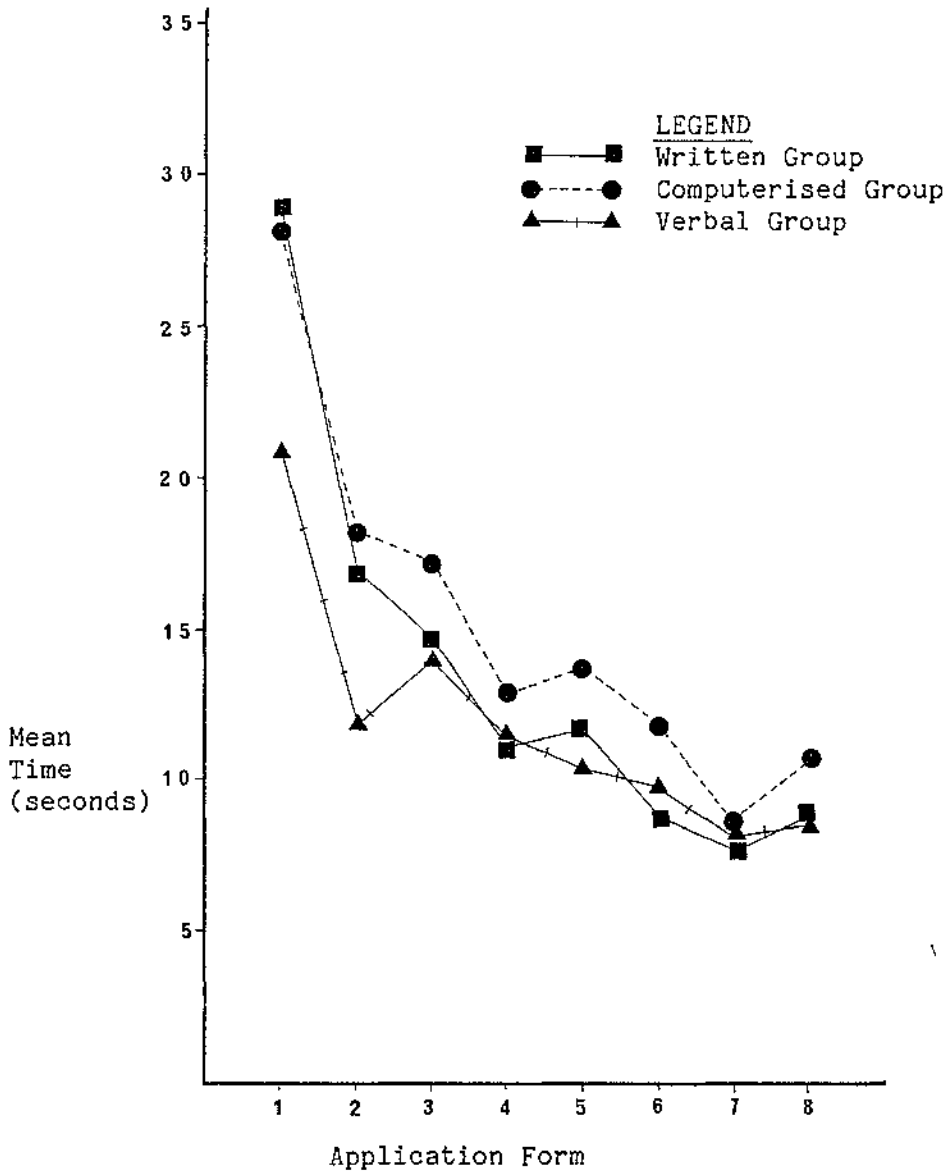


Figure 18: Mean time in seconds taken by subjects in each group to complete entering application form information.

effect for group was also significant ($F(2,42) = 8.693$, $p<.001$) emphasising that group membership affected time per question. By decomposing this main effect it was possible to establish which of the groups were contributing to the significance, see Figure 19. Three comparisons were made, did time vary significantly between 1) the written and computerised group, 2) the computerised and verbal groups and 3) the verbal and written. It was found that written and computerised groups did not differ significantly ($F(1,42) = 3.836$, $p>.05$), but that computerised and verbal groups , and written and verbal groups did differ significantly ($F(1,42) = 17.3626$, $p<.001$, $F(1,42) = 4.8767$, $p<.05$, respectively). Thus the higher times taken for the written and computerised differed significantly from those for the verbal group.

This first analysis also revealed a significant difference between time taken on each applicant form, see Figure 20. A trend analysis was performed to determine if a significant downward trend of time values was apparent. This trend analysis was performed following the guidelines presented by Keppel (1982, pp 437-441). Coefficients for the calculation of linear trends were obtained from tables produced by Pearson and Hartley (1970). Results of the analysis are presented in Table 8.

The interaction is significant indicating that the slopes of the learning curves are linear but are not the same at all levels of A, refer again to Figure 18. In addition, both the A and B_{linear} main effects are significant. The significance of the B_{linear} main effect ($F(1,42) = 595.7991$, $p<.001$) suggests that the means on the application forms decrease linearly. Group also exerts a significant effect on the linear nature of the means ($F(2,42) = 10.8305$, $p<.001$).

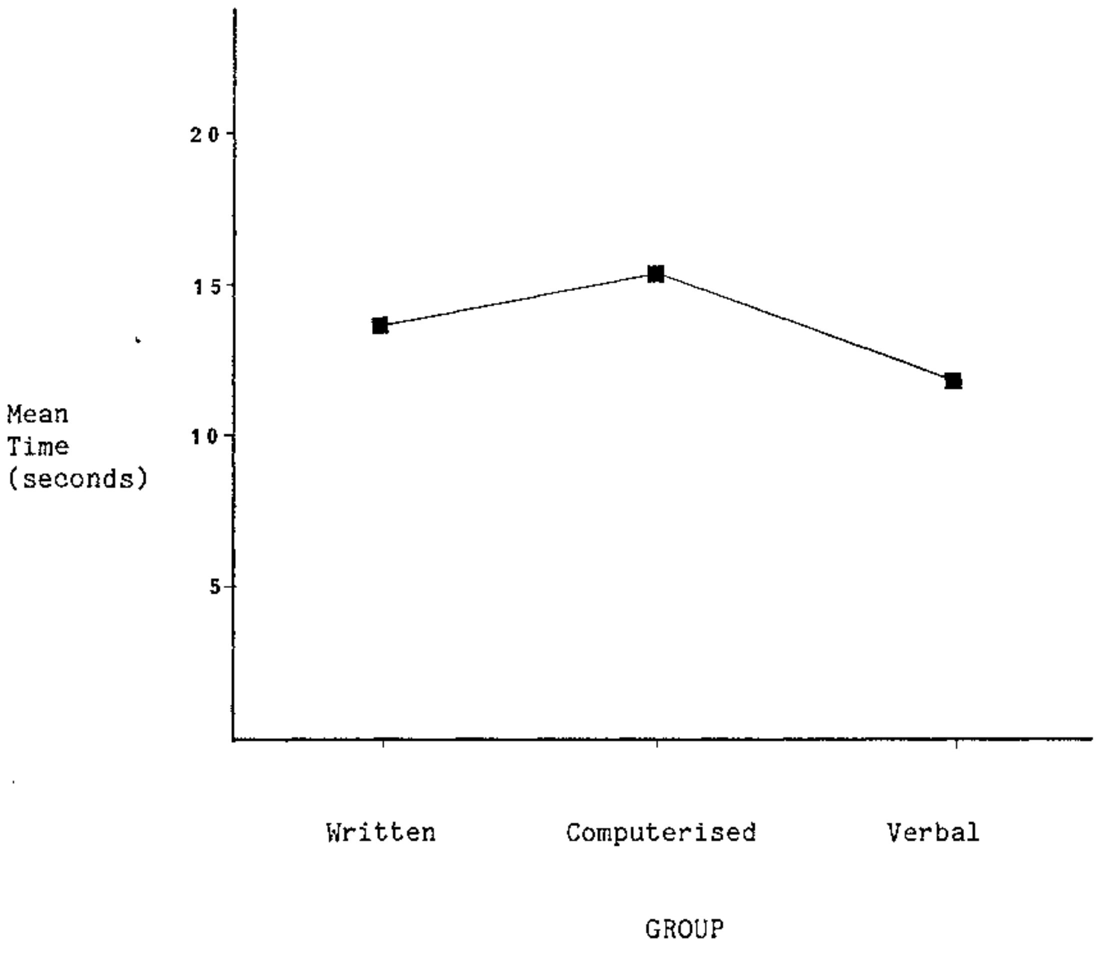


Figure 19: Group main effect times from the first ANOVA.

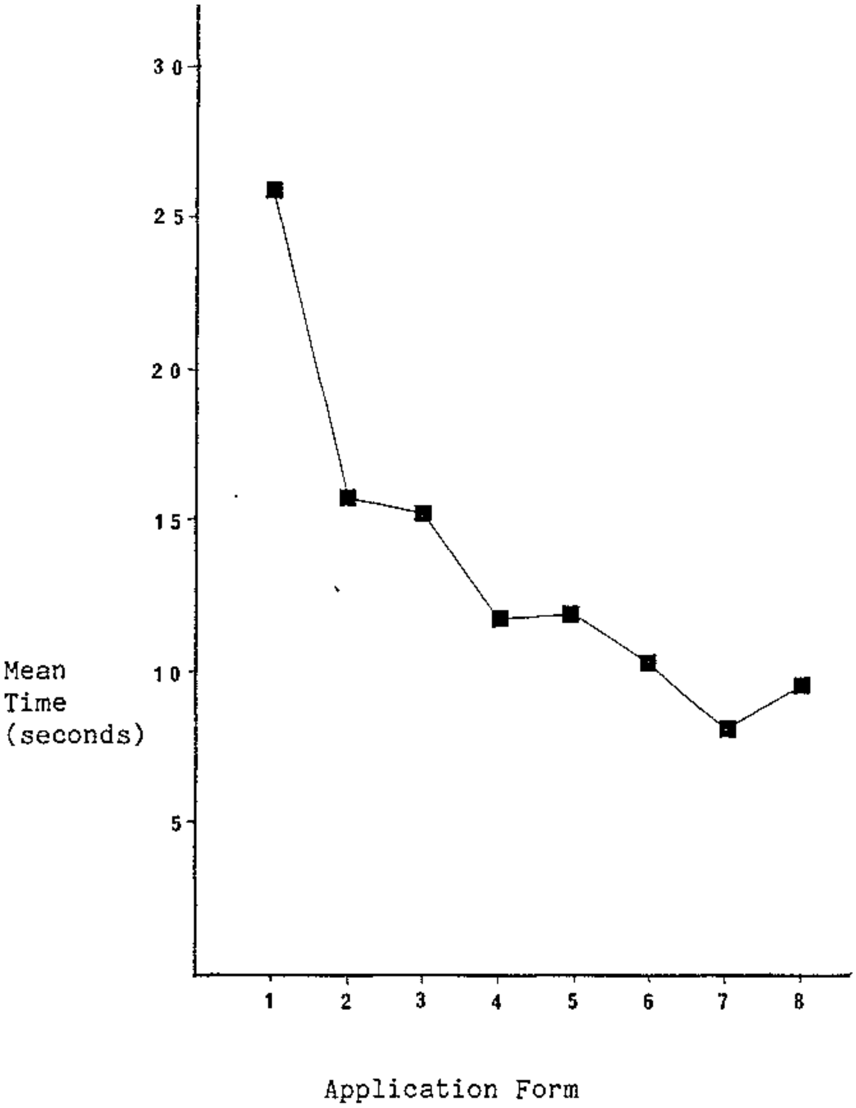


Figure 20: Main effect for application form times from the first ANOVA.

TABLE 8

Summary of the Trend Analysis.

Source	S.S.	D.F.	M.S.	F	Prob
BTWN SUBJECTS					
GROUP	542,258,097.3556	2	271,129,048.6778	10.8305	<0.001
ERROR	1,051,423,402.8	42	25,033,890.5429		
WITHIN SUBJECTS					
$APP.FORM_{linear}$	6,345,474,966.9444	1	6,345,474,966.9444	595.7991	<0.001
$GROUPxAPP.FORM_{linear}$	321,883,104.9556	2	160,941,552.4778	15.1114	<0.001
ERROR	447,315,169.6	42	10,650,361.1809		

The second ANOVA performed investigated the relationship between trial group and application form questions based on the frequency of errors. As was the case for "time" above, "errors" refer to errors per question. Again this is because the number of questions in each application form varied. See Table 9.

The interaction between group and questions was insignificant, as was the main effect for group. Thus revealing that trial group membership did not affect the errors made. The number of errors made on each question was significant, indicating some questions caused significantly more, or significantly fewer errors than others. Figure 21 presents this result graphically. Questions 7 and 16, and to lesser extent question 17, appear to have more errors made on them than other questions. A test of the significance of this apparent relationship was made by comparing questions 7, 16 and 17 with all the other questions.

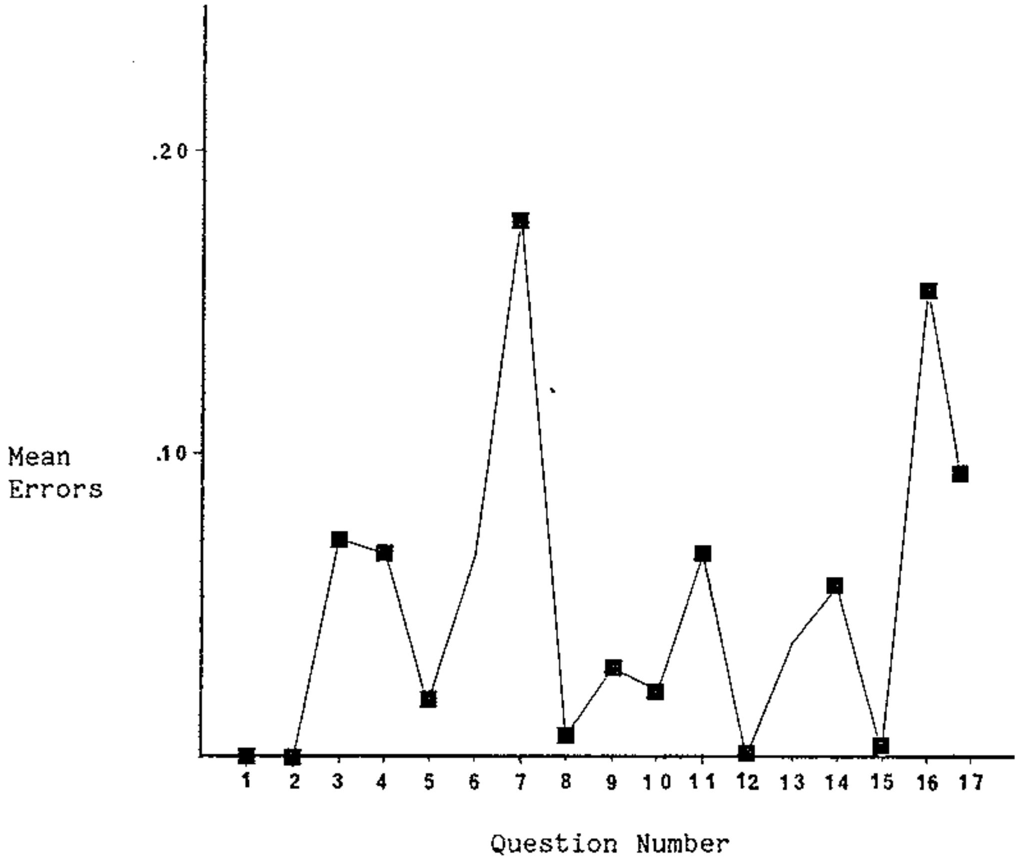


Figure 21: Main effect for question errors from the second ANOVA.

TABLE 9

ANOVA Summary Table Showing Errors per Question for Subjects across Group and Question.

Source	S. S.	D. F.	M. S.	F	Prob
BTWN SUBJECTS					
GROUP	0.0288	2	0.0144	0.465	non sig.
ERROR BTWN.	1.2981	42	0.0309		
WITHIN SUBJECTS					
QUESTIONS	2.2073	16	0.1267	9.042	<0.001
GROUPxQUESTIONS	0.4582	32	0.0143	1.022	non sig.
ERROR 2	9.4170	672	0.0140		
COMPARISON OF REPEATED MEASURE MEANS					
$QUESTIONS_{comp}$	3.6080	1	3.6080	18.7474	<0.001
$L.S._{comp}$	7.5057	39	0.1925		

It should be pointed out that for this and the remaining comparisons, separate error terms were calculated. This approach was taken following the arguments set out by Keppel (1982, pp446-447).

The results of the comparison are reported in the lower part of Table 9. A significant effect was revealed ($F(1,39) = 18.7474$, $p<.001$), thus the errors made on questions 7, 16 and 17 differ significantly from those on the remaining questions.

The third ANOVA also tested the variability of error, this time the effect of group and application form was investigated. See Table 10.

The main effect for group in this case was not significant ($F(2,42) = 0.045$, $p>.05$), thus the errors made could not be attributed to group membership. The application form main effect was significant however, ($F(7,294) = 2.676$, $p<.05$), thus errors caused by different

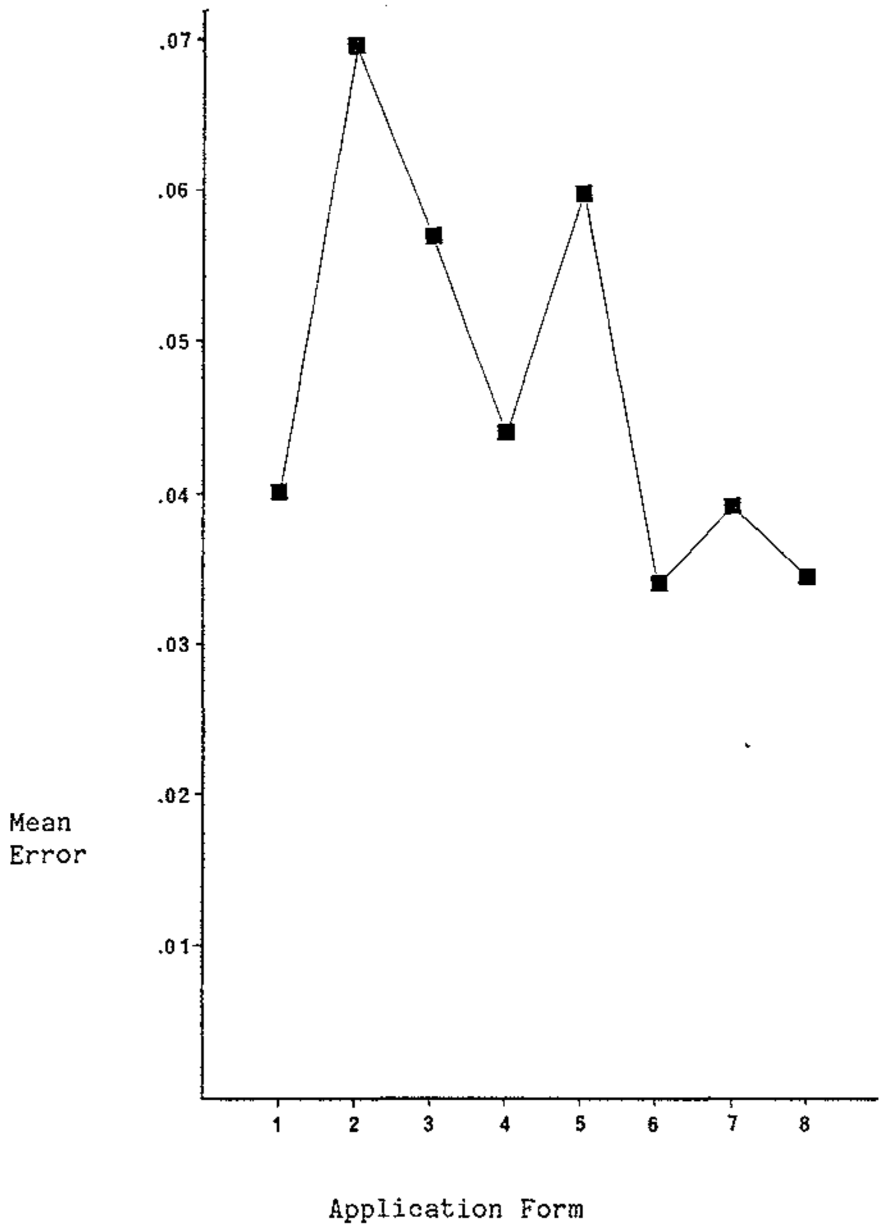


Figure 22: Main effect for application form errors from the second ANOVA.

application forms were significantly different. Figure 22 illustrates the errors made on each application form. The interaction was found to be significant ($F(14,294) = 2.809$, $p < .001$). Figure 23 reveals that on application forms 2 and 3, the errors made in the written trials appear different to those of the computerised and verbal trials. The significance of these differences were tested.

TABLE 10

ANOVA Summary Table Comparing Errors per Question for each Subject across Group and Application Form.

Source	S. S.	D. F.	M. S.	F	Prob
BTWN SUBJECTS					
GROUP	0.0009	2	0.0005	0.045	non-sig.
ERROR BTWN.	0.4274	42	0.0102		
WITHIN SUBJECTS					
APP. FORM	0.0564	7	0.0081	2.676	<0.05
GROUPxAPP. FORM	0.1185	14	0.0085	2.809	<0.001
ERROR 2	0.8858	294	0.0030		

The method used is one where use of interaction contrasts help to pin-point the interaction, a highly specific $GROUP_{comp}xAPP.FORM_{comp}$ interaction is calculated. The interaction of interest here is the interaction contrast between the written group the average of the computerised and verbal groups with a comparison between application forms two and three.

The results, presented in Table 11, reveal a significant $GROUP_{comp}xAPP.FORM_{comp}$ interaction ($F(1,42) = 12.259$, $p < .005$). The mean of the computerised and verbal groups on application form two and application form three differ significantly therefore from the two

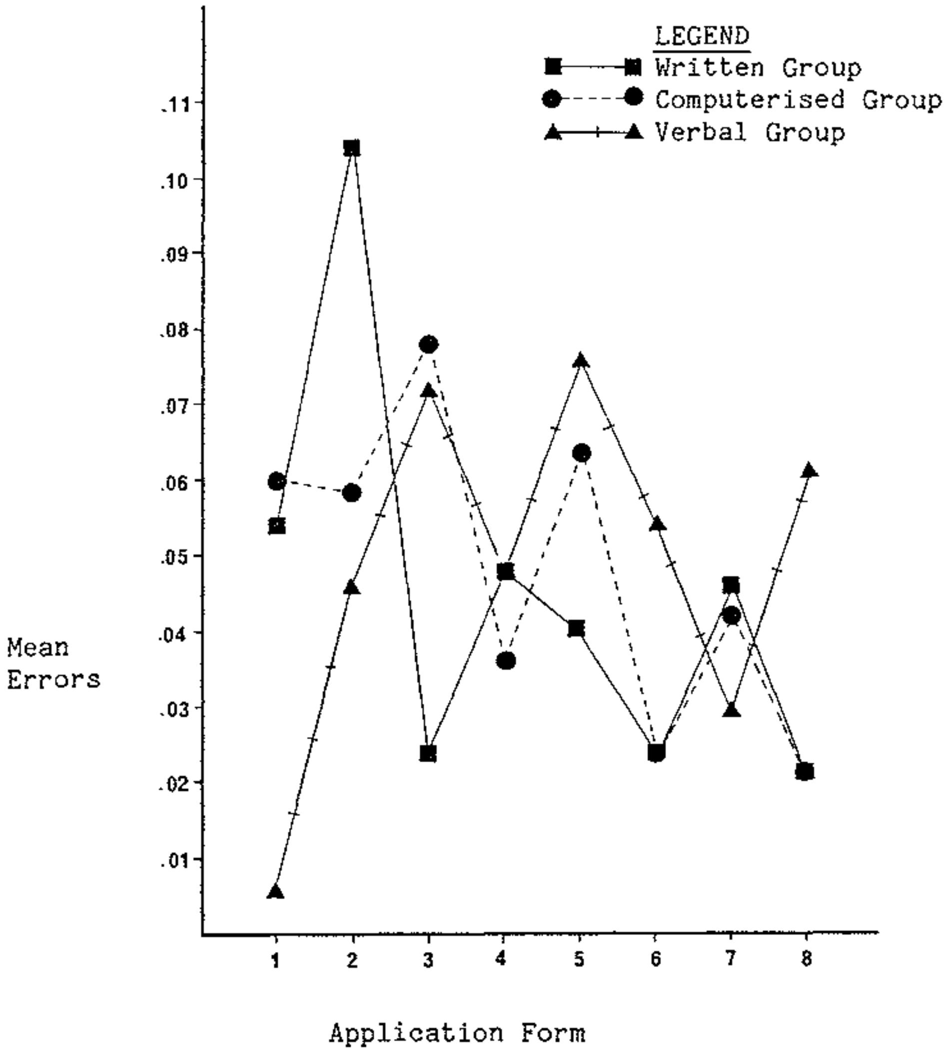


Figure 23: Mean number of errors made by subjects in each group when entering application form information based on results from the third ANOVA.

application form values for the written group. Further the significance of $APP.FORM_{comp}$ ($F(1,42) = 11.5827$, $p<.005$) indicates that application forms two and three differ significantly. $GROUP_{comp}$ returned an insignificant result ($F(1,42) = 0.000008627$, $p>.05$).

TABLE 11

Summary of Comparison Calculations.

Source	S.S.	D.F.	M.S.	F	Prob
BTWN SUBJECTS					
$GROUP_{comp}$	0.00000008867	1	0.0000000887	8.627 E-06	non sig.
ERROR BTWN	0.4274	42	0.0102		
WITHIN SUBJECTS					
$APP.FORM_{comp}$	0.0005046	1	0.0005046	11.5827	<0.005
$GROUP_{comp}$x$APP.FORM_{comp}$	0.0005339	1	0.0005339	12.2549	<0.005
ERROR	0.001829	42	0.00004357		

Questionnaires.

Introductory Questionnaire

Subjects in response to the first introductory questionnaire item "What do you think of computers", answered relatively consistently. The responses were judged as falling into one of four categories - clearly positive, clearly negative, wary and non-committal (Refer to Table 12). While there was some overlap between these categories, evidenced by difficulty in assigning one or two responses, the answers provided could generally be easily placed.

TABLE 12

Comparison of "What do you think of computers?" responses across groups.

	COMPUTERISED	VERBAL	WRITTEN
CLEARLY POSITIVE	4	4	3
CLEARLY NEGATIVE	3	3	2
"WARY"	4	4	5
NON-COMMITTAL	4	4	5
TOTAL	15	15	15

The computerised and verbal group responses were categorised identically with four clearly positive, three clearly negative, four wary and four non-committal responses being given. The written group differed slightly with three clearly positive, two clearly negative, five wary and five non-committal. It was with this groups responses that the greatest classification problems occurred.

Question two was in all instances answered in the vein that "yes, computers are useful". In six cases however, the "yes" response was accompanied by a more negative qualifier such as "or so I'm told", "as long as people don't rely on them too much" and "but it is unfortunate that they threaten jobs".

Question three, "What should computers be used for?", yielded mainly data analysis and data storage and retrieval replies (18 out of 45 subjects mentioned this role). Several subjects responded with "anything" (17 out of 45), others mentioned "teaching aids", "work aids", "business aids" and "games". Few therefore saw computers as directly linked to a decision making role.

The fourth question sought first, to establish that computers had a role to play in nursing and second to gain an idea of what this role might be. Interestingly none of the subjects saw the computer in anything but an administrative role, reflecting to some extent the general roles seen as appropriate to computers.

The final question sought affirmation that subjects were computer novices with no more than a few hours contact with a computer. Seventeen subjects had never touched a computer, of the remaining 28, 12 had used a computer once, the rest reported "limited use".

Final Questionnaire

When asked what they had liked about the programme, most subjects, 40 out of 45, mentioned that it was easy to use, easy to understand and/or it was fast. Two subjects responded negatively, one saying they liked "nothing" (a computerised trialist), the other "not much" (a written trialist). Other responses included, pleasure that the programme signalled when an error occurred, "tells you when you make a mistake" and "liked error messages explaining why an error had been made", also "no one intervening made it less nerve wracking" (a written subject) was seen as positive. See Table 13.

Question two requested criticism of the programme. See Table 14. Several subjects (nine out of the 45) found difficulty answering two specific questions, these were question 6 and question 7. Comments indicated that wording of these questions made them confusing. Of the nine subjects who had problems with these questions, five were in the verbal trial group, two in the written trial group and two in the computerised trial group.

TABLE 13

Comparison of "What did you like about the programme?" responses (Final Questionnaire, question 1) across group.

	COMPUTERISED	VERBAL	WRITTEN	TOTAL
EASY TO USE/EASY TO UNDERSTAND/FAST	13	14	13	40
NEGATIVE RESPONSE	1	0	1	2
ERROR MESSAGES	0	2	3	5
INTERESTING	1	0	1	2
				49

TABLE 14

Comparison of "What did you dislike about the programme?" responses (Final Questionnaire, question 2) across groups.

	COMPUTERISED	VERBAL	WRITTEN	TOTAL
QUESTIONS 6 AND 7	2	5	2	9
FLIPPING PAGES OF APPLICATION FORM	3	1	2	6
FLIPPING PAGES OF MANUAL	0	0	5	5
VALIDITY OF METHOD	2	7	1	10
TASK IS BORING OR REPETITIVE	2	2	2	6
SWITCHING BTW. Y/N AND NUMERIC RESPONSES ANNOYING	0	2	0	2
NO CRITICISM MADE	3	0	6	9
				47

Trial group did appear to influence the type of response made to this question. Of the nine subjects making no criticism of the programme, six were "written" trialists, three were "computerised" trialists thus none were from the "verbal" group. Other aspects of the programme at issue reflected to some extent the nature of the trial. Five written subjects had as their only criticism annoyance at flipping through the manual pages for information. One questioned the validity of the selection method, two found the task boring and/or repetitive and two expressed frustration with flipping through the application form.

The computerised version yielded similar results. Of the 11 subjects answering this question, boredom (2 of the 11 found the trial boring), and the validity of the approach (another two subjects questioned this) were seen as negative issues. Three of the remaining five subjects expressed annoyance at flipping backwards and forwards through the application form information.

Verbal group trialists, as stated above, seemed most likely to experience difficulty answering questions 6 and 7. A further seven subjects questioned the validity of the technique, particularly the reasons for choosing the variables. Two subjects felt the task would quickly become boring, one found the turning of application form pages annoying and two did not like the switching between `yes'/`no' and numeric answers.

Not every subject answered questions three and four. Of those who answered question three (a total of 20 subjects), nine mentioned the need to synchronise the order of questions in the application form with those on the screen and seven suggested the rewording of some questions (see Table 15).

TABLE 15

Comparison of "What improvements could be made to the programme?" responses (Final Questionnaire, question 3) across group.

	COMPUTERISED	VERBAL	WRITTEN	TOTAL
FEWER/CLEARER INSTRUCTIONS	1	0	1	2
REMOVE NEED TO FLIP THROUGH MANUAL	0	0	1	1
ORDER QUESTIONS SO THEY SYNCHRONISE WITH FORM	4	1	4	9
IMPROVE WORDING OF QUESTION 6 AND 7	1	2	4	7
MAKE PREVIOUS SCREEN AVAILABLE	0	0	1	1
TOTALS	6	3	11	20

Looking at these results by group, the "written" trialists seemed the most likely to make suggestions of improvement. Eleven of the 15 subjects in this group answered question 3 compared to six in the "computerised" group and three in the "verbal" group. Four "written" subjects mentioned synchronising questions, four wanted better question wording, one wanted fewer instructions, one found the flipping of manual pages tedious "it would be easier to have all the information on the screen", and one sought the provision of screen recall commands. The "verbal" subjects wanted either question synchronisation (one out of three) or improved question wording (two out of three). Finally the "computerised" subjects also referred predominantly to question ordering (four of six) and better wording (one of six). The remaining subject sought fewer, clearer instructions.

Sixteen subjects took the opportunity to add their own comments in question 4 (see Table 16).

TABLE 16

Comparison of "Any other comments?" responses (Final Questionnaire, question 4) across groups.

	COMPUTERISED	VERBAL	WRITTEN	TOTAL
POSITIVE COMMENT	2	5	2	9
QUERY ABOUT MISTAKE CORRECTION	2	0	0	2
CONCERN ABOUT SOCIAL IMPLICATIONS	0	1	0	1
CONCERN ABOUT THE VARIABLES SELECTED	0	2	1	3
WOULD TO HAVE A TURN ON THEIR OWN	0	1	0	1
TOTALS	4	9	3	16

Of these, nine made positive statements about the programme, "it was fun, I'd like another go", "enjoyed the session", three were concerned about the reasons certain variables had been selected, two queried some aspect of the programme, "can mistyped information be rectified?", one expressed concern about the social implications of the exercise and one expressed the desire to have a go on their own (this subject was in the "verbal" trial).

Finally refer to Table 17, in which a summary of the computerised groups use of the different descriptive forms are presented. Nine

TABLE 17

Summary of the use of the descriptive alternatives by the computerised group.

SUBJECT	APPLICANT FORM NUMBER* 1	2	3	4	5	6	7	8
1	F	F	F	F	F	S	S	S
2	S	S	F	F	F	F	F	F
3	F	F	F	F	F	F	F	F
4	F	S	S	S	S	S	S	P
5	F	F	S	F	F	F	F	F
6	F	F	F	F	F	F	F	F
7	F	F	F	F	F	F	F	F
8	F	F	F	F	F	F	F	F
9	F	F	S	P	P	P	P	P
10	F	F	F	F	F	F	F	F
11	F	F	S	S	P	P	P	P
12	F	F	F	F	F	F	F	F
13	F	F	F	F	F	F	F	F
14	F	F	F	F	F	F	F	F
15	F	F	F	F	F	F	F	F

* Note: F = Full Description
S = Summary Description
P = Prompt

subjects used the "full" description throughout the trial, two selected the summary version for one or two applicant forms then returned to the full description, one subject progressed to the summary form and then used the form to enter the remaining applicant data sets and three used the summary form and then progressed to the prompt. Across the entire 120 applicant forms processed by the 15 subjects, the prompt version was used nine times (7.5%), the summary 16 times (13%) and the full description 95 times (79%).

CHAPTER SEVEN

DISCUSSION

The hypothesis that the application form data would be able to discriminate between success and failure applicant groups has been supported by the results of the present study. A classification success of 87.94% in the validation sample and 86.42% in the cross validation sample indicates a marked amount of function stability (Sands, 1978). It is more often the case when using WAB techniques to find decreases in the significances of validation and cross validation analyses. (Cascio, 1976: Schwab & Oliver, 1974). The 10% decrease (90.74% - 80.39%) in classification success across validation and cross validation samples reported by George (1983) was not apparent in the present results. Inspection of Tables 3 and 4 does reveal that the successful classification of applicants unsuccessful on the course decreased from near to 40% in the validation sample to 0% in the cross validation sample. The small number of subjects in the unsuccessful group of the hold out sample detracts somewhat from the impact of this result.

Table 18 presents the relative contributions made to overall discrimination by variables selected into the discriminant function calculated by George (1983). These can be compared with the results in Table 3.

Three variables are found to contribute to discrimination in both analyses, age, number of U.E. subjects held and number of jobs not liked. Parallels can be drawn between other variables, although the way in which they have been measured differs. For example, dependants are identified as important in both analyses.

TABLE 18

Relative contribution to overall discrimination of variables into the discriminant function. (George, 1983).

VARIABLE	RELATIVE CONTRIBUTION
MARITAL STATUS	18.12%
AVERAGE SC MARK	15.83%
AGE	15.58%
ARRANGEMENTS FOR DEPENDANTS CARE	11.27%
TOTAL UE SUBJECTS HELD	9.92%
NUMBER OF DEPENDANTS	9.30%
TOTAL SC PASSES	9.23%
AGE OF CHILDREN	6.74%
TYPES OF WORK NOT WANTED	4.03%

The variables selected in the discriminant function of the present study appear to fall into groups emphasising the following factors, see Table 19.

Age and the two dependants' related questions have clear connections, particularly in terms of dependant age group selected. When grouped these items contribute over a quarter of the total variance in the function(29.14%). Questions relating to qualifications, that is, U.E. subjects passed, U.E. attempts and any nursing qualifications, add a further 20% to the variance. Work history also appears to exert some influence, specifically work with a nursing emphasis. Six variables reflect this area of information and combined provide 36% of the variance. The remaining variables describe whether applicants have ever stayed in hospital, reasons for wanting to be a

TABLE 19

Groupings of discriminating variables.

GROUPS			
STAGE IN LIFE	QUALIFICATIONS	WORK HISTORY	CAREER DECISION
AGE	NUMBER OF U.E. SUBJECTS PASSED	WORKED IN NURSING	HOSPITAL STAY
DEPENDANTS BETWEEN 6 & 13	NUMBER OF U.E. ATTEMPTS	WORKED AS A NURSE AID	REASON FOR CHOOSING NURSING - EXPERIENCE
ANY DEPENDANTS	ANY NURSING QUALIFICATIONS	WORKED AS A NURSING STUDENT	OTHER REASONS FOR CHOOSING NURSING
		WORKED AS A COMMUNITY NURSE	NUMBER OF JOBS DISLIKED
		WORKED AS A PSYCHIATRIC NURSE	
		WORKED AS AN ENROLLED NURSE	

nurse, and the number of jobs applicants would not like to do. As a group these variables appear to relate to the decision to become a nurse.

The items selected to discriminate between applicants held few surprises. "Stage in life" variables indicated that older applicants were more likely to experience difficulty in completing the first year than were younger applicants. Having dependants between 6 and 13 points to the fact that women are likely to want to return to the work force at this stage in their lives. However the later stage of life is likely to count against the individual, reasons for this can at this stage only be guessed at.

Dependants, through their prominence as a discriminating variable, should be taken as one force likely to exert an influence. The double role of student and home worker will, in all probability put pressure on these applicants that younger, childless candidates will not encounter. The pre-adolescent ages of the children further suggest that parental supervision of them in after school hours and holidays is necessary. Conflicts in allocation of time to children and study are therefore probable.

Qualifications and their associated study skills appear to add to an individuals probability of completing the first year of study. Applicants with University Entrance passes and/or specific nursing related qualifications will clearly have an advantage over those with fewer academic qualifications in the present case. The academic orientation of the course precludes entrance (in most cases) of applicants with less than school certificate passes. It appears from the results that higher school qualifications go some way towards improving the individuals chances of succeeding in the first year of the course. The number of attempts at U.E. perhaps emphasises that academic success and application is of importance considering the nature of the course.

Experience in nursing also appears to be of some importance. The type of experience is also surprisingly influential. Enrolled nursing experience in particular is likely to impact in a negative way, contributing to the likelihood of failing the first year. Enrolled and community nursing may represent nursing areas that in previous years were performed by individuals with few educational qualifications.

The last variable category has been loosely described as "career decision" and includes variables that represent contributions to the decision making process. A hospital stay could provide strong, positive role models for candidates and the opportunity to describe unappealing work, clarify for the individual just what they seek from their nursing career. By giving reasons for choosing nursing, applicants are encouraged to determine for themselves a set of goals.

That the factors described above are easily identified suggests that future application forms should seek to obtain specific information aiding in the description of individuals along these lines. At this stage application form information is able to successfully classify candidates into likely to succeed and likely to fail groups. This classification may be used for screening purposes with a view to decreasing the number of students failing the first year. As such the predictive validity of the technique appears to be relatively good. However the overall success of classification is subject to improvement. The assignment of subjects to the "unlikely to succeed" groups in the analysis and holdout samples is an obvious area needing improvement. The testing of the discriminant function with future intakes is perhaps one method of attempting this.

These results may also be used in a more "person management" sense. That is several variables identified appear to reduce some individuals chances of succeeding on the course. This being the case the Polytechnic has the option of changing conditions in an effort to deflect the negative consequences of some factors. For example, making available creche and after school child minding services for students with dependants must go some way towards improving these students chances for success. Also planning assignment completion dates to fall before school holidays begin, so individuals with children are not forced to juggle school holiday family activities with course requirements. Further, the academic bent of the course suggests the need for study skill workshops for those identified through WAB techniques as being at risk of failing these course requirements.

Which variables are appropriate to a candidate screening function and which to a more general description of the individual, is a decision those using WAB techniques must make. In many instances the choice will be moderated by laws governing fair employment, or, as in this case, selection practices. Race and sex could not therefore be used for selection. Instead, should for example, race prove to discriminate between success and failure on the course, steps can be taken to identify the reasons and start the removal of obstacles.

The course could be divided into academic and practical sessions. Application form variables that point to successful performance in these portions could be identified and more specific selection procedures developed around these. The opportunity exists then to extend the results into other areas.

With hindsight some improvement in WAB information collection and coding could be instituted. Application forms in future years could be designed to include items investigating traits identified through detailed job analysis as being of importance to successful completion of the first year. Only research of WAB development that follows task analysis will reveal if the present method of entering existing application form items differentiates sufficiently between the groups. It may be the case that important preselection information is omitted from the forms because job analyses have not been performed.

A greater total sample size and in particular larger failure group may aid in the discrimination of applicants into likely to succeed and likely to fail groups. A New Zealand wide study of applicant selection for nursing courses should provide a sufficiently large subject pool. However the logistics of such an exercise may preclude it ever being attempted. One major problem would be the collecting together of similar variables. Application forms differ between polytechnics meaning some work at identifying common item themes may be required.

The results obtained for the first part of this study reveal the WAB as being a valid applicant screening method. The success of the classification procedure and cross validation coupled with easy development, simple scoring, low cost, fairness and easy conceptual understanding make it a screening procedure that warrants a place in any selection situation.

The results of the analyses of variance reveal that for the time data, time varied significantly across group and application form. Figure 18 illustrates the general trend with the computerised subjects taking the longest on each application form, followed by the written subjects with the verbal group having the fastest times.

The difference in times taken can be explained through the following factors. The need to read information in the computerised and written trials has probably meant that task completion times for these groups are longer than those for the verbal group. The facility to draw additional explanation of what was required was unavailable to these groups. Verbal trialists used the experimenter's help to gain for themselves quick affirmation that the data they were entering was in the correct format.

The significance of the linear decrease in times taken across the application forms reveals that a learning curve pattern of response had occurred. With the completion of the first two applicant forms therefore it seems that subjects had grasped the major idea of the programme. The big drop in time between the first and second forms is not repeated in any of the following inter-form distances.

The second ANOVA performed compared the errors made by subjects on each question across groups. While group did not influence errors, certain questions did cause significantly more errors than others. Three questions in particular showed greater error scores, these were questions 7, 16 and 17, when compared with the remaining questions this difference was found to be significant. The third hypothesis, addressing uniformity of items must therefore be rejected. Question 7 in particular had been singled out by subjects as being problematic. Being a long question the time required to read through it properly may have been more than many subjects were prepared to spend, particularly since no other question required as much reading time. Incomplete comprehension of the question, through limited attention to detail may therefore explain the relatively high number of errors.

Question 16, "Has the applicant any dependants?", may have caused errors as it was a question requiring a yes/no response. The information in the application form was presented numerically, that is "What are the ages of any dependants?", thus subjects were forced to translate the information into the form required. Question 17, "How many dependants are between the ages of 6 and 13 years?", was incorrectly comprehended in many instances with subjects tending to enter either the total number of dependant children or the actual age of a dependant.

Most of the errors stemmed from misunderstanding questions. Whether this was caused by incomprehension of information because questions were difficult or because time was not taken to read the questions properly is not clear. The desire to "beat the clock" may have been of some influence. The experimenter was seated close by the trialist and had two stopwatches. This could have led to subjects feeling pressured, despite assurances being given that the programme was timed not the person.

The third ANOVA was also concerned with error measurement, this time errors accrued on each application form were compared for subjects across groups. Again group was found to be an insignificant factor in the number of errors made. However Figure 23 revealed an apparent divergence on application forms 2 and 3 between the "verbal" and "computerised" groups and the "written" group. When tested the interaction affect in the comparison was found to be significant. It is implied therefore that group 3, the "written" group, obtained scores significantly different from those obtained by groups 1 and 2, "verbal" and "written", on application forms 2 and 3.

"Written" subjects scored more errors than any group on any form for application form 2. The errors were significantly greater than the number of errors scored by the "computerised" or "verbal" group. The time taken to complete this form was no different to that taken by the "verbal" group. Why therefore did this error rate occur? The second application form required more information to complete than the first, thus "written" trialists were forced to read through a number of pages of information in the manual. The pressure of time may have led them to skim through the descriptions, taking insufficient time to absorb properly the responses required. This being the case, the subjects then had less chance of responding correctly to the questions asked. The third questionnaire concentrated on fewer questions, total time taken on the application form was only slightly less than that spent on the previous lengthy form. The error rate of subjects decreased as time was taken to understand questions properly (see Figure 18).

The insignificance of the group effect on number of errors made across applicant form can be explained to some extent by the widely diverse error rates across the groups. The verbal group in particular exhibited an almost quadratic error pattern, starting with very few errors on the first application form, these increased dramatically on the second to fifth forms then dropped again on the sixth and seventh, finally rising again on the eighth form to almost reach the peak error rates of forms 3 and 5. The latter stages of the experimental trial for this group, that is forms 5-8, exhibited error rates for the verbal group different to those of the computerised and written groups. For three of the four forms the verbal group had more errors while on the seventh form they had fewer. The suggestion is therefore that the verbal group did differ from the computerised and written groups and that this was apparent after the completion of the first four application forms. With a greater number of trials this pattern may have been more marked. This conclusion is similar to that reached by Kennedy (1975), where manual and on-line programme instructions were found to yield performances that were similar.

The classification of subjects into computer attitude groups revealed that the members of the groups were similar in their regard for computers. The off-line introduction to the programme and to the computer could therefore be appropriately used for each group. That all subjects found the programme easy to use, leads to the conclusion that the ideas behind the programme and instruction in computer use presented prior to each trial, adequately introduced subjects to the task.

The application form items were not expected to cause significantly different numbers of errors between groups. However performance on the experimental task resulted in some differences. The verbal group experienced most problems with questions 6 and 7. These questions required subjects to read through an applicants reasons for choosing nursing as a career. The descriptions of how to respond to the questions were lengthy, as subjects needed to determine if certain reasons were given within the answer. This lengthy description has apparently led to most confusion amongst the verbal group. The fact that subjects had to retain verbally presented information describing which responses to the questions fitted the purpose of the question is one explanation for the confusion. The expressed dislike for question 7 is accompanied by the decrements in performance for this question evidenced in the results.

Question 16 and 17 also resulted in greater numbers of errors. Here however subjects did not perceive the question as particularly problematic. Having stated this however, two subjects in the verbal group did express a dislike of the switching between yes/no and numeric answers. Question 16 is a yes/no question that follows a numeric question and question 17 is another numeric question. This gives rise to yes/no, numeric, yes/no, numeric sequence and would appear to be the source of the annoyance. One solution to this problem would be to group all yes/no and all numeric questions together. However such an approach would then make the ordering of questions in the programme very different to those on the application form.

The synchronisation of questions on screen and in the form was one improvement to the programme nine subjects (out of 20) identified. Most subjects were observed during the trials to turn backwards through the application form looking for information. The items on work experience in particular led to this behaviour. The relevant data could be obtained from two parts of the questionnaire. It appears that memory of passing the information previously stopped subjects from turning the page to find the data. The initial trials made with five subjects had indicated that this backwards searching for information might be problematic. Experimental subjects were therefore instructed to look through the first one or two forms to establish for themselves the location and the content of the items. Again it appears from the results obtained that subjects did not understand that some information was presented twice and that data could be found for all but the last two question by progressing to the next page. For questions 16 and 17 subjects returned to the front page of the form for the answers. This was not however seen as breaking up the sequential "flow" of responding as the forms were in nearly all cases closed again after use.

As a tool the programme represents an ideal approach for staff wishing to determine if a student, struggling through the course, was classified as a border line candidate initially. The opportunity for ongoing updating of classification data sets is very real. Further the technique could be diversified for the making of ongoing assessments. Students when half way through the course could be compared with past students achievements at the same stage. Attributes of past students who went on to fail or achieved borderline passes could be identified and present students exhibiting the same traits provided with extra assistance. The ability of the system to show up "at risk" individuals could thereby lead to an improvement in the subjects chances of passing the course. The benefits to both the polytechnic and the individual concerned warrant the time to identify these traits. Aspects of successful performance in both the academic and practical elements of the course could be identified.

The actual use of the programme within the experiment led to some interesting conclusions. The verbal version of the programme, while the fastest, could not be put forward as the best version as it required the presence of a trainer. The difference in speed from the other two methods was not sufficiently great to warrant the conclusion that this method was the superior. Indeed there is the indication that errors made by the verbal group were higher than other groups. Once the nature of the task was thought to be understood and time taken on each trial was decreasing the number of errors made remained higher than for other groups. It appears therefore that the provision of help only when it was requested meant that a question misinterpreted early in the interaction would continue to be so.

This is in fact true of all groups, however the descriptions had to be read in the other instances meaning that, initially at least, subjects were forced to take some time to understand the information presented. Verbal subjects were observed to ask for information only once before answering a question. If not immediately understood guesses would have to be made about the appropriateness of a response. Following the entry of the first three application forms, subjects in the verbal group seldom requested a questions description, the errors earned therefore on the last four application forms can be tentatively explained by the misinterpretation of earlier presented information.

Some assessment of user performance aimed at ensuring that information has been interpreted correctly and to establish that facilities are being first utilised and second used appropriately may be necessary. In a wider field this would seem particularly sensible as user satisfaction with a system may be enhanced if all features intended for use are in fact used.

Results for the computerised and written applications were virtually indistinguishable. In terms of time taken on the task the two groups did not differ significantly. However the written group differed from the computerised group when the errors made on the second and third application forms were compared.

With these facts in mind the computerised version could tentatively be posited as the version yielding the best user performance, providing support for the second research hypothesis. The self-contained nature of this version was seen as one of its main advantages. The manual of the "written" version, found by some to be cumbersome, would not therefore be needed. Indeed support for this was found in the comments of the "written" group. When given the opportunity, some subjects expressed frustration at having to flip through the pages of the manual while others actually suggested putting the information on the screen to improve the cohesion of the procedure. Having said this, there is also some support for retaining a manual, particularly in situations where certain user attributes are displayed. If an individual is hesitant about using a computer the added assurance provided by a manual may in fact aid their on-task performance. While an individual who displays confidence (as measured by preliminary procedures) will be likely to cope without a manual. It seems that users receive a certain amount of security knowing that help is available in a form with which they are familiar. This leads to the conclusion that the "best" approach to software presentation may still require some of accompanying documentation. It may be that in the next decade of computer use the need for this decreases as computers become more familiar to the general population.

During trials with the computerised group, a note of which level of information a subject chose was made.

Results indicate that subjects made varying use of the availability of both help information and the alternative information presentations. Subjects who quickly grasped the sequence of events and understood the task aims were observed to be those who were prepared to experiment with the alternative presentations. The summary version was used more often than the prompt, in fact, the prompt was never used without the summary having first been used. Subjects tended then to take things one step at a time, indeed in two instances subjects selected the summary, felt uncomfortable with it and returned to the full description. The opportunity to select and reselect a presentation appeared to be reasonably well used.

The biggest problem found in the present study was how to ensure that subjects read thoroughly the material presented to them. Comments in the final questionnaire that specifically addressed the validity of the method show that subjects did not absorb the description of variable selection. The errors made on the questions further illustrate the misinterpretation of information for two questions at least (questions 6 and 7). This criticism must be moderated somewhat however as the wording of these questions was found difficult. In other cases misinterpretation was evident.

Within the wider sphere of interface testing the present experiment exhibits the advantages that can be obtained following any one of several models.

The Gould and Lewis (1985) model of system design as presented above has been partially replicated. The first and second principles have been followed. Thus an early focus on the user population and empirical measurement of the prototypes has led to some reworking of the overall software design. The need for iterative testing and retesting while acknowledged, has within this study only been initialised and has yet to be completed. The effectiveness of this approach is evident through identification of flaws in the relatively simple programme of the study. Users comments and their observed reactions to certain of the programmes features provide further important information. Error messages in particular drew informative responses. Subjects seemed to appreciate them and were a little surprised that a machine could apparently recognise their mistakes.

In terms of Yestingsmeiers (1984) suggestions for interface design and testing, the present experiment also goes some way towards fulfilling a proposed approach. The first study phase where preliminary designs are returned to the end user(s) for perusal, was included in an adapted form in the present study. This step establishes that the designer is following the correct line and that all aspects of the programme are being addressed. The absence of an easily identified end-user population meant that the proposed design was presented to the Head of the Department of Nursing for comment. In addition initial design developments were tested with a small sample of five users.

The second phase was met through including software features established in Human Factors research (eg. Shneiderman, 1980) as being important to the human-computer interface. This warranted the inclusion of for example, "HELP" facilities, meaningful feedback, easily understood error messages and an uncluttered screen presentation. Also within this phase initial software testing is performed, the five subjects incorporated into this first phase provided preliminary information on the syntactic and semantic aspects of the programme.

The last phase involved "final" testing and documentation. In the present study feedback on the programme and its accompanying manuals was obtained from the experiment. The users were provided with materials aiming to introduce computers and describe the software's purpose. None of the subjects appeared to have any difficulty understanding this material, no questions were asked and no reference made to it in the questions assessing users feelings about the programme.

Having followed Yestingsmeiers procedures, results again indicated that some redesign was necessary before the programme could be put into general use.

The benefits of incorporating certain features into the overall software design were not directly investigated. Thus for example, providing for closure, user control and wording instructions in natural language, while included into the programme were not specifically measured. However the ease with which most subjects were able to implement the programme and the general liking expressed by many for it, leads to the conclusion that these features contribute as intended to the programmes easy use.

In general the adherence to a design methodology increases the reliability of a programme (Kopetz, 1976). Through the testing and debugging performed by the researcher before experimentation began, efforts at achieving reliability were made. Thus the programme was run with test data sets developed to specifically establish that error facilities were operationalised as intended. Further the output from the programme was checked for its content.

Steps toward ensuring validity were taken first through assessment of five early programme trialists and then via the computerised experimental groups results. As this testing occurs early within the overall design process and then again following some redesign of the programme, the development process outlined by Fujii (1981) has been replicated.

A dynamic rather than static approach to software validation has therefore been performed (Howden, 1981). Some static validation was however included, specifically, the experimenters method of designing the programme led to paper and pencil testing of the procedure. Since only the experimenter used the techniques, conclusions as to the programmes efficiency cannot be made.

Having assessed the approach taken within the present study, the directions that might be taken in the future with first WAB development, second human-computer interactive systems and lastly computerised screening or selection processes will be discussed.

The availability of computer data processing facilities has promoted in recent times (Proctor, Lassiter & Soyars, 1976; Matteson, 1978; Owens & Schoenfeldt, 1979) the use of multivariate analysis procedures when determining WAB weights. This approach is supported by the success of classification in the present study. Future work in this area might build on the advantages computer technology engenders. Thus data files can be easily updated and revised. Indeed the addition of each completed years results to the existing file will serve to strengthen the classification results obtained in the present study. The variables identified as contributing to the discrimination will forseeably increase the amount of discriminating power they presently exhibit.

When compared to alternative weighting techniques discriminant analysis appears as a viable alternative that produces useful information (the relationship of variables to the criterion) not available via non-multivariate procedures.

In terms of the computerisation of WAB methods, future developments may build on the present software or incorporate its ideas within their structure. Extensions made could include the output of lists ranking applicants according to the probability scores obtained or alternatively listing applicants alphabetically. The variable weightings for those identified as borderline passes or unlikely to pass the first year could be produced so that factors counting against these candidates might be identified and steps taken to reduce their impact.

The overall aim is clearly to improve current selection results and at the same time simplify the methods utilised.

Future interface design need not necessarily follow the steps of the present study. Good, Whiteside, Wixon and Jones (1984) describe an alternative approach where novice user behaviour was taken as being inherently sensible. The computer software developed for these users was therefore developed to adapt to their behaviours. The goal was to build an interface based on observation and analysis of user behaviour with minimal shaping of that behaviour. The interface contained no help, no menus, no documentation and no instructions. Users were set several tasks to be performed using an electronic mailing procedure. Subjects were instructed to proceed using commands they thought would achieve each task's objectives. General guidelines on command structure were given, for example, commands were usually verbs and were short. Where commands issued by subjects were not recognised and acted upon by the computer, they were intercepted by a hidden human operator and translated into a command the computer could understand. The user was therefore given the impression that the command was automatically handled. Each subjects behaviour was analysed and their method of issuing commands incorporated into the software. The ultimate aim was that the software interface be capable of accepting and dealing with a high proportion of novice users spontaneous commands. Initial research results were encouraging, with, by the end of the experiment the software able to recognise over 75% of users spontaneous commands. Further, subjects were able to complete meaningful and useful work within an hour without documentation or a help facility. With regard to the present study therefore user responses would ideally be self-generated. Reductions in training times may occur, however the simplicity of the software may limit the amount of improvement possible.

A second alternative to the training of novice users in software use is the fostering of mimicry. Good et al (1984) describe the phenomenon which became evident in users deprived of instruction in system use. When faced with an unfamiliar system subjects were observed to use any clues to appropriate response that might be given. The suggestion is then that examples rather than instructions of appropriate responses will more readily be learnt by subjects. A users natural inclination is to respond to the computer in a way similar to the computer's own messages.

A third possibility for novice training is the introduction of On-line tutorials. A novel approach is the interactive on-line tutorial where a novice user is instructed by the computer to carry out commands. Al-Awar, Chapanis and Ford (1981), outline reasons for the apparent effectiveness of this mode of instruction. Included in their discussion are the following points.

- the user does not have to keep shifting attention between the terminal and the instructional material.
- the user practices the very skills needed to operate the system.
- the user is able to work alone, at their own pace and without the embarrassment of mistakes made before a human instructor or fellow students.

Al-Awar et al stipulate that designers of interactive, on-line tutorials must deal with issues of instructional design and with problems associated with the novelty of the computer environment. It is also suggested that repeated testing and refinement of the software take place.

With the implementation of a programme in the "field" valuable information can be collected through monitoring of the systems use. Goodwin (1982) was interested in comparing the use of two message handling systems. The functions the systems performed were similar, their respective user interfaces differed however. Both systems were tested in the field, over one month therefore all message handling commands issued by users were recorded. It was found one systems features were more extensively utilised than the other. When reasons for this were investigated it was found that the system used more extensively and successfully provided better feedback and tolerance of novice users. The tolerance apparently encouraged users to explore more of the system and thereby use more of its features.

Mayer's suggestion in 1967 that investigation of how to make computers more approachable and more effectively used, is twenty years later again being made.

The ongoing assessment of software is a theme also addressed by Carey (1982). Carey emphasises that assessment of the elements of a task is necessary as stress and job dissatisfaction are seen as linked to inadequate systems. If staff are reluctant to use a system, that system cannot be seen as being successful.

Field use of the software designed for the present study should therefore be accompanied by ongoing assessment. In particular, measurement of user errors and feedback on the programmes suitability to long term use need to be monitored.

The scope for computerising aspects of the personnel selection function seems unlimited. The present results while obviously uncovering issues of appropriate interface design also show that the execution of the weighting procedure was successful.

CHAPTER EIGHT.

OVERALL SUMMARY AND CONCLUSIONS.

The WAB provided good overall classification of subjects in both the analysis and holdout samples. However some improvement in the assignment of individuals to the unlikely to succeed group is desirable. This may be achieved through extra attention to job analysis techniques and the subsequent inclusion of weighting variables. In terms of the resources available however the results obtained are very reasonable. Indeed the ease with which it was possible to develop the weighted application blank must point to the relevance of the technique to the field of selection. In combination with other selection techniques therefore the WAB is able to provide a potent pre-selection classification of applicants into likely to succeed and unlikely to succeed groupings.

As was outlined in the discussion no one mode of software instruction could be viewed as the best. Instead the combination of on-line error messages and help facilities coupled with a written manual is likely to provide the most successful approach. Having said this however, future research into, for example, user derived software and the new field of on-line tutorial instruction are likely to markedly alter current practices of software instruction.

A further feature identified as important in developing software is that of providing an adaptable interface that is able to cater to a wide spectrum of computer user. The novice users participating in the present research highlighted this fact. It took little time for them to become familiar with the requirements of the software leading them to placing increasing demands on the softwares ability to adapt. There is an increasing awareness amongst researchers in this field that their target user population will become more, rather than less, diverse.

The feasibility of computerising the WAB selection function has been borne out by this research. The scope for computerised selection tools would seem unlimited. Whether in the future a job applicant is screened initially by computer before any contact with a human selector remains to be seen. As a method a speeding up the selection process however the use of one or more computerised tools would seem entirely probable.

In conclusion then the fields of computer-human interface design and personnel selection have been married through the computerisation of a selection technique. Researchers of this area would do well to bear in mind the many issues of interface design and subsequent presentation. Novice users in particular, if dissatisfied with a package are unlikely to be easily enticed into experimenting with second package. The potential for improvement in both the reliability and validity of selection is very real, with perhaps the biggest advantage being the consistency with which it is possible to present tests or even patterned interviews. With the increasing visibility of computers in day to day living the eventual acceptability of such an approach to selection is virtually assured.

References

Al-Awar, J. Chapanis, W. & Ford, W. R. (1981). Tutorials for the first time computer user. I.E.E.E. Transactions on Professional Communications, 24 30-37.

Allen, R. B. (1982). Cognitive factors in human interaction with computers. Behaviour and Information Technology, 1 257-278.

Arvey, R. D. (1979). Unfair discrimination in the employment interview: Legal and psychological aspects. Psychological Bulletin, 86 736-765.

Ash, P. & Kroeker, L. P. (1975). Personnel selection, classification and placement. Annual Review of Psychology, 26 481-507.

Asher, J. S. & Sciarrino, J. A. (1974). Realistic work samples: A review. Personnel Psychology, 27 519-534.

Binet, A. & Simon, T. H. (1905) Methodes nouvelles pour le diagnostic du niveau intellectual des anormaux. Annee Psychologique, 11 191-244.

Blum, M. & Naylor, J. (1968). Industrial Psychology. New York: Harper and Row.

Bo, K. (1982). Human computer interaction. Computer, 15 9-11.

Buros, O. K. (1978). The eighth mental measurements yearbook. Highland Park, New Jersey: Gryphon.

Carey, T. (1982). User differences in interface design. Computer, 15 14-20.

Carroll, J. M. & Carruthers, C. (1984). Training wheels in a user interface.Communications of the A. C. M., 27 800-806.

Cascio, W. F. (1975). Accuracy of verifiable biographical information blank responses. Journal of Applied Psychology, 60 767-769.

Chapanis, A. (1976). Engineering Psychology. In M. D. Dunnette (Ed.), Handbook of industrial and organisational psychology (pp. 697-744). Chicago, Illinois: Rand McNally.

Chapanis, A. (1970). Systems staffing. In De Greene, K. B. (Ed.), Systems psychology (pp. 357-382). New York: McGraw Hill.

Chapanis, A. (1965). Words, words, words. Human Factors, 7 1-17.

Cooley, W. W. & Lohnes, P. R. (1971). Multivariate data analysis. New York: Wiley.

Davis, R. (1983). Task analysis and user errors: A methodology for assessing interactions. International Journal of Man-Machine Studies, 19 561-574.

Davis, R. (1983). User error or computer error? Observations on a statistics package. International Journal of Man-Machine Studies, 19 359-376.

De Greene, K. B. (1970). Systems psychology. New York: McGraw Hill.

Downs, S. (1977). Trainability testing: A practical approach to selection. (Training Information Paper, Number 11). London: Her Majesty's Stationery Office.

Dubois, P. H. (1970). A history of psychological testing. Boston: Allen and Bacon.

Du Bolay, B. & O'Shea, T. (1981). Teaching novices programming. In Coombs, M. J. & Alty, J. L. (Eds) Computing skills and the user interface (pp.147-200). New York: Academic Press.

Dunnette, M. D. (1979). Personnel selection and classification systems. Annual Review of Psychology, 30 477-525.

Dunnette, M. D. & Borman, W. (1979). Personnel selection and classification systems. Annual Review of Psychology, 30 477-526.

Dwyer, B. (1981). Programming for users: A bit of psychology. Computers and People, 30 11-14 26.

Eason, K. D. (1976). Understanding the naive computer user. The Computer Journal, 19 3-7.

Eason, K. B. & Damodaran, L. (1981). THe needs of the commercial user. In Coombs, M. J. & Alty, J. L. (Eds) Computing skills and the user interface (pp.115-139). New York: Academic Press.

Ebhardt, B. T. & Muchinsky, P. M. (1982). An empirical investigation of the factor stability of the Owen's Biographical Questionnaire. Journal of Applied Psychology, 67 138-145.

Engel, S. E. & Granda, R. E. (1975). Guidelines for man-display interfaces. I.B.M. Poughkeepsie Laboratory Technical Report TR 00.2720

England, G. W. (1971). Development and use of weighted application blanks (Second Edition). Dubique: W. C. Brown.

England, G. W. (1961). The development and use of weighted application blanks. Dubique, Iowa: W. C. Brown Company.

Federico, S. M., Federico, P. A. & Lundquist, G. W. (1976). Predicting womens turnover as function of extent of met salary expectations and biodemographic data. Personnel Psychology, 29 559-566.

Finkle, R. B. (1976). Managerial assessment centres. In M. D. Dunnette (Ed.), Handbook of industrial and organisational psychology (pp. 861-888). Chicago: Rand McNally.

Foley, J. D. & Wallace, V. L. (1974). The art of graphic man-machine conversation. Proceedings of the I.E.E.E., 62 462-471.

Franz, C. R. & Robey, D. (1984). An investigation of user-led systems design: Rationale and political perspectives. Communications of the A. C. M., 27 1202-1209.

Freeburg, N. E. (1967). The biographical information blank as a predictor of student achievement: A review. Psychological Reports, 20 911-925.

Freyd, M. (1923). Measurement in vocational selection: An outline of research procedure. Journal of Personnel Research, 2 215-249 268-284 377-385.

Fujii, M. S. (1981). Independent verification of highly reliable programs. Proceedings COMPSAC77, I.E.E.E., 38-44.

Galton, F. (1902). Life history album (2nd ed.). New York: MacMillan.

George, D. I. (1983). Weighted application forms: A comparison of methodologies. Unpublished Honours Project.

Goldstein, I. L. (1971). The application blank: How honest are the responses. _Journal of Applies Psychology_, _55_ 491-492.

Good, M. D. Whiteside, J. A. Wixon, D. R. & Jones, S. J. (1984). Building a user derived interface. _Communications of the A.C.M._, _27_ 1032-1043.

Gould, J. D. & Lewis, C. (1985). Designing for usability: Key principles and what designers think. _Communications of the A.C.M._, _28_ 300-311.

Green, T. R. G. (1980). Programming as a cognitive activity. In Smith, H. T. & Green T. R. G. (Eds) _Human interaction with computers_. London, New York: Academic Press.

Gross, M. L. (1962). _The brain watchers_. New York: Random House.

Guion, R. M. (1976). Recruiting, selection job placement. In Dunnette, M. D. _Handbook of industrial and organisational psychology_. Chicago, Illinois: Rand McNally.

Guion, R. M. (1965). _Personnel testing_. New York: McGraw-Hill

Hansen, W. J. (1971). User engineering principles for interactive systems. _Proceedings of the Fall Joint Computer Conference_, _39_ 523-532.

Hill, I. D. (1972). Wouldn't it be nice if we could write computer programmes in ordinary english: or would it? _The Computer Bulletin_, _16_ 306-312.

Hinrichs, J. R. (1976). Reliability of measurement in situational exercises: An assessment of assessment center testing. _Personnel Psychology_, _29_ 31-40.

Hollingworth, H. L. (1923). _Judging human character_. New York: D. Appleton-Century.

Houghton, R. C. (1984). On-line help systems: A conspectus. _Communications of the A.C.M._, _27_ 126-133.

Howden, W. E. (1981). Introduction to software validation. In Miller, E. & Howden, W. E. (Eds) _Tutorial: Software testing and validation techniques_. New York: I.E.E.E. Computer Society Press.

Hull, C. L. (1928). _Aptitude testing_. New York: Harcourt, Brace and World.

Human Rights Commision Act (1977). Wellington, New Zealand: Government Printer.

I.P.M. Information Services (1982). _Computerising personnel systems: How to choose and where to go_. London: Institute of Personnel Management.

Jagodzinski, A. P. (1983). A theoretical basis for the representation of on-line computer systems to naive users. International Journal of Man-Machine Studies, 18, 215-252.

James, E. B. (1981). The user interface: How we may compute. In Coombs, M. J. & Alty, J. L. (Eds) Computing skills and the user interface. New York: Academic Press.

Kemp, R. (1982). Pascal for students. London: Edward Arnold (Publishers) Limited.

Kennedy, T. C. S. (1975). Some behavioural factors affecting the training of naive users of an interactive computer system. International Journal of Man-Machine Studies, 7 817-834.

Kennedy, T. C. S. (1974). The design of interactive procedures for man-machine communication. International Journal of Man-Machine Studies, 6 309-334.

Klecka, W. R. (1975). Discriminant analysis. In Nie, N. H. Hull, C. H. Jenkins, J. G. Steinbrenner, K. & Bent, D. H. (Eds) S.P.S.S.: Statistical Package for the Social Sciences. New York: McGraw-Hill Book Company.

Klimoski, R. J. & Strickland, W. J. (1977). Assessment centers: Valid or merely prescient. Personnel Psychology, 30 353-361.

Kopetz, H. (1979). Software reliability. London: The MacMillan Press Limited

Lachenbrach, P. (1975). Discriminant analysis. New York: Hafner Press.

Landy, F. J. (1985). Psychology of work behaviour (Third Edition). Homewood, Illinois: The Dorsey Press.

Landy, F. J. (1976). The validity of the interview in police officer selection. Journal of Applied Psychology, 61 193-198.

Landy, F. J. and Trumbo, D. A. (1980). Psychology of work behaviour. Homewood, Illinois: The Dorsey Press.

Latham, G. P. Saari, L. M. Pursell, E. D. & Campion, M. A. (1980). The situational interview. Journal of Applied Psychology, 65 422-427.

Lawrence, D. G. Salsburg, B. L. Dawson, J. G. & Fasman, Z. D. (1982). Design and use of weighted application blanks. Personnel Administrator, 27 47-53.

Lee, R. & Booth, J. M. (1974). A utility analysis of a weighted application blank designed to predict turnover for clerical employees. Journal of Applied Psychology, 59 516-518.

Llewelyn, A. I. & Wickens, R. F. (1968). The testing of computer software. In Naur, P. & Randell, B. (Eds) Software engineering (NATO Science Committee) Brussels: NATO, Scientific Affairs Division.

Lopez, F. M. (1966). Evaluating executive decision making: The in-basket technique. (A.M.A. Research Study, 75) New York: American Management Association.

Martin, J. (1973). Design of man-computer dialogues. New York: Prentice-Hall.

Matteson, M. T. (1978). An alternative approach to using biographical data for predicting job success. Journal of Occupational Psychology, 51 155-162.

Mayer, S. R. (1967). Computer based subsystems for training the users of computer systems. I.E.E.E. Transactions on Human Factors in Electronics, 8 70-75.

McMurray, R. N. (1945). Validating the patterned interview. Personnel, 23 263-272.

Miller, L. A. & Thomas, J. C. (1977). Behavioural issues in the use of interactive systems. International Journal of Man-Machine Studies, 9 509-536.

Miner, J. B. (1970). Executive and personnel interviews as predictors of consulting success. Personnel Psychology, 23 521-538.

Monahan, C. J. & Muchinsky, P. M. (1983). Three decades of personnel selection research: A state of the art analysis and evaluation. Journal of Occupational Psychology, 56 215-225.

Morland, D. V. (1983). Human factors guidelines for terminal interface design. Communications of the A.C.M., 26 484-494.

Morrison, P. R. (1983). A survey of attitudes towards computers. Communications of the A.C.M., 26 1051-1057.

Mount, M. K. Muchinsky, P. M. & Hanser, L. M. (1977). The predictive validity of a work sample: A laboratory study. Personnel Psychology, 30 637-645.

Mozieco, H. (1982). A human-computer interface to accomodate user learning stages. Communications of the A.C.M., 24 100-104.

Muchinsky, P. M. & Tuttle, M. L. (1979). Employees turnover: An empirical and methodological assessment. Journal of Vocational Behaviour, 14 43-77.

Muensterberg, H. (1914). Psychological and industrial efficiency. Boston: Houghton-Miflin.

Neiner, A. G. & Owens, W. A. (1985). Using biodata to predict job choice among college graduates. Journal of Applied Psychology, 70 127-136.

Nickerson, R. S. (1981). Why interactive computer systems are sometimes not used by people who might benefit from them, International Journal of Man-Machine Studies, 15 469-483.

Nickerson, R. S. (1969). Man-computer interaction: A challenge for human factors research. I.E.E.E. Transaction on Man-Machine Systems, 10 164-180.

Owens, W. A. (1976). Background data. In Dunnette, M. D. Handbook of industrial and organisational psychology. Chicago, Illinois: Rand-McNally.

Owens, W. A. & Schoenfeldt, L. F. (1979). Towards a classification of persons. Journal of Applied Psychology, Monograph, 65 569-607.

Pace, L. A. & Schoenfeldt, L. F. (1977). Legal concerns in the use of weighted applications. Personnel Psychology, 30 159-166.

Paxton, A. L. & Turner, E. J. (1984). The application of human factors to the needs of the novice computer user. International Journal of Man-Machine Studies, 20 137-156.

Petrick, S. R. (1976). On natural language based computer systems. I.B.M. Journal of Research, 20 314-325.

Pew, R. W. & Rollins, A. M. (1975). Dialog specification procedure (Revised Edition). Cambridge, Massachusetts: Bolt, Beranek and Newman.

Proctor, J. H. Cassiter, W. E. & Soyars, W. B. (1976). Prediction of young naval officer retention. Personnel Psychology, 29 567-581.

Prother, D. C. Berry, G. A. Bermuday, J. M. (1972).The effect of prompting and feedback on performance during learning, stress and transfer of perceptual skill. Proceedings of the Annual Conference of the American Psychological Association, 7 643.

Reilly, R. R. & Chao, G. T. (1982). Validity and fairness of some alternative employee selection procedures. Personnel Psychology, 35 1-62.

Roach, D. E. (1971). Double cross validation of a weighted application blank over time. Journal of Applied Psychology, 55 157-160.

Robertson, I. T. & Kandola, R. S. (1982). Work sample tests: Validity, adverse impact and applicant reaction. Journal of Occupational Psychology, 55 177-183.

Robertson, I. T. & Mindel, R. M. (1980). A study of trainability testing. Journal of Occupational Psychology, 53 131-138.

Rosenbaum, R. W. (1976). Predictability of employee theft using weighted application blanks. Journal of Applied Psychology, 61 94-98.

Rouse, W. B. (1979). Human factors engineering. In Connolly, K. (Ed) Psychology survey number 2. London: George Allen and Unwin Limited.

Saja, A. D. (1985). The cognitive model: An approach to designing the human-computer interface. SIGCHI Bulletin, 16 36-40.

Schmidt, F. L. Grenthal, A. L. Hunter, J. E. Berner, J. G. & Seaton, F. (1977). Job samples versus paper and pencil trades and technical tests: Adverse impact and examinee attitudes. Personnel psychology, 30 187-197.

Schneider, G. M. Weingart, S. W. & Perlman, D. M. (1978). An introduction to programming and problem solving with Pascal. New York: John Wiley and Sons.

Schoenfeldt, L. F. (1974). Utilisation of manpower: Development and evaluation of an assessment-classification model for matching individuals with jobs. Journal of Applied Psychology, 59 583-595.

Schwab, D. P. & Oliver, R. L. (1974). Predicting tenure with biographical data: Exhuming buried evidence. Personnel Psychology, 27 125-128.

Segal, B. Z. (1975). Effects of method error interruption on student performance at interactive terminals. Illinois: University of Illinois Press.

Shneiderman, B. (1983). High tech can stimulate creative action: The increased ease of use of computers supports individual competence and productive work. SIGCHI Bulletin, 14 6-7.

Shneiderman, B. (1982). The future of interactive systems and the emergence of direct manipulation. Behaviour and Information Technology, 1 237-256.

Shneiderman, B. (1980). Software psychology: Human factors in computer and information systems. Cambridge, Massachusetts: Winthrop Publishers.

Shneiderman, B. (1979). Human factors experiments in designing interactive systems. Computer, 12 9-19.

Short, L. E. (1979). Now you can micro-computerise your personnel system. Personnel Journal, 57 622-625.

Smith, D. (1978). Selection and the decision making process. Management Decision, 16 23-31.

Smith, M. C. & George, D. I. (1985). Forecasting using weighted application forms: A comparison of old and new methodologies. Awaiting publication.

Sneeringer, j. (1978). User interface design and text editing: A case study. Software: Practice and Experience, 8 543-557.

Tatsuoka, M. M. (1970). Discriminant Analysis: The study of group differences. Champaign, Illinois: Institute for Personality and Ability Testing.

Terman, L. M. (1917). A trial of mental and pedagogical tests in the civil service examination for policemen and firemen. Journal of Applied Psychology, 1 17-29.

Tesler, L. (1983). Enlisting user help in software design. SIGCHI Bulletin, 14 5-9.

Treu, S. (1977). Interactive command language design based on required mental work. International Journal of Man-Machine Studies, 7 135-149.

Uhlaner, J. E. (1977). The research psychologist in the army 1917-1977 (Research Report 1155). Alexandria, Virginia: U.S. Army Research Institute.

Walker, A. J. (1980). The 10 most common mistakes in developing computer based personnel systems. Personnel Administrator, 25 39-42.

Walther, G. H. & O'Neil, H. F. (1974). On-line user-computer interface: The effects of interface flexibility, terminal type and experience on performance. A.F.I.P.S. Conference Proceedings, 43 379-384.

Wasserman, T. (1973). The design of idiot proof interactive systems. Proceeding of the National Computer Conference, 42.

Waters, L. K. Roach, D. & Waters, C. W. (1976). Estimates of future tenure, satisfaction and biographical variables as predictors of termination. Personnel Psychology, 29 57-60.

Weiss, D. J. (1976). Multivariate procedures. In Dunnette, M. D. (ed.) Handbook of industrial and organisational psychology. Chicago: Rand McNally.

Wells, D. L. & Muchinsky, P. M. (1985). Performance antecedents of voluntary and involuntary managerial turnover. Journal of Applied Psychology, 70 329-336.

Wernimont, P. F. & Campbell, J. P. (1968). Signs, samples and criteria. Journal of Applied Psychology, 52 372-376.

Whyte, W. H. (1956). The organisation man. New York: Simon and Schuster.

Wickens, C .D. & Kramer, A. (1985). Engineering psychology. Annual Review of Psychology, 36 307-348.

Winograd, T. (1977). A framework for understanding discourse. In Just, M. A. and Carpenter, P. A. (Eds) Cognitive processes in comprehension. New York: Lawrence Erlbaum Associates.

Wright, C. R. (1969). Summary of research on the selection interview since 1960. Personnel Psychology, 22 391-413.

Yestingsmeier, J. (1984). Human factors considerations in development of interactive software. SIGCHI Bulletin, 16 24-27.

Zedeck, S. (1971). Identification of moderator variables by discriminant analysis in a multi-predictable group validation model. Journal of Applied Psychology, 55 364-371.

APPENDIX ONE

INTRODUCTORY QUESTIONS ON ATTITUDES TO COMPUTERS

Please answer the following questions as honestly as possible.
You should not need to spend any more than 5 minutes on them.

1. What do you think of computers?

____
____
____
____

2. Are computers useful?

____
____
____
____

3. What should computers be used for?

____
____
____
____

4. Is there a place for computers in nursing?

____
____
____
____

5. Have you ever used a computer?

____
____
____
____

Thank-you.
Please read through the introduction to the experiment now.

APPENDIX TWO.

```
program nurses(namefile);
(* A  program  to  calculate  the weighted  scores  of  comprhensive *)
(* nursing course applicants.  An effort has been made to make  this *)
(* program as user friendly as possible. *)

const max = 17;
      space = ' ';

type  RawInfo    = string [30];
      LongString = array[1..16]  of string[75];
      Prompts    = array[1..max] of string[40];
      Advise     = array[1..max] of LongString;
      Short      = array[1..max] of LongString;
      Info       = array[1..max] of RawInfo;

var   NoYes   : array[1..3] of char;
      session : RawInfo;
      prompt  : Prompts;
      advice  : Advise;
      brief   : Short;
      I,
      digit,
      AddOne,
      promptType : integer;
      response,
      entry      : RawInfo;
      help       : Info;
      namefile   : text;
      appname    : string[30];
      ans        : array[1..max] of integer;
      answer     : rawInfo;
      line       : string[79];
      eighth     : char;
      failcalc,
      passcalc,
      failsum,
      passsum,
      failscore,
      passscore,
      probfail,
      probpass   : real;
      INum,
      SkipIt     : boolean;


procedure Intro;
(* An introduction to the program that is used to give the program *)
(* name and copyright???!                                           *)


begin
```

```
   clrscr;
   gotoxy(1,5);
   write('     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ');
   writeln('* * *');
   writeln('     *',space:65,'*');
   writeln('     *',space:65,'*');
   writeln('     *',space:65,'*');
   writeln('     *',space:65,'*');
   writeln('     *',space:65,'*');
   write('     *',space:13,'PREDICTION OF APPLICANT SUCCESS PROGRAM');
   writeln(space:13,'*');
   writeln('     *',space:65,'*');
   write('     *',space:7,'  Predicts the likely success of applicants for the');
   writeln(space:7,'*');
   write('     *',space:7,'  Manawatu Polytechnic Comprehensive Nursing Course');
   writeln(space:7,'*');
   writeln('     *',space:65,'*');
   writeln('     *',space:65,'*');
   writeln('     *',space:22,'Written By J A Smith ',space:22,'*');
   writeln('     *',space:65,'*');
   writeln('     *',space:65,'*');
   writeln('     *',space:65,'*');
   write('     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ');
   writeln('* * *');
   writeln;
   writeln;
   write('          Press RETURN to continue ...');
   readln;
end; (* intro *)


procedure StepOne(var session:RawInfo);
(* This is the introduction to the program for subjects in the pilot study. *)
(* It involves selection of the session type the subject is participating in*)

begin
   clrscr;
   gotoxy(5,5);
   writeln('Please type the number of the session type you are ');
   writeln('     participating in.');
   writeln;
   writeln('     1. Computerised');
   writeln('     2. Written');
   writeln('     3. Verbal');
   writeln;
   writeln;
   write('     Session NUMBER is ...');
   readln(session);
end; (*StepOne*)



procedure ChoosePrompt(var PromptType : integer);
```

```
(* A procedure to choose the type of prompt the user wishes to work with.  *)
(* Choices include a full description of the information wanted, a summary *)
(* of the information or a brief prompt for those familiar with the program*)


begin
   clrscr;
   gotoxy(5,5);
   writeln('Please type the number of the desired option in the space');
   writeln('     provided. If you have not used this program before it is ');
   writeln('     suggested that you choose number 1. ');
   writeln;
   writeln;
   writeln('     1. This is the full description. First a summary of the ');
   writeln('        question is given, followed by a full description and then a ');
   writeln('        brief prompt after which you need to enter the required data.');
   writeln;
   writeln('     2. This is a summary description. A summary question is ');
   writeln('        given followed by the brief prompt after which you need to ');
   writeln('        enter the data.');
   writeln;
   writeln('     3. This option includes only the brief prompt. It is ');
   writeln('        designed for those familiar with the program.');
   writeln;
   writeln;
   write('        The chosen option is ...');
   readln(PromptType);

end; (*ChoosePrompt*)



procedure Number;
(* Can be called whenever a response is to be an integer. Will ask that *)
(* the data entered is an integer in the range 0 to 98 *)

begin
   writeln;
   writeln('    You need to enter a number between 0 and 99 here. ');
end; (*number*)



procedure GiveValue(var digit:integer);
(* A procedure to assign a value of 1 to a ''yes'' answer or 0 to a ''n'' *)
(* answer                                                                 *)

begin
   if (entry = 'y') or (entry = 'yes') then
      digit := 1;
   if (entry = 'n') or (entry = 'no') then
      digit := 0;
end; (*GiveValue*)
```

```
procedure description;
(* Provides more detailed information. Can be initiated by both help  *)
(* commands and in PromptType[1]                                      *)

var N : integer;

begin
   advice[1,1] := '1. Type the applicants surname and then their initials.';
   advice[1,2] := '            Don''t use full stops after the initials.';
   advice[2,1] := '2. Type the applicants age in years only. Don''t worry';
   advice[2,2] := '            about months.';
   advice[3,1] := '3. Type the total number of subjects passed. ';
   advice[3,2] := ' ';
   advice[3,3] := '            If U.E. has been attempted more than once, tally the ';
   advice[3,4] := '            number of subjects passed across years.';
   advice[4,1] := '4. Type the number times the applicant has been a sixth';
   advice[4,2] := '            former.';
   advice[5,1] := '5. Has the applicant completed all or part of a nursing qualification?';
   advice[5,2] := ' ';
   advice[5,3] := '            Type ''no'' (or ''n'') if a nursing qualification has NOT';
   advice[5,4] := '            been attempted.';
   advice[5,5] := '   ';
   advice[5,6] := '            Type ''yes'' (or ''y'') if the applicant has completed ';
   advice[5,7] := '            all or part of a nursing qualification';
   advice[6,1] := '6. Was the fact that the applicant had some experience';
   advice[6,2] := '            nursing and had enjoyed the work, given as a reason for';
   advice[6,3] := '            choosing nursing as a career?';
   advice[6,4] := ' ';
   advice[6,5] := '            Type ''yes'' (or ''y'') if this is given as a reason.';
   advice[6,6] := '            Type ''no'' (or ''n'') if this is NOT given as a reason.';
   advice[7,1] := '7. Has the applicant written as a reason for wanting ';
   advice[7,2] := '            to become a nurse a point OTHER THAN :';
   advice[7,3] := '                  - wanting to work with people.';
   advice[7,4] := '                  - career prospects are good.';
   advice[7,5] := '                  - has some experience and enjoys the work.';
   advice[7,6] := '               or  - would find nursing self-satisfying.';
   advice[7,7] := ' ';
   advice[7,8] := '            If they have type ''yes'' or ''y''';
   advice[7,9] := '            If they have not type ''no'' or ''n''';
   advice[8,1] := '8. In this case the work needs to be for a period longer';
   advice[8,2] := '            than 4 weeks.';
   advice[8,3] := '            The following are some examples of the type of job ';
   advice[8,4] := '            the applicant might have held :';
   advice[8,5] := '                   - general nursing student/nurse.';
   advice[8,6] := '                   - nurse aide.';
   advice[8,7] := '                   - enrolled nurse.';
   advice[8,8] := '                   - psychiatric nurse.';
   advice[8,9] := '                   - anciliary staff.';
   advice[8,10] := '                  - community work (of a nursing nature).';
   advice[8,11] := ' ';
   advice[8,12] := '           Type ''yes'' or ''y'' if the applicant has worked in a ';
   advice[8,13] := '           nursing related field for longer than 4 weeks.';
   advice[8,14] := '           Type ''no'' or ''n'' if the applicant has not worked in';
   advice[8,15] := '           a nursing related field or has worked but for a period ';
   advice[8,16] := '           of less than 4 weeks.';
```

```
advice[9,1]  := '9. Type ''yes'' or ''y'' if they have been either a general';
advice[9,2]  := '       nurse or nursing student. Note that this does not include';
advice[9,3]  := '       enrolled nurses.';
advice[9,4]  := ' ';
advice[9,5]  := '       Type ''no'' or ''n'' if they have not.';
advice[10,1] := '10. Has the applicant worked as a nurse aide?';
advice[10,2] := ' ';
advice[10,3] := '        Type ''yes'' or ''y'' if they have worked as a nurse aide.';
advice[10,4] := ' ';
advice[10,5] := '        Type ''no'' or ''n'' if they have not.';
advice[11,1] := '11. Has the applicant been an enrolled nurse?';
advice[11,2] := '  ';
advice[11,3] := '        Type ''yes'' or ''y'' if the applicant been an enrolled';
advice[11,4] := '        nurse.';
advice[11,5] := '        Type ''no'' or ''n'' if they haven''t.';
advice[12,1] := '12. Has the applicant worked as a psychiatraic nurse?';
advice[12,2] := '  ';
advice[12,3] := '        Type ''yes'' or ''y'' if the applicant has worked as a';
advice[12,4] := '        psychiatric nurse.';
advice[12,5] := '        Type ''no'' or ''n'' if the applicant hasn''t.';
advice[13,1] := '13. Has the applicant ever worked in the community in ';
advice[13,2] := '        a nursing capcity?';
advice[13,3] := '  ';
advice[13,4] := '        Type ''yes'' or ''y'' if the applicant has worked in ';
advice[13,5] := '        the community in a nursing capacity.';
advice[13,6] := '        Type ''no'' or ''n'' if the applicant hasn''t.';
advice[14,1] := '14. Has the applicant ever stayed in hospital?';
advice[14,2] := '  ';
advice[14,3] := '        Type ''yes'' or ''y'' if the applicant has stayed in ';
advice[14,4] := '        hospital.';
advice[14,5] := '        Type ''no'' or ''n'' if the applicant has never stayed in';
advice[14,6] := '        hospital.';
advice[17,1] := '17. Type the number of jobs or type of work the applicant';
advice[17,2] := '        WOULD NOT LIKE TO DO. The types of work need to be ';
advice[17,3] := '        completely different.';
advice[15,1] := '15. Has the applicant any dependants?';
advice[15,2] := '  ';
advice[15,3] := '        Type ''yes'' or ''y'' if the applicant does have ';
advice[15,4] := '        dependants.';
advice[15,5] := '        Type ''no'' or ''n'' if the applicant does not have ';
advice[15,6] := '        dependants.';
advice[16,1] := '16. Type in the actual number of dependants the applicant has ';
advice[16,2] := '        between the ages of 6 and 13.';

for n := 1 to 16 do
  writeln(advice[I,N]);
  write('-------------------------------------------------------------');
  writeln('------------------');
  write(prompt[I]);

end; (*description*)


procedure summary;
```

```
(* A procedure to call up the summary descriptions.*)

var N : integer;

begin
   brief[1,1]  := '      Applicants name?';
   brief[2,1]  := '      Age of applicant?';
   brief[3,1]  := '      Number of U.E./sixth form subjects passed?';
   brief[4,1]  := '      Number of times U.E./sixth form has been attempted?';
   brief[5,1]  := '      Has the applicant ever completed all or part of a ';
   brief[5,2]  := '      nursing qualification?';
   brief[6,1]  := '      Reason - has some experience and enjoys the work?';
   brief[7,1]  := '      Reason - other?';
   brief[8,1]  := '      Has the applicant ever worked in nursing or a ';
   brief[8,2]  := '      nursing related field?';
   brief[9,1]  := '      Has the applicant ever been a general nurse/nursing ';
   brief[9,2]  := '      student?';
   brief[10,1] := '      Has the applicant ever been a nurse aide?';
   brief[11,1] := '      Has the applicant ever been an enrolled nurse?';
   brief[12,1] := '      Has the applicant ever been a psychiatric nurse?';
   brief[13,1] := '      Has the applicant ever worked as a community nurse?';
   brief[14,1] := '      Has the applicant ever stayed in hospital?';
   brief[17,1] := '      How many different jobs would the applicant NOT like ';
   brief[17,2] := '      to do?';
   brief[15,1] := '      Has the applicant any dependants?';
   brief[16,1] := '      How many dependants between 6 and 13 does the applicant';
   brief[16,2] := '      have?';

   for N := 1 to 2 do
      writeln(brief[I,N]);
end; (*summary*)


procedure tally(var Failcalc,Passcalc:real);
(* A procedure to calculate the classification scores for each applicant *)
(* using the fail year 1/pass year1 criterion                            *)


var fail    : array[1..max] of real;
    pass    : array[1..max] of real;

begin
   fail[2]  :=   3.252303; pass[2]  :=   2.963068;
   fail[3]  :=   1.899426; pass[3]  :=   2.244800;
   fail[4]  :=  13.819460; pass[4]  :=  11.918920;
   fail[5]  :=   3.005411; pass[5]  :=  -0.365703;
   fail[6]  :=  -0.012252; pass[6]  :=  -1.790690;
   fail[7]  :=   2.660632; pass[7]  :=   1.708202;
   fail[8]  :=   3.788690; pass[8]  :=   1.644330;
   fail[9]  := -11.283700; pass[9]  :=   9.002857;
   fail[10] :=  -1.949925; pass[10] :=  -0.666090;
   fail[11] :=  -0.190565; pass[11] :=   4.649414;
   fail[12] :=  -3.380067; pass[12] :=  -0.746157;
   fail[13] :=  -1.098124; pass[13] :=   1.226727;
   fail[14] :=  -1.628914; pass[14] :=  -0.928559;
```

```
   fail[17] :=    2.451078;  pass[17] :=    1.881007;
   fail[15] :=  -17.926990;  pass[15] :=  -20.422460;
   fail[16] :=   -4.625068;  pass[16] :=   -1.227493;

   failcalc := 0;
   passcalc := 0;
   failcalc := digit*fail[I];
   passcalc := digit*pass[I];
   end; (*tally*)

procedure calculate(var failscore,passscore : real);
(* Procedure to calculate the discriminant scores.                    *)

var fail     : array[1..max] of real;
    pass     : array[1..max] of real;

begin
   fail[2]  :=    3.252303;  pass[2]  :=    2.963068;
   fail[3]  :=    1.899426;  pass[3]  :=    2.244800;
   fail[4]  :=   13.819460;  pass[4]  :=   11.918920;
   fail[5]  :=    3.005411;  pass[5]  :=   -0.365703;
   fail[6]  :=   -0.012252;  pass[6]  :=   -1.790690;
   fail[7]  :=    2.660632;  pass[7]  :=    1.708202;
   fail[8]  :=    3.788690;  pass[8]  :=    1.644330;
   fail[9]  :=  -11.283700;  pass[9]  :=    9.002857;
   fail[10] :=   -1.949925;  pass[10] :=   -0.666090;
   fail[11] :=   -0.190565;  pass[11] :=    4.649414;
   fail[12] :=   -3.380067;  pass[12] :=   -0.746157;
   fail[13] :=   -1.098124;  pass[13] :=    1.226727;
   fail[14] :=   -1.628914;  pass[14] :=   -0.928658;
   fail[17] :=    2.451078;  pass[17] :=    1.881007;
   fail[15] :=  -17.926990;  pass[15] :=  -20.422460;
   fail[16] :=   -4.625068;  pass[16] :=   -1.227493;


   failscore := failsum-47.56702+ln(0.13);
   passscore := passsum-38.08433+ln(0.87);

end; (*calculate*)

procedure probability(var probfail,probpass : real);
(* A procedure to calculate the probability scores of failing/passing the   *)
(* first year of the course                                                 *)

var prob,
    probab,
    probarg,
    probability : real;

begin
   prob        := 0;
   probarg     := 0;
   probability := 0;
   prob := failscore-passscore;
   if prob>0 then
```

```
      prob := -prob;
   probarg := exp(prob);
   probability := probarg/(1+probarg);
   if failscore>=passscore then
      begin
         probpass:= probability;
         probfail:= 1-probability;
      end
   else
      begin
         probpass:= 1-probability;
         probfail:= probability;
      end;
   clrscr;
   gotoxy(6,10);
   writeln('The probability of this applicant failing year 1 of the course is : ',probfail:5:2);
   writeln;
   writeln('     The probability of this applicant passing year 1 of the course is : ',probpass:5:2)

   writeln;
   write('     Press RETURN to continue ...');
   readln;
end; (* probability *)


procedure SetUpFile;
(* procedure to initialise a file at the start of the session*)

var line : string[20];
begin
   clrscr;
   gotoxy(10,5);
   writeln('Welcome to the Manawatu Polytechnic Comprehensive Nursing Course''s');
   writeln('                    "Applicant Success Prediction" program. ');
   writeln;
   writeln('         The first thing you need to do is enter ONE WORD under which the ');
   writeln('         program can store the applicant information you are about to enter.');
   writeln;
   writeln('        * Make a note of the word you choose. *');
   writeln;
   writeln('        * DON''T use a word you or someone else has used before. *');
   writeln;
   writeln;
   writeln('         Please type ONE WORD under which the applicant information ');
   write('        can be stored ...');
   readln(answer);

   if answer<>'c' then
      begin
         assign(namefile,answer);
         rewrite(namefile);
      end;
   if answer='c' then
      begin
         answer := '                    ';
         write('     Please type the name used in the previous session ...');
```

```
      readln(answer);
      assign(namefile,answer);
      reset(namefile);
      while not eof(namefile) do
         readln(namefile);
    end;
  end; (*SetUpFile*)

procedure fileput;
(* A procedure to read information entered by the user into a text file for *)
(* retrieval at a later time. The information is stored under the sessions  *)
(* name.                                                                     *)

begin
  writeln(namefile,appname);
  for I := 2 to max do
    writeln(namefile,ans[I]);
  writeln(namefile,failscore);
  writeln(namefile,passscore);
  writeln(namefile,probfail);
  writeln(namefile,probpass);
  AppName := '                              ';
end; (* fileput *)



procedure Conclusion(failscore,passscore : real);
(* A procedure to write out the final conclusion on the applicant based on *)
(* the information entered*)

begin
  clrscr;
  gotoxy(5,5);
  writeln('From the information that has been entered this applicant is ');
  writeln;
  if (failscore=passscore) or (failscore>passscore) then
    writeln('           UNLIKELY');
  if failscore<passscore then
    writeln('          LIKELY');
  writeln;
  writeln('     to succeed after one year of study.');
  writeln('     NOTE however that this is a TENTATIVE prediction based on ');
  writeln('     the attributes of past applicants. You need to refer to any ');
  writeln('     other sources of information you have before making a final ');
  writeln('     decision.');
  writeln;
  writeln;
  write('     Press RETURN to continue ...');
  readln;

end; (* Conclusion *)


function QuestEight:char;
```

```
(* a function to provide extra information on how to answer Q, 8*)

begin
   writeln('     Note that if this question is answered ''no'' or ''n'', ');
   writeln('     questions 9-13 inclusive will be skipped and also answered''no'' or''n''.');
   writeln;
   writeln;
end; (*QuestEight*)


procedure options;
(* Special procedure to process prompt 1 (applicants name) or to provide   *)
(* extra information if the prompt is number 8*)

begin
   if I=1 then
      begin
         gotoxy(5,5);
         writeln('If you want more information before trying to answer this ');
         writeln('question, type - help ');
         write('-----------------------------------------------------------------');
         writeln('-----------------');
         writeln;
         writeln;
         writeln;
         if PromptType=2 then
            summary;
         write(prompt[I]);
         readln(entry);
         if entry='help' then
            begin
               clrscr;
               gotoxy(7,5);
               description;
               readln(entry);
            end;
         AppName := entry;
         I := I+1;
      end;
   if I=8 then
      eighth := questeight;
end;(* options *)


Procedure StartNum;
(* A procedure to indicate the user the method for getting more information*)

begin
   clrscr;
   gotoxy(5,5);
   writeln('If you want more information on how to ');
   writeln('answer this question, type - 99 ');
   writeln;
   write('-----------------------------------------------------------------');
   writeln('-----------------');
```

```
   writeln;
   writeln;
   if PromptType=2 then
      summary;
   write(prompt[I]);
end; (* StartNum *)

procedure ioCheck;
(* A procedure for checking that an entry is a legitimate digit          *)

begin
   {$I-}readln(digit){$I+};
   if IOresult<>0 then
      repeat
         writeln;
         write('     Please type in a number between 0 and 98 ...');
         {$I-}readln(digit){$I+};
      until IOresult=0;
end; (* ioCheck *)

Procedure ExtraNum(var digit:integer);
(* A procedure to provide extra information on the questions in the block *)
(* if the extra information is requested                                  *)

begin
   clrscr;
   gotoxy(7,5);
   description;
   ioCheck;
end; (* ExtraNum *)


Procedure CatchMistake(var digit:integer);
(* A procedure to catcha n answer that is <0 or >99 *)

begin
   number;
   write(prompt[I]);
   readln(digit);
end; (* CatchMistake *)


procedure SumInitialise(var FailSum,PassSum : real);
(* A procedure to initialise fail and pass sums to 0. *)

begin
   failsum := 0;
   passsum := 0;
end;




Procedure AddSums(failcalc,passcalc : real; var failsum, passsum : real);
```

```
(* A procedure to add the calculated score for the variable to overall scores*)
(* for that person.                                                           *)

begin
   if I=2 then
      SumInitialise(failsum,passsum);
   failsum := failsum+failcalc;
   passsum := passsum+passcalc;
end; (* AddSums *)


procedure StartWord;
(* A procedure to indicate to the user the method for getting more information*)

begin
   clrscr;
   gotoxy(5,5);
   writeln('If you want more information on how to answer ');
   writeln('this question, type - help ');
   writeln;
   write('-------------------------------------------------------------');
   writeln('-----------------');
   writeln;
   writeln;
end; (* StartWord *)


procedure ExtraWord(var entry:rawinfo);
(* A procedure to provide extra information on the current question if it is *)
(* asked                                                                      *)

begin
   gotoxy(7,5);
   description;
   readln(entry);
end; (*ExtraWord*)


procedure CatchError(var entry : rawInfo);
(* A rpocedure to catch any entry that is not yes,y,no or n.  *)

begin
   write('     You need to answer ''yes'' (or ''y'') or ');
   writeln(' ''no'' (or ''n'') for this question.');
   writeln;

   write(prompt[I]);
   readln(entry);
end; (* CatchError *)



procedure check(var I,digit : integer);
```

```
(* A procedure to check the values of I and digit at selected points,      *)
(* i.e. when Jose has made a mistake and can't find where it is!!          *)

begin
   writeln('The value of digit is ...',digit);
   writeln('The value of I is ...',I);
   write('Press RETURN to continue ...');
   readln;
end; (* check *)


procedure CalcCheck;
(* A procedure to check the values of calculations    *)

begin
   writeln('fail ',failcalc,'pass ',passcalc);
   write('press RETURN to continue ...');
   readln;
end; (* CalcCheck *)

procedure CheckSum;
(* A procedure to check the values of the summing of scores  *)

begin
   writeln('fail ',failsum,'pass ',passsum);
   write('Press RETURN to continue ...');
   readln;
end; (* CheckSum *)


procedure again(var PromptType : integer);
(* A procedure which asks the user if they want to continue entering       *)
(* information.                                                            *)

begin
   I := 0;
   clrscr;
   gotoxy(5,7);
   writeln('Do you want to continue entering information in to the program?');
   writeln;
   writeln('     Type ''yes'' (or ''y'') if you do want to add more information.');
   writeln;
   writeln('     Type ''no'' (or ''n'') if you have had enough.');
   writeln;
   write('    Are you continuing ...');
   readln(entry);
   while not ((entry='yes') or (entry='y') or (entry='no') or (entry='n')) do
      begin
         writeln;
         write('     You need to answer ''yes'' (or ''y'') or ');
         writeln(' ''no'' (or ''n'') for this question.');
         writeln;
         write('     Are you continuing ...');
         readln(entry);
      end;
```

```
    if (entry='yes') or (entry='y') then
       begin
          StepOne(session);
          if session='1' then
             begin
                promptType := -1;
                I := -1;
                ChoosePrompt(PromptType);
             end;
          if (session='2') or (session='3') then
             begin
                PromptType := 0;
                I := 0;
             end;
       end;
    if (entry='no') or (entry='n') then
       begin
          close(namefile);
          session := '0';
          promptType := -1;
          I := -1;
          clrscr;
          gotoxy(6,10);
          writeln('     The program has finished.');
          writeln;
          write('     Press RETURN ...');
          readln;
          entry := 'xxx';
       end;
end; (* again *)


procedure WipeFile;
(* A procedure to remove a file used in the session.If the user wishes to    *)
(* continue adding to the same file, the file is saved                       *)

begin
   clrscr;
   gotoxy(5,5);
   writeln('You have finished this session. Do you want to save the ');
   writeln('     applicant information you have just typed in?');
   write('     Type ''yes'' (or ''y'') or ''no'' (or ''n'') ...');
   readln(response);
   while not ((response='yes') or (response='y') or (response='no')
   or (response='n')) do
      begin
         writeln;
         write('     You need to answer ''yes'' (or ''y'') or ');
         writeln(' ''no'' (or ''n'') for this question.');
         writeln;
         write('     Do you want to save the applicant information? ...');
         readln(response);
      end;
   if (response='yes') or (response='y') then
      begin
```

```
        fileput;
        writeln;
        writeln;
        writeln('     The applicant information will be saved under the ');
        writeln('     session name ...',answer);
        writeln;
        writeln;
        write('     Press RETURN to continue ...');
        readln;
        again(PromptType);
      end;
   if (response='no') or (response='n') then
      begin
        writeln;
        writeln;
        write('     The information worked on in this session will NOT be ');
        writeln('saved.');
        writeln;
        writeln;
        write('     Press RETURN to continue ...');
        readln;
        again(PromptType);
      end;
end; (*WipeFile*)


procedure EightSkip(var I:integer);
(* A procedure to skip questions 9 to 14 inclusive if question 8 is answered *)
(* negatively                                                                *)

var n : integer;
begin
   if (entry='no') or (entry='n') then
      begin
        for n := 9 to 14 do
           ans[n] := 0;
        I := 13;
      end;
end; (* EightSkip *)


procedure SixteenSkip(var I:integer);
(* A procedure to skip question 16 if question 15 is answered no.          *)

begin
   if ((I=16) and (SkipIt=true)) then
      begin
        ans[I] := 0;
        I := I+1;
      end;
end; (* SixteenSkip *)


procedure three;
```

```
(* A procedure where all the commands required to elicit the applicant  *)
(* information are asked in the briefest form                           *)

begin
   for I := 1 to max do
      begin
         if (I=1)  then
            begin
               clrscr;
               options;
            end;
         while (I=2) or (I=3) or (I=4) or (I=16) or (I=17) do
            begin
               SixteenSkip(I);
               StartNum;
               ioCheck;
               if digit=99 then
                  ExtraNum(digit);
               if (digit<0) or (digit>99) then
                  CatchMistake(digit);
               if (digit>=0)  and (digit<99) then
                  begin
                     ans[I] := digit;
                     tally(failcalc,passcalc);
                     if I=2 then
                        SumInitialise(failsum,passsum);
                     AddSums(failcalc,passcalc,failsum,passsum);
                     if I=17 then
                        begin
                           calculate(failscore,passscore);
                           probability(probfail,probpass);
                           conclusion(failscore,passscore);
                           wipefile;
                        end;
                     if I<17 then
                        I := I+1;
                  end;
            end;
         while ((I>=5) and (I<=15)) do
            begin
               StartWord;
               if I=8 then
                  options;
               write(prompt[I]);
               readln(entry);
               if entry='help' then
                  begin
                     clrscr;
                     ExtraWord(entry);
                  end;
               if not ((entry='yes') or (entry='y') or (entry='no') or
                  (entry='n')) then
                  CatchError(entry);
               if (entry='yes') or (entry='y') or (entry='no') or
                  (entry='n') then
```

```
                begin
                   SkipIt := false;
                   digit := 0;
                   givevalue(digit);
                   if ((I=15) and ((entry='n') or (entry='no'))) then
                      SkipIt := true;
                   ans[I] := digit;
                   tally(failcalc,passcalc);
                   Addsums(failcalc,passcalc,failsum,passsum);
                   if I=8 then
                      eightskip(I);
                   if (I<16) then
                      I := I+1;
                end;
          end;
       if I<17 then
          I := I-1;
    end;
end; (* three *)

procedure two;
(* A procedure to calculate the values associated with prompt type 2 in the *)
(* summary presentation format                                                  *)

begin
   for I := 1 to max do
      begin
         if I=1 then
            begin
               clrscr;
               options;
            end;
         while ((I=2) or (I=3) or (I=4) or (I=16) or (I=17)) do
            begin
               SixteenSkip(I);
               StartNum;
               ioCheck;
               if digit=99 then
                  ExtraNum(digit);
               if (digit<0) or (digit>99) then
                  CatchMistake(digit);
               if (digit>=0) and (digit<99) then
                  begin
                     ans[I] := digit;
                     tally(failcalc,passcalc);
                     AddSums(failcalc,passcalc,failsum,passsum);
                     if I=17 then
                        begin
                           calculate(failscore,passscore);
                           probability(probfail,probpass);
                           conclusion(failscore,passscore);
                           wipefile;
                        end;
                     if I<17 then
                        I :=I+1;
```

```
                  end;
               end;
            while ((I>=5) and (I<=15)) do
               begin
                  StartWord;
                  if I=8 then
                     options;
                  summary;
                  write(prompt[I]);
                  readln(entry);
                  if entry='help' then
                     begin
                        clrscr;
                        ExtraWord(entry);
                     end;
                  if not ((entry='yes') or (entry='y') or (entry='no') or
                     (entry='n')) then
                     CatchError(entry);
                  if ((entry='yes') or (entry='y') or (entry='no') or
                     (entry='n')) then
                     begin
                        SkipIt := false;
                        digit := 0;
                        givevalue(digit);
                        if ((I=15) and ((entry='n') or (entry='no'))) then
                           SkipIt := true;
                        ans[I] := digit;
                        tally(failcalc,passcalc);
                        AddSums(failcalc,passcalc,failsum,passsum);
                        if I=8 then
                           eightskip(I);
                        if (I<16) then
                           I := I+1;
                     end;
               end;
            if I<17 then
               I := I-1;
         end;
end; (* two *)

procedure one;
(* A procedure to calculate the required values using the full description. *)

begin
   for I:= 1 to max do
      begin
         if I=1 then
            begin
               clrscr;
               extraword(entry);
               AppName := entry;
               I := I+1;
            end;
         while ((I=2) or (I=3) or (I=4) or (I=16) or (I=17)) do
            begin
```

```
               SixteenSkip(I);
               ExtraNum(digit);
               if (digit<0) or (digit>99) then
                  CatchMistake(digit);
               if (digit>=0) and (digit<99) then
                  begin
                     ans[I] := digit;
                     tally(failcalc,passcalc);
                     AddSums(failcalc,passcalc,failsum,passsum);
                     if I=17 then
                        begin
                           calculate(failscore,passscore);
                           probability(probfail,probpass);
                           conclusion(failscore,passscore);
                           wipefile;
                        end;
                     if I<17 then
                        I :=I+1;
                  end;
            end;
         while ((I>=5) and (I<=15)) do
            begin
               if I=8 then
                  begin
                     clrscr;
                     eighth := questeight;
                  end;
               if I<>8 then
                  clrscr;
               ExtraWord(entry);
               if not ((entry='yes') or (entry='y') or (entry='no') or
                  (entry='n')) then
                  CatchError(entry);
               if ((entry='yes') or (entry='y') or (entry='no') or
                  (entry='n')) then
                  begin
                     SkipIt := false;
                     digit := 0;
                     givevalue(digit);
                     if ((I=15) and ((entry='n') or (entry='no'))) then
                        SkipIt := true;
                     ans[I] := digit;
                     tally(failcalc,passcalc);
                     addsums(failcalc,passcalc,failsum,passsum);
                     if I=8 then
                        eightskip(I);
                     if (I<16) then
                        I := I+1;
                  end;
            end;
         if I<17 then
            I := I-1;
      end;
end; (* one *)
```

```
procedure scant;
(* A skeleton set of instructions designed to used in conjunction with      *)
(* written or verbal directions                                             *)

begin
  if I=0 then
    I :=1;
  for I:= 1 to max do
        begin
          if I=1 then
            begin
              clrscr;
              gotoxy(10,10);
              write(prompt[I]);
              readln(entry);
              AppName := entry;
              I := I+1;
            end;
          while ((I=2) or (I=3) or (I=4) or (I=16) or (I=17)) do
            begin
              SixteenSkip(I);
              clrscr;
              gotoxy(10,10);
              write(prompt[I]);
              ioCheck;
              if (digit<0) or (digit>99) then
                CatchMistake(digit);
              if (digit>=0) and (digit<99) then
                begin
                  ans[I] := digit;
                  tally(failcalc,passcalc);
                  AddSums(failcalc,passcalc,failsum,passsum);
                  if I=17 then
                    begin
                      calculate(failscore,passscore);
                      probability(probfail,probpass);
                      conclusion(failscore,passscore);
                      wipefile;
                    end;
                  if I<17 then
                    I :=I+1;
                end;
            end;
          while ((I>=5) and (I<=15)) do
            begin
              clrscr;
              gotoxy(10,10);
              write(prompt[I]);
              readln(entry);
              if not ((entry='yes') or (entry='y') or (entry='no') or
                (entry='n')) then
                CatchError(entry);
              if ((entry='yes') or (entry='y') or (entry='no') or
                (entry='n')) then
                begin
```

```
                SkipIt := false;
                digit := 0;
                givevalue(digit);
                if ((I=15) and ((entry='n') or (entry='no'))) then
                   SkipIt := true;
                ans[I] := digit;
                tally(failcalc,passcalc);
                addsums(failcalc,passcalc,failsum,passsum);
                if I=8 then
                   eightskip(I);
                if (I<16) then
                   I := I+1;
             end;
          end;
       if I<17 then
          I := I-1;
    end;
end; (* scant *)



(****************************************)
(*                                      *)
(*         MAIN PROGRAM                 *)
(*                                      *)
(****************************************)

Begin
   prompt[1]  := '    1. Name ...';
   prompt[2]  := '    2. Age ...';
   prompt[3]  := '    3. U.E. passes ...';
   prompt[4]  := '    4. U.E. attempts ...';
   prompt[5]  := '    5. Nursing qualification ...';
   prompt[6]  := '    6. Reason - experience ...';
   prompt[7]  := '    7. Reason - other ...';
   prompt[8]  := '    8. Worked in nursing ...';
   prompt[9]  := '    9. General nurse/nursing student ...';
   prompt[10] := '    10. Nurse aide ...';
   prompt[11] := '    11. Enrolled nurse ...';
   prompt[12] := '    12. Psychiatric nurse ...';
   prompt[13] := '    13. Community nurse ...';
   prompt[14] := '    14. Hospital stay ...';
   prompt[17] := '    17. Work not liked ...';
   prompt[15] := '    15. Dependants ...';
   prompt[16] := '    16. Dependants aged 6-13 ...';

   intro;
   SetUpFile;
   StepOne(session);
   if session='1' then
      begin
         ChoosePrompt(PromptType);
         while (PromptType>=1) and (PromptType<=3) do
            begin
               if PromptType=3 then
```

```
                three;
              if PromptType=2 then
                two;
              if PromptType=1 then
                one;
            end;
      end;
   if (session='2') or (session='3') then
      scant;
   if PromptType=0 then
      scant;
end.
```

APPENDIX THREE

OFF-LINE MANUAL

INTRODUCTION TO USING THE ATTACHED EXPLANATIONS

The following descriptions can be used to gain extra information on commands given by the program. Two sheets of descriptions are available.

1. A full description of the information to be provided is given. If you have not used the program before these are probably the commands you need to refer to.

2. A summary of the information required is given. If you have used the program but feel you aren't as familiar with it as you could be, use these commands to prompt you for the required information.

FULL DESCRIPTIONS OF EACH OF THE QUESTIONS ASKED

1. Type the applicants surname and then their initials. Don't use full stops after the initials.

2. Type the applicants age in years only. Don't worry about months.

3. Type the total number of subjects passed.
If U.E. has been attempted more than once, tally the number of subjects passed across years.

4. Type the number of times the applicant has been a sixth former.

5. Has the applicant completed all or part of a nursing qualification?

   Type 'yes' (or 'y') if the applicant has completed all or part of a nursing qualification.
   Type 'no' (or 'n') if a nursing qualification has NOT been attempted.

6. Was the fact that the applicant had some experience nursing and had enjoyed the work, given as a reason for choosing nursing as a career.

   Type 'yes' (or 'y') if this is given as a reason.
   Type 'no' (or 'n') if this is NOT given as a reason.

7. Has the applicant written as a reason for wanting to become a nurse a point OTHER THAN :
   - wanting to work with people
   - career prospects are good
   - has some experience and enjoys the work

   or - would find nursing self-satisfying

   If they have type 'yes (or 'y').
   If they have not type 'no' (or 'n').

8. In this case the work needs to be for a period longer than 4 weeks.
The following are some examples of the type of job the applicant might have held :
   - general nursing student/nurse,
   - nurse aide,
   - enrolled nurse
   - psychiatric nurse,
   - ancilliary staff,
   - community work (of a nursing nature).

   Type 'yes' (or'y') if the applicant has worked in a nursing related field for longer than four weeks.
   Type 'no' (or 'n') if the applicant has not worked in a nursing related field or has worked but for a period of less than four weeks.

9. Type 'yes' (or 'y') if they have been a general nurse or nursing student. Note that this does not include enrolled nurses.
   Type 'no' (or 'n') if they have not.

10. Has the applicant worked as a nurse aide?

    Type 'yes' (or 'y') if they have worked as a nurse aide.
    Type 'no' (or 'n') if they have not.

11. Has the applicant been an enrolled nurse?

    Type 'yes' (or 'y') if the applicant has been an enrolled nurse.
    Type 'no' (or 'n') if they have not.

12. Has the applicant worked as a psychiatric nurse?

    Type 'yes' (or 'y') if the applicant has worked as a psychiatric nurse.
    Type 'no' (or 'n') if the applicant has not.

13. Has the applicant ever worked in the community in a nursing capacity?

    Type 'yes' (or 'y') if the applicant has worked in the community in a nursing capacity.
    Type 'no' (or 'n') if the applicant has not.

14. Has the applicant ever stayed in hospital?

Type 'yes' (or 'y') if the applicant has stayed in hospital. Type 'no' (or 'n') if the applicant has never stayed in hospital.

15. Has the applicant any dependants?

Type 'yes' (or 'y') if the applicant does have dependents.
Type 'no' (or 'n') if the applicant does not have dependents.

16. Type the actual number of dependents the applicant has between the ages of 6 and 13.

17. Type the number of jobs or the type of work the applicant WOULD NOT LIKE TO DO.

Note that the types of work need to be completely different.

BRIEF DESCRIPTIONS OF EACH QUESTION ASKED

1. Applicants name.

2. Age of applicant.

3. Number of U.E./sixth form subjects passed.

4. Number of times U.E./sixth form has been attempted.

5. Has the applicant ever completed all or part of a nursing qualification?

6. Reason - has some experience and enjoys the work.

7. Reason - other?

8. Has the applicant ever worked in nursing or a nursing related field?

9. Has the applicant ever been a general nurse/nursing student?

10. Has the applicant ever been a nurse aide?

11. Has the applicant ever been an enrolled nurse?

12. Has the applicant ever been a psychhiatric nurse?

13. Has the applicant ever worked as a community nurse?

14. Has the applicant ever had to stay in hospital?

15. Has the applicant any dependents?

16. How many dependants between 6 and 13 does the applicant have?

17. How many different jobs would the applicant NOT like to do?

APPENDIX FOUR.

FINAL QUESTIONNAIRE

Please answer the following questions as fully as possible.

Note that an honest opinion of the program you have just used would be appreciated.

1. What did you like about the program you have just used?

----

2. What did you dislike about the program you have just used?

----

3. How could this program be improved to make it easier to use?

----

4. Please make any other comments you would like to in the space provided below.

Thank you for the time you have spent helping with this study.

If you would like any information on the results obtained please leave your name with the experimenter. A summary of the results will be sent to you as soon as it is available.

APPENDIX FIVE.

Application Form Coding Form

VARIABLE	CODE	
Marital Status	00 = Single	01 = married
Age	Actual Number	
Sex	01 = male	02 = female
Number of secondary schools attended	Actual Number	
Number of School Certificate subjects attempted	Actual Number	
Number of School Certificate subjects passed	Actual Number	
Average School Certificate mark	Actual Number	
Number of U.E. subjects passed	Actual Number	
Number of times U. E. attempted	Actual Number	
Mean Sixth Form Certificate mark	Actual Number	
Number of bursary subjects attempted	Actual Number	
Average Bursary mark	Actual Number	
Nursing qualifications	00 = No	01 = Yes
University qualifications	00 = No	01 = Yes
Other qualifications	00 = No	01 = Yes
Number of different jobs held	Actual Number	

Reasons for Wanting to be a Nurse

- work with people	00 = No	01 = Yes
- career prospects	00 = No	01 = Yes
- has gained some experience and enjoys the work	00 = No	01 = Yes
- would be self satisfying	00 = No	01 = Yes
- other	00 = No	01 = Yes

When did the applicant decide to become a nurse?	00 = within the last five years 01 = more than five years ago.

Reasons for choosing Comprehensive nursing

- course content is better	00 = No	01 = Yes
- Future prospects are wider	00 = No	01 = Yes
- feels particularly suited to this form of training	00 = No	01 = Yes
- other	00 = No	01 = Yes
Has the applicant to any other nursing training programme	00 = No	01 = Yes

Has the applicant ever worked in a any of the following nursing related jobs

- general nursing student	00 = No	01 = Yes
- nurse aid	00 = No	01 = Yes
- enrolled nurse	00 = No	01 = Yes
- psychiatric nurse	00 = No	01 = Yes
- community nurse	00 = No	01 = Yes
Has the applicant ever been a patient in hospital	00 = No	01 = Yes

How many people provided the applicant with information on a nursing career.	Actual Number
Number of leisure activities in a typical week	Actual Number
How many jobs would the applicant not like to do	Actual Number

Does the applicant have any dependants

- 0 to 5 years old	00 = No	01 = Yes
- 6 to 13 years old	00 = No	01 = Yes
- 14 to 16 years old	00 = No	01 = Yes
- older than 17 years	00 = No	01 = Yes
Have arrangements been made for the care of dependants	00 = No	01 = Yes
Does the applicant have a criminal record	00 = No	01 = Yes

APPENDIX SIX

TRIAL INSTRUCTIONS.

You are participating in a computerised trial.

You need to

1. Read through the attached material.

2. Work through the program once you are ready to.

3. Use the HELP facilities in the program if you need more information to answer a question.
   Refer to the top of the screen to find out how to get help information.

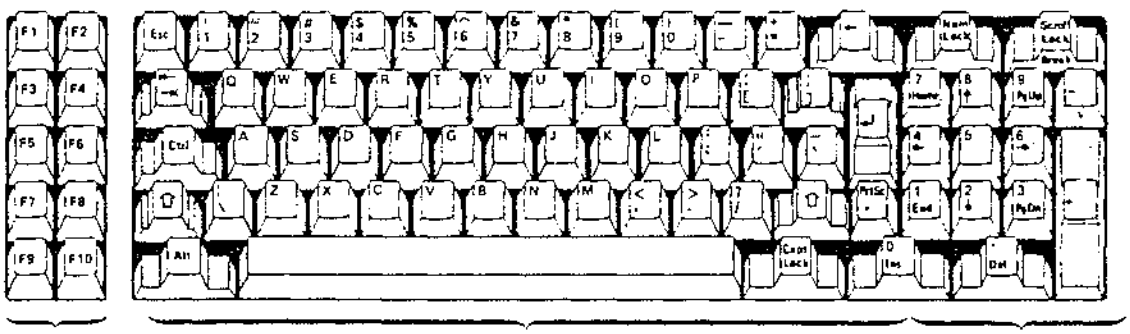
INTRODUCTION

The purpose of this introduction is to provide information for the first time user of a micro computer. Simple explanations of common jargon used by computer users is given.

The Keyboard

The keyboard of the micro(s) you will be using is laid out like that of a typewriter, but there are some differences.
Have a look at the keyboard while you read through the following

The Keyboard



descriptions.
Before you start to work on the computer it is probably a good idea to explain a few points about computer use.
One of the quirks of computers is the need to "send" information.
This means that after you have typed each piece of data asked for, you need to push the RETURN key. This is a big key on the left hand side of the keyboard. It has a crooked arrow on it.

ie.

Have a look at the keyboard to make sure you can see the RETURN key.

Remember to press this key when when you have finished typing a response to a question, it sends your response to the computer (it's function is very like that of a carriage return on a typewriter).

The other key that is important is the left facing arrow above the RETURN key. This key is also known as the "backspace" key. It can be used to delete characters and move the cursor left BEFORE you press the RETURN key to send your answer to the computer.

For example if the computer asks
    Applicants age?
You are asked to enter the applicants age. If by mistake you type

    '91'

instead of

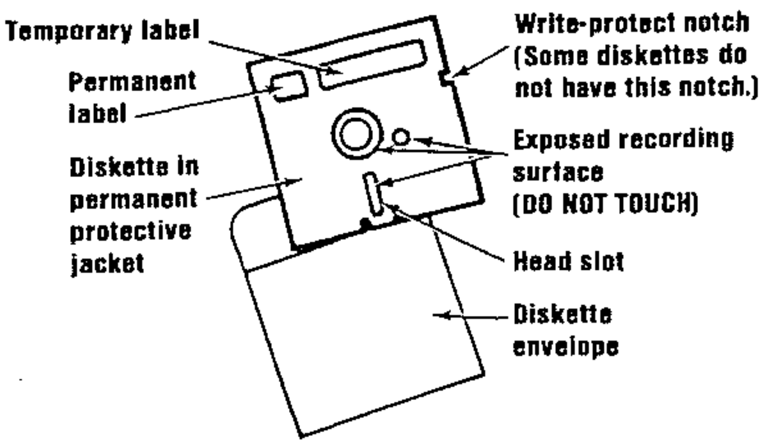
    '19'

you can use the backspace key twice to erase the '91'. You are then able to type the correct answer of '19' in its place.

You should not need to use any keys other than 'y', 'e', 's', 'n', 'o', '1' '2', '3', '4', '5', '6', '7', '8', '9', '0', RETURN and 'backspace' (unless indicated) in the program you are about to use. Note the difference between the '0'(zero) and 'o'/'O'(oh) keys. All other keys on the keyboard can in effect be ignored.

Floppy Diskettes.

The micro computer(s) (micro(s)) you will be using uses 5 and 1/4 inch diskettes for storing information. Refer to the diagram



included.
The permanent protective jacket contains a flexible diskette that is

coated with a magnetic substance. When in use, the diskette spins inside the jacket. The read/write head of the drive comes into contact with the recording surface through the long hole in the protective jacket, the "head slot". Information is read onto or read from the magnetic surface of the diskette, similar to the way an ordinary tape-recorder operates.

Floppy diskettes are fragile and sensitive. They should be looked after with care.

1. Don't bend them.

2. Do not touch the exposed recording surfaces.

3. Protect them by always putting them back in their envelopes as soon as you remove them from the diskette drives.

4. Store them away from heat and from magnetic field sources such as telephones, dictation equipment and electronic calculators.

5. Do not remove them while the drive is running - you will be able to hear the motor running and the red light above the drive will be on.

OVERVIEW OF WHAT THE PROGRAM DOES.

The program you will be using represents a new way of processing nursing course applicant information.

The main aim is to analyse each applicants information in such a way that when it has all been entered into the program an indication can be given as to whether the applicant is likely to succeed in the first year of the Polytechnic Nursing Course.

This is achieved by

1) Using information in an applicants application form to answer 17 questions provided by the computer.
   Note that in some instances not all 17 questions will be asked, i.e. some may be skipped.
   You need to answer each question AND send the answer to the computer using the RETURN key.

2) Using this information the computer calculates two probability scores.
   One represents the probability that the applicant will fail year 1.
   One represents the probability that the applicant will pass year 1.

3) These two scores are compared.
   If the probability of failing is greater than the probability of passing, the computer reports that the applicant is UNLIKELY to succeed in the year 1.
   If the probability of passing is greater than the probability of failing, the computer reports that the applicant is LIKELY to succeed in the year 1.

The main task of the program is now complete.
The next two questions asked each require you to make a decision.

First, do you want to save this last applicants information.

Usually you WILL, i.e. you will type 'yes'

BUT

if you realise you have made a mistake typing in the applicants information you would type 'no' AND then re-enter the applicants data.

Finally, do you want to continue entering information.

If you have more applicant information to enter, type 'yes'.
The program will start again.
You will need to select the type of trial you are participating in. This will be the same as previously.
If you are participating in a computerised session you will be able to select a new prompt-type, or if you prefer, continue with the same prompt-type.

If you have finished, type 'no'.

Starting work with the computer.

1. Before you do anything read through the following instructions so you are familiar with the task to be performed.
   Note that the instructions differ slightly for the KAYPRO and I.B.M. computers.
   Have a look at which make of computer you are using and read the appropriate instructions.

2. Switch the computer on at the wall and at the back of the machine.

3. If the disk-drive door is not already open (it should be), open it.
   If you are working on a KAYPRO, rotate the disk drive arm 90 degrees anti-clockwise.
   If you are working on an I.B.M. lift the disk drive door.

4. Insert the diskette into the disk drive.

   On a KAYPRO the disk drive will be vertical in which case the diskette should have its label facing the screen and side notch pointing downwards.

   If the machine you are working at is an I.B.M. the slots of the disk drives will be horizontal. In this case the left hand disk drive should be used. Insert the diskette with its label facing upwards and notch on the left hand side.

5. Shut the disk drive door. By either rotating the arm, or lowering the door.

6. The computer will take a little time to prepare itself to receive your instructions.

7. When ready, work through the program using the applicant data provided in the attached forms.

8. Remember to press RETURN after you have entered each piece of information.

9. When you have finished the program remove the diskette and store it in it's protective jacket.
   Leave the disk drive door OPEN.
   Switch the machine off at it's rear.
   Switch off the power at the wall.

10. Now you are ready to begin.

GET - the ten Nursing Course Application Forms. These contain the data you will be typing into the computer.

- ensure that you have read the introductory comments.

- ensure that you have read the over view of the task you will be performing

FINAL INSTRUCTIONS

Now you are ready to begin.

GET - the ten Nursing Course Application Forms. These contain the data you will be typing into the computer.

- ensure that you have read the introductory comments.

- ensure that you have read the over view of the task you will be performing manawatu polytechnic

Please affix PHOTOGRAPH here

APPLICATION FORM FOR COMPREHENSIVE NURSING

CLOSING DATE FOR APPLICATIONS: 30 Sept 1985

SURNAME: ____________________ MR/MISS/MRS/MS

GIVEN NAMES: ____________________ PREFERRED NAME: ____________

DATE OF BIRTH: ____________ AGE AS AT 1/2/86: ______ YRS ______ MTHS

(N.B: Applications are not accepted unless applicant is 17 years by 1/2/86. Applications from those who are 17, but not 18 by 1 February 1986 are accepted, but MAY NOT be processed unless the class has not been filled by mid-January.)

CITIZENSHIP: ____________________ SEX: ________

ETHNIC ORIGIN: (Tick the group with which you most readily identify)
(N.B: This information is for statistical purposes only)

N.Z. Maori _____ Pacific Islander _____ European _____

Other: (Please specify) ____________________

MARITAL STATUS:

Divorced ____ Married ____ Separated ____ Single ____ Widowed ____

No. of Dependants ______ Ages of Dependants ____________

HOME ADDRESS: ____________________

____

____________________ PHONE: ____________

NEXT OF KIN: ____________________

ADDRESS: ____________________

____

____________________ PHONE: ____________

Office use only

SECONDARY SCHOOLS ATTENDED:

NAME OF SCHOOL	COMMENCING		ENDING	
	MTH	YEAR	MTH	YEAR

Total Duration of Secondary Education: ______years ______months

If not attending school at present, state date of leaving________

SUBJECTS STUDIED

List 5,6,7 Form subjects	School Cert mark		6th Form Cert		U.E. (enter "A" if accred)		Univ. Bursary or Schol.	
		YR		YR		YR		YR

SUMMARY

Total No. of School Cert. subjects passed (ie 50% or more) ______

Mean (average) mark of all School Cert. subjects passed ______

Total No. of U.E. subjects passed (ie 50% or more) ______

If appropriate, please notify our Grey Street Office as soon as accrediting results are received by either phoning 85-673 or forward result slip to Manawatu Polytechnic, Private Bag, Palmerston North.

OTHER EDUCATIONAL QUALIFICATIONS:

INSTITUTION	SUBJECT/COURSE If Univ. specify papers	YEAR	GRADE

PREVIOUS OCCUPATIONAL EXPERIENCE:

EMPLOYER	NATURE OF WORK/POSITION	PERIOD OF EMPLOYMENT FROM	TO	REASON FOR LEAVING

DO YOU HOLD ANY NURSING QUALIFICATION? Yes ☐

No ☐

IF YES, SPECIFY: ______________________________

____

N.B: DOCUMENTARY EVIDENCE OF YOUR HIGHEST QUALIFICATION IS REQUIRED

Please supply names and addresses of two referees (not relations). One must be your present employer, if currently employed, OR the Principal of the secondary school/educational institution you are attending.

1) NAME OF PERSON TO CONTACT: ______________________________

NAME OF SCHOOL OR CURRENT EMPLOYER: ______________________________

ADDRESS: ______________________________

____

2) NAME: ______________________________

ADDRESS: ______________________________

____

PREFERENCE LIST FOR SPECIFIC COURSES:

Please note that preference is given to applicants from within our region.

> N.B: Unless applicants who live outside our region provide an exceptional reason for attending Manawatu Polytechnic, their application will not be processed unless the class cannot be filled from within our region.

You should also be aware that Technical Institutes offering nursing courses (i.e: those listed below) share lists of names and addresses of applicants to ensure that those to whom they offer places are not holding more than one offer.

Write in order your preference from (1) most preferred to (3) least preferred from the following list:

Auckland Technical Institute; Christchurch Polytechnic; Hawke's Bay Community College; Manawatu Polytechnic; Manukau Technical Institute; Nelson Polytechnic; Northland Community College; Otago Polytechnic; Southland Community College; Taranaki Polytechnic; Waiariki Community College; Waikato Technical Institute; Wellington Polytechnic.

1. ______________________________

2. ______________________________

3. ______________________________

> *N.B: Students who live in a town offering a comprehensive nursing course are eligible for boarding bursaries ONLY after they have been rejected for that course and gain acceptance in another. Evidence of that rejection is required before a Boarding Bursary will be given.*

Office use only

QUESTIONNAIRE FOR APPLICANTS FOR NURSING COURSE

1) Why do you want to become a nurse?

2) When did you decide to become a nurse?

3) Have you applied in previous years for admission to any other nursing course or programme? YES / NO (delete one)

If yes, please specify which, and state the outcome of your application.

4) Do you intend making application to any other educational or training programme? YES / NO (delete one)

If yes, specify the programme and institution.

5) Do you have any close relations or immediate family who are health professionals? YES / NO (delete one)

If yes, state their relationship to you and the name of the profession to which they belong.

6) List your sources of information about nursing.

Office use only

7) How did you hear about this particular course? (i.e: Manawatu Polytechnic Comprehensive Nursing course)

8) Have you ever worked, voluntarily or for remuneration in a hospital or community health service? YES / NO (delete one)

If yes, state where, when and what you did.

9) Have you ever been a patient in a hospital? YES / NO (delete one)
If yes, when and for how long.

10) List your leisure activities in a particular week.

List any community activities in which you are currently involved.

If you are accepted for the course, which of the above activities would you continue while you are a nursing student?

11) What sort of work would you NOT like to do? (Please list)

12) Do you find it easy to approach new people? YES / NO (delete one)

13) Do you prefer to work or study alone/or with others. (delete one)

14) Do you spend a lot of time within groups? YES / NO (delete one)

15) When you are in a group do you (tick those that apply to you)

... always participate actively?

... participate at an average level?

... listen to views of others?

... stand up for your own views?

... commonly take a leading role?

... feel you contribute something worthwhile?

... find it easy to express your ideas?

16) List, in order of importance, the ten characteristics which you feel best describe yourself.

17) If accepted for the course at Manawatu Polytechnic, what arrangements will you make regarding accommodation? (tick one)

... Living at Home

... Other Living arrangements already made

... Seeking hostel

... Seeking flat

... Seeking private board

18) If you have dependents, what arrangements will you make for their care?

19) Have you ever been convicted of a criminal offence? YES / NO
If yes, state nature of offence and conviction date.

TO THE BEST OF MY KNOWLEDGE THE INFORMATION PRESENTED IN THIS QUESTIONNAIRE IS ACCURATE.

SIGNATURE OF APPLICANT: ______________________

DATE: ______________

Over the page is a copy of the form supplied to the referees you name.
We would like YOU to fill this copy in as you think others see you.