The University of Newcastle upon Tyne

Department of Computing Science

A Knowledge–Based Decision Support System for Computer Disaster Prevention In IT Centres

by

Tawfig Y. Danish

PhD Thesis

December 1994

Acknowledgements

I would like to express my sincere thanks to my supervisor, Dr Lindsay Marshall, for his guidance and supervision during the preparation of this thesis, and for his assistance in supplying a great deal of literature related to the subject. I also would like express my deep thanks to the Faculty of Science for provisions of facilities, and to the Computing Department staff for their help. Special thanks goes to Dr Martin Mclauchlan and Dr B. N. Rossiter for their useful comments during the course of study, and also for Shirley Craig at the Computing library for her assistance in providing many related materials and references.

I acknowledge, and am in gratitude to, the Ministry of PTT in Saudi Arabia for funding this project and providing the necessary financial support. I also extend my best gratitude to the Saudi Arabian Ministry of Higher Education for giving me the opportunity to study in the UK.

Finally, but not least, I wish to express my utmost thanks to my family for their generous sacrifice, inspiration and moral support, and for their prayers, without which this would not have been possible.

Abstract

In analysing the extent to which adequate research work may have been undertaken in the specific area of computer disaster prevention, it was found that little work had been done. In the real-life situation, it was also concluded that, in the vast majority of cases, no adequate disaster prevention controls were in use at IT installations. Guidance for the analysis and management of the risk associated with computer disasters, as a result, has also been inadequate and lacking in uniformity, specially in the areas of risk identification and risk entities interactions and relationships.

This research has involved developing and delivering a methodology which would help IT risk managers in implementing effective computer disaster prevention controls. A knowledge based system (KBS) approach has been used to build a prototype system which provides full support in this important area of decision making, and to show how the representation of risks can be handled.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Background

The term 'Computer Disaster' has been defined as an interruption in Data Processing services (Swank, 1981). Computer Disasters may be rare events, but can, in extreme cases, result in bankruptcy by causing an organisation to lose the ability or opportunity to continue trading (Krauss, 1980).

According to Parker (1981) and Carroll (1984) external disasters may be intentional (e.g. from espionage and other malicious motives); accidental (e.g. from mistakes) or natural (e.g. flood, earthquake, weather conditions). These examples of external disasters showed that service interruptions (sometimes referred to as denial of service) may result from the following kinds of impact (or risk exposures):-

a) directly from damage to IT facilities, e.g. buildings, plant rooms, computer equipment etc.; or

b) indirectly from loss of electrical power, water supply, communications, key staff, access to working areas etc.

These types of direct and indirect impacts have not received adequate attention, even though the total losses resulting from a significant interruption of service can be disastrous. They typically include much more than just the costs of hardware

and software. Table 1.1 illustrates the kinds of loss which may result if a commercial organisation is denied the contribution which IT normally makes to its operations.

Table 1.1 - Types of Loss and Consequences

Type of Loss	Consequence
Direct Financial Losses	Destruction of Equipment and Facilities
	Loss of Sales
	Loss of Production
Indirect Financial Losses	Long-term Loss of Customers
	Extra Payments to Staff
	Uncollected Receivables
	Undetected Fraud
	Payment of Fines and Damages
Loss of Control	Loss of Integrity of Data
	Erroneous Business Decisions
Embarrassment to the Organisation	Unfavourable Media Exposure

(Goldblum of Butler Cox, sponsored by Amdahl, 1982)

The amount of loss arising increases with the duration of the interruption. This duration will also increase depending upon the extent of the direct and indirect impacts mentioned above. For example, the loss of a building may represent a worst case scenario, while loss of power would be less significant. The consequences of service interruption caused by the loss of the building would be catastrophic because re-construction may take several months, with a corresponding delay in the resumption of normal operations. This scenario would not arise if adequate protective measures were taken.

Of course, some organisations may already have contingency plans, and may achieve some level of recovery from disaster, but only if the risk does not exceed a reasonable level. Historically, where means of protection have been adopted, they have usually been in the form of Insurance cover or Disaster Recovery services.

Insurance cover is possible to some extent, and may be regarded as the last line of defence against unexpected loss or damage (Shain, 1989). Traditionally, organisations have only insured their IT installations against physical damage, but this cover only provides financial compensation for replacing damaged facilities, and does not replace the facilities themselves. That is for the organisation to do (Pinder and Hoover, 1990). Recently, specialised computer insurance policies have become available to cover some of the consequential losses which are illustrated in the table above. These policies may not always be sufficient to compensate for other consequences, such as unfavourable media exposure, or loss of goodwill (Haack, 1984; Palmer and Potter, 1989). Most of these policies are expensive, and difficult for both the organisation and the insurer to assess, because of the lack of empirical data. For this reason, the insurance industry's approach is cautious (Shain, 1989: Smith, 1989; Pinder and Hoover 1990). The premium for insurance against business interruption resulting from computer disasters could, for example, be almost four times the premium for insuring the IT property against loss or damage (Roman, 1986). The cost may increase significantly, depending upon the level of existing security measures at the site (Palmer and Potter 1989). Organisations that have disaster recovery plans may pay lower insurance premiums. Therefore, the primary control of risk through a disaster recovery plan may be preferred over insurance arrangements.

Although organisations have recognised the limitations of insurance cover, and the large consequential losses which may arise from Computer Disasters, only a minority of them have even considered Disaster Recovery Plans as a means of

ensuring service continuity. These plans enable service resumption after a disaster. They normally include some form of standby arrangement under which service can be resumed at an alternative site while the lost facility is replaced. Although Disaster Recovery and Standby Arrangements provide relief from the loss of some facilities, these measures also have several limitations in:-

1. providing only temporary help;
2. suffering inevitable delays in resuming lost services;
3. usually being less than a perfect match to the facilities they replace; and
4. incurring costs which make no contribution to the replacement of lost facilities, which has to be done.

The fact is, then, that even a combination of Insurance and Disaster Recovery service leaves an IT installation in a less than ideal position to ensure service continuity.

1.2 Problem Description

At risk in any computer installation are substantial investments in assets which include equipment, software and data, as well as expensive staff with scarce specialist training. The greatest organisational value of most IT installations, and the only reason for their existence is, of course, in the service which they contribute. Although IT services have become critical to the operation of organisations, more attention has been given to disaster recovery than to disaster prevention (Faithfull and Watt, 1991). The principles of Computer Disaster Prevention differ from those of Computer Disaster Recovery. Disaster Recovery seeks to minimise the loss, and recovery time, from events which have already

happened. Disaster Prevention, on the other hand, seeks to avoid problems in the future by providing adequate protective measures, so that consequential losses are minimised (Faithful and Watt, 1991).

A Disaster Prevention Policy, implemented within a structured plan, helps an organisation's management to specify which preventive measures are required to avoid or reduce the risks of Computer Disasters, and how those measures can be justified.

This study advances means of minimising the risks of disaster by the implementation of Disaster Prevention Policies, rather than just measures for recovery after a disaster, or the transfer of risks to insurance. This is not to say that insurance and disaster recovery measures are completely dismissed: in fact, their relevance in some situations is acknowledged.

In practice, the following situations may arise:-

a) an IT installation at high risk of fire may only merit insurance cover, even with the delays in service resumption which that implies, if the service it provides is of low value; and

b) an installation with a high flood risk, even if its service contribution is high, may only merit a Disaster Recovery service if the costs of flood prevention measures are so high as to be unjustifiable.

Selecting appropriate measures for a given site is part of an overall risk management exercise, in which risks to IT installations must first be defined and analysed (Campbell and Sands, 1979; De Backer, 1981; Hoffman, Michelman and Clements, 1985; Gilbert, 1989). In other words, the selection of risk management actions (including disaster prevention measures) needs to be based on justifying the costs of these measures by systematically identifying, analysing and assessing

actual risks. This selection process also needs to compare the costs and benefits of the full range of measures applicable to a given site.

The successful conduct of an effective risk management programme, as part of implementing a disaster prevention policy, is critically dependent upon identifying and analysing all of the associated risk entities (e.g. assets, threats, vulnerabilities and counter-measures) and their interactions (Mayerfield, 1988; Katzke, 1988). Clearly, these interactions are inherent in the relationships between these risk entities, and they influence the extent of risk exposure (e.g. to destruction of assets and/or interruption of service). These relationships between risk entities demand that the following factors, which could contribute to risk exposure, need more thorough identification and analysis than has been achieved in previous work (NBS Special Publication 500-133, 1985).

1.2.1 Factors which contribute to Risk Exposure

a) Geographical factors in the selection of suitable site locations for IT installations, such as elevation and the proximity to hazards.

b) Threats arising outside the immediate accommodation of IT installations, which may directly or indirectly affect the installation, such as high rainfall, river flooding.

c) The service contribution of assets (and controls) for the IT accommodation, including those located outside, such as the building, its access, power supplies and barriers.

d) The importance of personnel safety as a major component in ensuring service continuity.

e) The identification and selection of counter-measures that are appropriate against specific types of risk exposure; such as compliance with building standards to protect against destruction, and standby generators for service continuity.

The form of analysis needs to be sufficiently comprehensive that it takes account of the interactions between risk entities, and this is an area which has previously been acknowledged to be unclear (Clark, 1989). It was found necessary, as part of this research, to design a method for analysing these interactions and influences. The clarification of that area, which this study addresses, is therefore new.

1.3 Objectives of the Research

The central theme of this study has therefore been to address more fully the protection of IT services from interruptions caused by computer disasters arising from threats which originate outside the immediate accommodation of IT installations. This is done by enabling IT Risk Managers to identify and justify appropriate Disaster Prevention measures. To provide this, the following objectives were adopted:-

a) to develop a structured methodology to assist the identification, analysis, assessment and management of the risks of computer disaster;

b) to incorporate this methodology in a system suitable for use by IT Risk Managers; and

c) to ascertain and demonstrate the best available way of achieving a) and b).

As a result, organisations will be able to enjoy real operational and competitive advantages from the ability to continue trading with a minimum of IT service interruption.

1.4 The Organisation of the Research

To achieve the above set of objectives, the following main groups of research work were undertaken. As the research work progressed, it was found that each stage of research "fed into" the next, so that a cumulative effect resulted.

a) The available literature was studied to ascertain what ideas or other assistance with achieving the objectives could be gleaned from previous work. This review is reported in chapter 2.

b) The findings of the literature review were used to help determine a framework (described in chapter 3) which would handle the required IT risk management phases.

c) A methodology was formulated, based on the framework, which could be the foundation (described in chapter 4) for the decision support system for IT Risk Managers.

d) Available technologies were studied (as described in chapter 5), to ascertain which would be the most suitable to provide the framework and methodology in a form which would be usable by IT risk managers.

e) A prototype Knowledge Based Decision Support System for Computer Disaster Prevention was developed, as described in chapter 6, to demonstrate the suitability of the technology, and to

prove the validity of the methodology, in a system which can be used by IT Risk Managers.

1.5 Contributions of the Research

The results of this research, reported in detail in chapters 2 to 6 below, can be summarised as follows.

a) The Literature Review discovered some ideas, concepts and earlier methodologies which have contributed to the foundation on which the solution recommended was developed (e.g. CCTA, 1987).

b) A framework and a methodology for the risk management of computer disasters were developed, which provide the thoroughness in taking account of interactions and inter-dependecies between risk entities which had not previously been fully achieved.

c) The review of technologies to find a suitable basis for implementing the risk management methodology concluded that a Knowledge Based Systems approach would best achieve the objectives.

d) A Prototype Knowledge Based Decision Support System was developed which both demonstrates the efficacy of the methodology, and the ability of KBS to provide the implementation vehicle.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

The primary objectives of this chapter are to:-

a) report the documented historical background to the subject of computer disasters,

b) examine previous innovations in the development of computer disaster prevention and risk management practices (this examination is presented as three major development milestones), and

c) record the material which was studied covering the use of Knowledge Based Systems technology in this area.

2.2 Historical Background

This section reports the documented historical background to the subject of computer disasters. It includes the results of various surveys by reputable organisations and, as will be seen, these results justify the central theme in this

thesis: that the area of computer disaster prevention is one which warrants thorough treatment.

2.2.1 The Impact of Computer Disasters

The first study of the impact of computer disasters was reported by Aasgard et al (1979), when they examined how long different business functions would be able to operate without their IT systems. For 36 companies, only 28% of their operational activities would still be functioning after 5.5 days without an IT system. Finance companies in the sample estimated that only 13% of their operations would still be functioning after 5.5 days without IT systems. In another study, Christensen & Schkade (1987) concluded that 75% of organisations would have reached critical or total loss within two weeks of losing computer support. Loss of revenue, and additional costs, become substantial as the outage continues. One financial industry respondent stated "We would be out of business after one week." A manufacturing industry representative responded "After 25 days of loss of IT service, our chances of coming back as a corporation are about 20%." Organisations would experience significant loss of revenues in the event of loss of IT services. Reported loss of revenues reached as much as $400 million per day. The average estimated loss is 25% of daily revenues by day 7, rising to 42% by day 30. A recent study by Price Waterhouse (1990) showed a typical financial impact for UK based finance and manufacturing organisations as shown in Tables 2.1 and 2.2 below.

Table 2.1 - Financial Impacts of Computer Disasters - Finance House

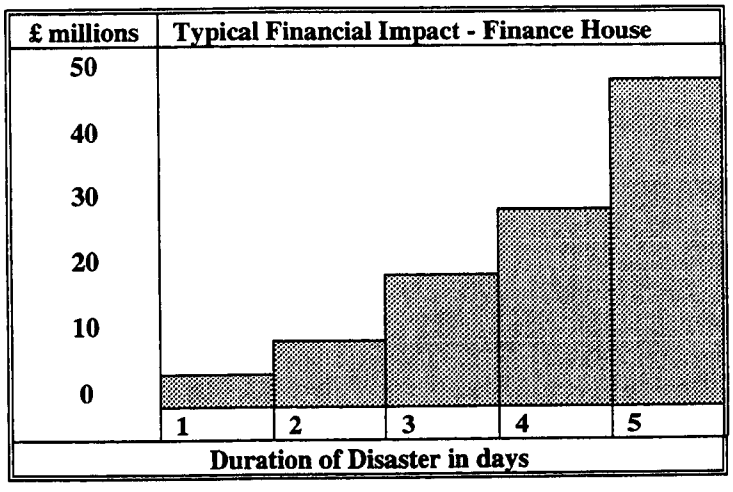
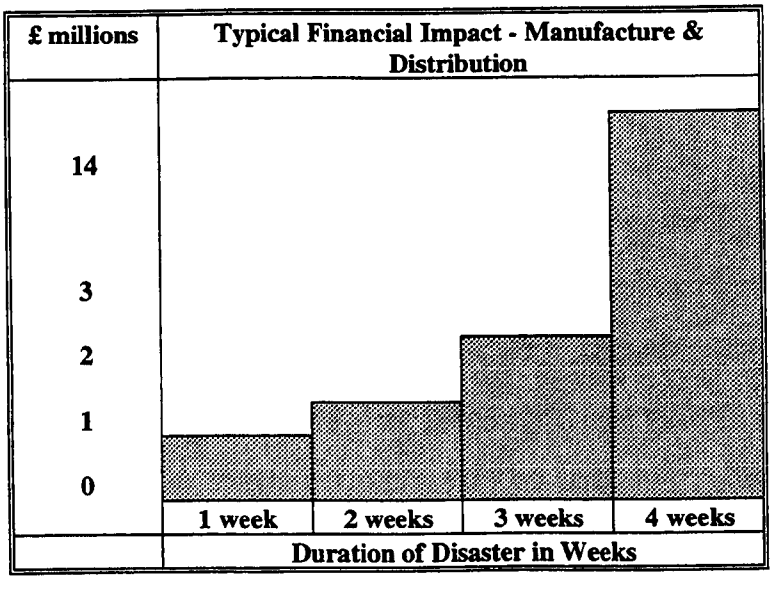
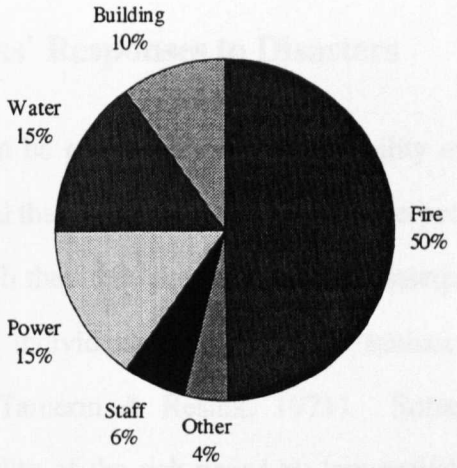


Table 2.2 - Financial Impacts of Computer Disasters - Manufacture & Distribution



A UK government report showed that half of the 70 IT installations in 37 government ministries and departments had inadequate disaster plans, or none at all. 12 had taken no action at all, despite two government installations suffering disasters on average each year (National Audit Office and CCTA, 1987). BIS Applied Systems produce a "Computer Disaster Casebook" which documents over 175 cases of computer disasters (CCTA, 1989). Analysis of these disasters gives a breakdown by cause, as shown in Figure 2.1 below.

Figure 2.1 - Causes of Computer Disasters



As is shown, the most common causes of computer disasters are fire, power and water. An IBM study, covering the period 1967 to 1978, identified 352 major computer disasters. 62% of these, in the USA, were caused by fire and flood. There were several reported cases of computers sited below river flood levels. There were also accounts of damage from fire hoses, even where the seat of the fire was adjacent to, but not in, the computer room (Tigo, 1989). Choice of site is often equivalent to choice of neighbours. The threats of fire and flood are considered the most probable threats by major insurers. Statistics available mainly from the USA underscore the point. The Office of Insurance Support Services of

the Federal Emergency Management Agency reports that claims paid to non-residential flood insurance policy-holders reached $72 million in 1986. Low-interest small business loans made to companies in disaster-struck regions in the same year totalled $232 million for damage from torrential rain. Damage from six hurricanes in that year exceeded $3.5 billion. According to the National Fire Protection Association, the average cost of a fire in a computer room in 1984 was $2.6 million (Tigo, 1989). These disasters can be matched, in percentage terms, by the BIS report above.

2.2.2 Organisations' Responses to Disasters

Computer Disasters can be perceived as low-probability events. Studies of low-probability events reveal that individuals are reluctant to protect themselves against rare events, even though they may produce serious consequences. In dealing with low-probability events, individuals prefer to take statistical chances rather than preventive measures (Tamerin & Resnik, 1971). Sometimes individuals may accept the actuarial reality of the risk posed by low-probability events, but refuse to associate that reality with themselves (Wernstein, 1980). If the probability of some event occurring is below some threshold level, individuals tend to assign zero to the probability of occurrence (Kunreuther, 1976).

The kind of attitude shown towards low-probability events is also reflected toward computer disasters in many organisations. Past studies indicate that organisations are not adequately prepared for computer disasters, because there is a lack of understanding among senior management of the consequences of a disaster for the organisation as a whole. Computer managers have tended to ignore the possibility of disasters. Most believe that they have no need to prepare for disasters, or insist

that disasters and security breaches only happen to others. Some learn the hard way that this is not so, and a few are nonetheless lucky to survive (Smith, 1989). Many managers have the attitude "It will never happen to us," and do not recognise the fact that they are particularly vulnerable to sudden, unexpected loss. Schwieger examined the attitude of computer managers to computer disasters, and has concluded that some managers have always considered disasters as a problem which does not apply to computer organisations (Schwieger, 1986). Marcella found that the reason that computer disaster planning has received low priority in organisations is that top management is not given the information required to make a valid judgement (Marcella, 1985).

The lack of staff education regarding potential problems, and the lack of reports concerning security requirements to senior management, have blocked progress toward considering Disaster Prevention measures. Boyer (1982) cited three main reasons for management's' failure to develop effective computer disaster plans, as follows.

a) The probability of a major computer disaster occurring is so low that a computer disaster plan is not considered to be economically feasible.

b) The development and maintenance costs of a computer disaster plan is high and requires justified solutions.

c) Qualified personnel to develop a computer disaster plan may not be available, and even if they are the time and effort required for development can be substantial.

These studies indicate a historical lack of sufficient attention to the subject of computer disaster planning. At the same time, the surveys quoted show indisputably that the risks which threaten IT installations, and the potential

consequences of failing to take protective measures against them, are of great significance.

2.3 Previous Innovations in the Development of Computer Disaster Prevention and Risk Management Practices

This section is structured to describe the major development milestones, and the conceptual and practical advances which they have introduced. Where appropriate, mention is made of "interim" progress which has occurred in the interval between the main advances.

Since computer disaster prevention is a form of risk management, it has been necessary to investigate available Risk Management methodologies, to show how these methodologies can be compared against the requirements and objectives of the present research. What they fail to achieve can then be identified. As a result, a Computer Disaster Prevention Risk Management Framework is developed in chapter 3.

The actual scope of the IT Risk Management process has been dictated in practice by the perceived requirements of a particular situation. These requirements may demand that the analysis focuses on the application's data or software; and/or on the installation's ability to continue uninterrupted operations. Assets to be investigated for relevant risk exposures can therefore be identified. For example, if the analysis focuses on data or software, their relevant exposures might be to disclosure, modification or destruction (FIPS PUB 65, 1979; DOA, 1977; SDC, 1979; CCTA, 1991). On the other hand, if the analysis focuses on the installation requirements, the relevant exposures might be to assets such as people, equipment,

hardware (including software) or even on the premises which accommodate these assets (FIPS PUB 31, 1974; BS 6266, 1982; BS 7083, 1989).

Before conducting a risk management programme, the intended scope and purpose of the analysis and assessment of risks needs to be ascertained. The project plan in any disaster prevention policy should consider:-

a) the types of assets which need to be protected;

b) the types of threats that may result in a disaster;

c) the types of counter-measures available, which can act as safeguards against those threats; and

d) the types of risk exposure which may result from threats occurring.

For the purpose of this study, computer disasters are those which do not arise from logical causes and their effects on software or data, in which a given loss or exposure may affect:-

e) their confidentiality or integrity if they were modified or disclosed; and

f) their non-availability as a result of destruction

due to any form of accidental error or intentional act. Therefore, security measures to handle logical threats are entirely excluded from this study.

As has been stated, this study has concentrated principally on computer disasters which could arise outside the immediate accommodation of the IT installation, and could result in physical damage and consequential losses. As a result, this Review focuses mainly upon previous work which has been done in that or related areas.

The important related terms which have come to be used within the IT risk management community are explained in chapter 3.

2.3.1 The Evolution of Risk Management for IT Installations (The first milestone)

Work in the area of Computer Disaster Prevention began in the early 1970's in the USA. The first milestone, represented by FIPS PUB 31 of 1974, was enforced by the Public Law B9-306 (Brooks Bill), Part 6 of which is entitled Code of Federal Regulations. This statute gave the Secretary of Commerce important responsibilities for improving the utilisation and management of IT installations in Federal government. To carry out the Secretary's responsibilities, the NBS (National Bureau of Standards, later NIST, the National Institute of Standards and Technology) provides leadership, technical guidance and co-ordination of government efforts in the development of guidelines and standards, through its Institute for Computer Sciences and Technology.

The subject of computer security is of great national interest in the USA. Its implementation involves the use of a balanced set of managerial and technological safeguards. Within the context of a total security programme, the NBS produced the first Guidelines for ADP Physical Security and Risk Management, which was made available for use by Federal agencies.

Although these Guidelines provided Federal agencies with a handbook for use when implementing structured physical security and risk management programmes in their IT installations, it was intended to be a basic reference document and check-list, for general use. The specific guidance given was principally concerned

with security, although some sources of advice for dealing with disaster prevention issues were included.

The relationship between computer disaster prevention policies, on the one hand, and computer security, on the other, needs explanation. Consideration of computer security has traditionally tended to concentrate mostly on loss or corruption of software and data. Disaster prevention controls are concerned to give protection to all exposed assets against loss or damage, including e.g. human health, safety and the working environment. Disaster prevention and computer security are, therefore, complementary to each other.

To illustrate the above distinction, the types of security risk exposure covered in FIPS PUB 31 were mostly associated with intentional acts and threats, such as theft, arson, vandalism, tampering etc., which could result in damage to physical assets or theft of information. As a result, controls or counter-measures could include, for example

a) physical barriers, such as fences, partitions, locked doors and guards; and

b) electronic devices, such as closed-circuit television, intrusion detectors etc.

Although accidental threats and natural disasters were also considered in the guidelines, major emphasis was given to the fire risk. Other risks, such as *flood, earthquake* and *windstorms*, received less attention. It has been concluded that these guidelines were based upon data and information supplied from many sources within the US government and private sectors, and reflected practise and technologies available at that time (1974). The authors stated that as new

knowledge, techniques and equipment became available in the future, the guidelines would need appropriate modification.

As experience was gained through the use and application of the guidelines, a basis for establishing security standards might be developed (FIPS PUB 31, 1974).

2.3.1.1 Summary of the First Milestone

To summarise, then, the first published spur to the disciplined consideration of IT Risk Management was the American Public Law B9-306. This led to the publication of FIPS PUB 31, which was essentially a catalogue of the individual concepts and measures which were available in the early 1970's. It covered security analysis, natural disasters, supporting utilities, system reliability, procedural measures and controls, off-site facilities, contingency plans, security awareness and security audit. It contained statistics and information relevant to the physical security of computer data and facilities, and referred to many applicable publications for the more exhaustive treatment of specific subjects.

2.3.2 Technology Assessment: Methods for Measuring Levels of Computer Security (The second milestone)

The report "Technology Assessment: Methods for Measuring Levels of Computer Security" (NBS, 1985) was the next major publication to review approaches and methodologies available. It considered the following, all of which had been developed after the initiative set in FIPS PUB 31 in 1974. (Comments below on the individual approaches etc. are those of NBS, unless the context indicates otherwise.)

2.3.2.1 FIPS PUB 65

FIPS PUB 65 (1979) was a follow-on to FIPS PUB 31. It was more oriented to threats internal to a computer system (e.g. to data files and applications) than FIPS PUB 31, which was more installation oriented. The methodology called for three major activities:

preliminary examination;

risk analysis; and

selection of counter-measures.

The risk assessment phase, in order to evaluate the loss exposure, required two estimations:

an annualised frequency of threat occurrence; and

the loss resulting from the impact on an asset when a threat materialised.

Multiplying these factors gave an expected annual loss from each threat. Summing the threat-asset pairs gave an Annual Loss Expectancy. It permitted the use of order-of-magnitude estimates, for example the estimated cost impact (i) is rounded to factors of ten as follows.

$10, let i=1

$100, let i=2

$1,000, let i=3

$10,000, let i=4

$100,000, let i=5

$1,000,000, let i=6

$10,000,000, let i=7

$100,000,000, let i=8

"The time needed for the analysis will be considerably reduced, and its usefulness will not be decreased, if both the impact and frequency estimates are rounded to the factors of ten shown [above]. There will be no significant difference in the overall exposure whether the damage from a certain event is estimated at $110,000 or $145,000."

FIPS PUB 65's major strength was that it produced "costed" output which could be easily understood by managers. In other words, they were given a dollar estimate of each loss exposure.

Its major weaknesses were that it required a highly skilled group of people working together, and lacked detailed guidance on how to perform the risk analysis stage. It has also been criticised for not encompassing the whole Risk Management process (Katzke, 1988). For example, the method does not

a) have a comprehensive approach for dealing with vulnerability analysis, safeguard selection or cost-benefit analysis;

b) give any guidance on how to handle the interactions of risk entities, e.g. the location of a computer room above an expected flood level is of little value if the staff access or power source are below this level.

The weaknesses described mean that FIPS PUB 65, although reinforcing some basic Risk Management principles, does not significantly advance the course of the present research. Considered in the context of later developments, it could be further criticised for postulating a discipline without really helping the IT Risk Manager to implement it. For example, the CRAMM approach (see below) includes a software package to assist in implementing a Risk Management methodology.

2.3.2.2 The US Air Force Risk Analysis Management Program (AFRAMP)

AFRAMP was essentially a more detailed (three volumes) implementation of FIPS PUB 65, and included procedures for certifying software and systems for the processing of sensitive data. AFRAMP was "shelved" by the US Air Force in 1980, and replaced by a set of questionnaires. The strengths of AFRAMP were summarised as a highly structured, methodical approach; extensive guidance on asset and threat evaluation; a carefully conceived mechanism for counter-measure selection; and clear assignment of responsibilities. Its potential weaknesses were in the very high level of effort required for implementation; and in the failure to allow an asset to be evaluated in more than one mode of impact (i.e. inability to distinguish between destruction, disclosure, denial of service, etc.). For the purposes of this study, it offers no advance because of the lack of thorough risk exposure identification (see chapter 3).

2.3.2.3 US Department of Agriculture Security Handbook

The US Department of Agriculture produced a "Security Handbook" (DOA, 1977) for use in assessing current security positions; to raise security awareness; and as a management tool for cost-effective resource allocation. Its methodology was similar to FIPS PUB 65, except that order-of-magnitude estimates were not permitted. Only two classes of risk - major and minor - were used. The principal categorising entity was threat rather than asset. Users were required to identify critical system assets and services; the sensitivity of data files; specify what additional counter-measures were required; and estimate threat impacts. The main

strength of the security handbook was in the degree of user involvement. Its major weaknesses were in the lack of guidance on threat identification and assessment; and in the need for a team for implementation. By 1985 the security handbook had been "shelved", and enquirers were being referred to FIPS PUB 65.

2.3.2.4 SDC US Navy Risk Assessment Methodology

The "SDC US Navy Risk Assessment Methodology" (SDC, 1979) was developed in 1979 and consisted of six phases. In the first three, threats, vulnerabilities and assets were identified and evaluated. Assets were evaluated in four modes of impact: unauthorised destruction, disclosure, modification and denial of service. Vulnerabilities were rated qualitatively - e.g. very low, medium. Assets could be evaluated qualitatively or quantitatively. Threats were matched against vulnerabilities using forms provided. "Attack frequencies" were computed using mathematical tables. These attack frequencies were multiplied by asset values to calculate Annual Loss Expectancies. The final stage was to select counter-measures. Precision ratings were called for when threat frequencies and asset values were estimated. A key distinguishing feature of the SDC US Navy Risk Assessment Methodology was the explicit treatment of vulnerabilities, which made the approach more situation-specific than other methodologies. Eight forms had to be used to perform the analysis. Thus, although this methodology was an interesting step beyond earlier methods in recognising some concerns about confidence and the precision of ratings, it suffered from relative difficulty of use by inexperienced users, and from the uncertainty involved in calculating costing from qualitative input. This methodology was an advance in its time, and it confirmed the need for thoroughness in vulnerability analysis which this research has also identified.

2.3.2.5 Risk Analysis and Management Program

The "Risk Analysis and Management Program" (IST/RAMP, 1979) from International Security Technology Inc. was developed in 1976. A front-end enhancement was later developed for it, to ease data entry. It was an automated method for obtaining quantitative estimates of expected losses. The software computed the expected annual loss for three asset types: applications, master files and "room plus contents". Threats were categorised by types of resulting loss, i.e. denial of service, fraud, information disclosure, physical damage and theft. The software allowed summation of all losses from a given threat, or associated with a given application. Iterations of the program allowed the selection of counter-measures to be optimised. IST/RAMP was completely numeric: it presented the expected losses in descending order. Even if some data were missing, it produced numerical answers. Therein lay a potential problem: even if there was little data to use, it would produce useful-looking but meaningless results (NBS, 1985). It was intended for use by a security analyst. In brief, the Risk Analysis and Management Program was an early advance in "automating away" the sheer volume of manual calculation which other methods involved. It still, however, needed a specialist security analyst. A mainframe computer is needed to operate IST/RAMP, although a PC-based data entry module is available.

The present study has extended the automation theme, by using KBS technology to produce a methodology which can be operated by IT personnel. The solution developed is capable of being run on a PC.

2.3.2.6 Relative Impact Measure of Vulnerability

Development of the "Relative Impact Measure of Vulnerability" (RIM) technique began in the late 1970's at SRI International, and was "still in its embryonic stages" (Nielsen and Ruder, 1980). It measured the relative impact on an organisation of vulnerabilities in its computer system integrity. It presented relative measures between two competing systems or configurations. The developers' belief was that using relative rather than absolute measures was advantageous because "comparative data" was both easier to collect and understand, and otherwise the calculation of conclusions followed familiar principles. A disadvantage was that there were no accurate costing figures to present to management. Thus, RIM only enabled the identification of the better of two alternatives. It provided no support to those responsible for selecting Risk Management measures specific to an individual installation. It therefore makes no contribution to the objectives of this research.

2.3.2.7 Fuzzy Risk Analysis

The "Fuzzy Risk Analysis" method was developed by Hoffman and Neitzel (1980). It was "still in the research stage" in 1985. A FORTRAN prototype was originally developed, and more recently a prototype interactive version was "under development". The rationale behind the system was that, since many risk evaluations were based on very little data, and were subject to large error, it was better to use "fuzzy" linguistic terms than numbers for representation. Confidence indicators were also related to each estimate input, to "weight" the estimate. From this input, a risk indicator was computed using mathematical methods grounded in fuzzy set theory. In contrast with the methods described above, fuzzy risk analysis

permits hierarchical levels of detail, so that estimates etc. can be made at a selected level. This method was still a research tool, and not ready for production use. Worse, it was not currently being worked on. Its potential perceived strength was that numerical estimates were not needed, thus allowing a "quick feel for where ... major risks are ...". Its weaknesses were that numerical estimates were not handled, although this was intended as a later development; that no checklist was provided; and that it only considered risks, not costs. For these reasons, this approach made no contribution to this research, where a main objective was to produce a rigorous and detailed methodology

2.3.2.8 Security Assessment Questionnaire

The "Security Assessment Questionnaire" was developed by IBM in 1980, and revised in 1985. It contained fourteen categories, which were divided into three key security areas: physical security, controls and procedures, and contingency planning. The categories included such aspects as fire, operational controls, and back-up. The end of each category gave space for rating risk for the entire category as extremely low / necessary / acceptable / high. There were also references to other helpful publications. The advantages of the questionnaire were that it was brief, and allowed the user a quick assessment of an installation's security status. The "big problem" was that there was no guidance on how to arrive at the risk rating for each category, so that the results were very dependent upon the skills of the user. Thus, for the purpose of this research, the questionnaire does not provide any material advance.

2.3.2.9 Summary of the Second Milestone

The main conclusions of the report "Technology Assessment: Methods for Measuring Levels of Computer Security" (NBS, 1985) were as follows.

a) The underlying structure of the methodologies was fairly constant.

b) All of the methodologies neglected relevant functions, issues and inter-relationships. As a result, they contributed to a misunderstanding of the evaluation process.

c) Little empirical data existed on the use, success or failure of the methodologies.

d) The different methodologies were not competing with each other.

e) Rating is a useful aid in simplification, but is susceptible to misinterpretation when numeric representations are used for the ratings.

f) That there was no widely accepted existing way of measuring levels of computer security.

As an intermediate summary, it is clear that between the early 1970's and 1985 much work had been directed to developing IT Risk Management. Although this common objective existed, (FIPS PUB 65) and (NBS, 1985) pointed to a clear lack of standards in terminology and methodology; and to the impossibility of comparing the merits of the approaches which had been developed. If this situation were to persist "we will continue to have methods that are as different as apples, oranges and pears, and which will produce results that cannot be compared in any meaningful way" (Katzke, 1985).

2.3.3 Computer Security Risk Management Model Builders Workshop (The third milestone)

A Computer Security Risk Management Model Builders' Workshop was held in Denver, Colorado in 1988. It was intended to stimulate the further work in Computer Risk Management which had been shown necessary (as above). It was also the platform for the next major review of the "state-of-the-art", in the form of individual presentations. The Workshop represents the third and final "milestone" referred to above, and is also the first point at which references to the use of KBS technology are significant.

The Workshop had two goals: to

> "converge on a single set of terms and a description of their relationships that adequately describe the process of measuring the security risk to computer-based systems and assuring adequate levels of security"; and
>
> "bring together key thinkers in the field of risk management to build a common purpose and understanding among them for the free interchange of ideas ...".

Working group sessions were held, where the aim was to

> "arrive at a general risk management model that is comprehensive and understandable, identifies all components of the risk management process, and defines the functional relationships between the components".

The most significant of the individual presentations are summarised below.

2.3.3.1 Government Perspective on Risk Management of Automated Information Systems

Katzke (1988) traced the U.S. Government's interest and activities in DP risk management, he

a) discussed the terms and concepts needed to understand risk management;

b) described FIPS PUB 65;

c) outlined advances and activities which had resulted in a better understanding of the risk management process; and

d) presented a plan for improving the Government's ability to perform risk management.

FIPS PUB 65 has been described above, where Katzke's comments on it were reported.

In later years, the National Bureau of Standards looked for a candidate standard for IT risk management. It concluded that it was too early to standardise, since no candidate(s) could serve because of

a) low user satisfaction;

b) incomplete development;

c) primitive user interfaces; or

d) incomplete or inconsistent detail.

More recently, the National Bureau of Standards had increased its activities in the risk management area because of:-

a) emerging methods which were better able to handle the risk management process, including safeguard selection and cost-benefit analysis;

b) the use of qualitative approaches;

c) reductions in the resources needed to carry out risk management, because of the increased use of computers for data collection and analysis;

d) orientation to PCs; and

e) better applicability to both computer applications and IT centres.

Most recently, the National Bureau of Standards and the National Computer Security Centre had

f) jointly established a risk management laboratory;

g) worked on the comparison of risk management methods;

h) set up a compendium of available risk management methods; and

i) developed "standard" test cases and scenarios to compare alternative risk management methods.

This paper is of interest as a chronicle of past work, and describes in overall terms the future intentions of the National Bureau of Standards.

2.3.3.2 A Matrix/Bayesian Approach to Risk Management of Information Systems

In "A Matrix/Bayesian Approach to Risk Management of Information Systems" (Mosleh; 1988) the author noted that the information systems security community was becoming increasingly aware of the power and effectiveness of risk assessment in making decisions regarding security improvement. This awareness and interest, however, had been accompanied by a sense of confusion as to what exactly to expect from risk assessment, and how to perform it. The main source of this confusion had been the wide range of interpretations of risk assessment in general, and the existence of a large number of methods and tools to conduct it. Despite this fact, very few attempts had been made by the experts in the field to study the issues involved, and to try resolve the differences in interpretations and methods in a systematic and scientific way. The purpose of the paper was to offer some ideas about the meaning of risk, and to present a mathematical framework for assessing it.

He defined risk management as an activity involving two major steps, viz.:-

1. Risk Assessment, and

2. Cost Effective Risk Reduction.

Risk Assessment in its most comprehensive form was an activity which produced answers to the questions:-

What can go wrong?

What would be its consequences?

How likely is its occurrence?

How certain are we about the answers to the first three questions?

He went on to discuss the representation of risk; the likely elements of a risk model for information systems, including the identification of threats, impact analysis, and the evaluation of consequences; the development of a logic model; the development of a matrix to represent elements in risk models; the quantification of risk models, including point estimating, and the quantification of risk with uncertainty; and cost-effective risk reduction.

The author's achievements in showing how to apply Bayesian methods to risk assessment and management are not doubted. He recognised vulnerability as an important element in modelling risk, and treated vulnerability simply as a result of the absence of safeguards.

A principle objective of the current project, however, is to produce a methodology for thorough IT risk management. To achieve that thoroughness, it needs to recognise that levels of vulnerability can only accurately be determined by considering the interactions and inter-dependencies among threats, assets and counter-measures. The methodology also needs to be capable of use by practising IT managers. They are unlikely to have the time or expertise to follow the detailed analytical methods adopted in Mosleh's paper.

2.3.3.3 Using Binary Schemas to Model Risk Analysis

Lewis (1988) proposed the use of semantic binary schemas for modelling risk analysis concepts, and presented a preliminary binary schema model of a selected set of concepts. These concepts were described by a set of terms, whose definitions and interrelationships were clarified through the use of a schema. It

was argued that binary schemas formed a useful tool for modelling risk analysis concepts, and could provide a framework for the comparison of different risk analysis models.

The author pointed to inconsistent use of terminology in computer security risk analysis, which made it difficult to compare different tools. The use of binary schemas clarified issues, and the intention was to aid understanding and comparison.

Information (in this case on assets, threats etc.) in a binary schema was represented by the classification of objects into categories, and by logical associations between categories. This approach was more flexible than a classical database model, and experience reported in the paper had shown the author that this form of modelling was useful in the context of risk analysis.

The model advocated did not address the threat environment, which the author considered sufficiently complicated for representation by a separate model. The author also acknowledged that a successful risk analysis model needed to be able to describe functional dependencies, but said " ... it is not yet clear if the binary schema approach can be manipulated to do this in a reasonable fashion." In the present study, it has been found that the thorough consideration of the threat environment, and of functional dependencies, are essential in analysing risk exposures.

2.3.3.4 CCTA Risk Analysis and Management Methodology

The "CCTA Risk Analysis and Management Methodology" (CRAMM), (Moses and Glover, 1988) is a methodology for IT security developed for the UK

government by its Central Computing and Telecommunications Agency (CCTA). The requirements set for a government risk analysis and management methodology, in 1985, were that it must be:-

i) able to deal with government operational and administrative systems of all sizes;

ii) able to encompass all aspects of IT security;

iii) compatible with existing government IT security guidance;

iv) suitable for use during system development, as well as for existing installations;

v) easy to use by staff without IT security experience;

vi) able to allow reviews to be carried out quickly;

vii) able to be used with an "automated support tool"; and

viii) able to produce results which are understandable to non-technical management.

No existing methodology was found to meet those requirements in full, so CRAMM and its software support tool were developed, and were launched in 1988. The underlying model accommodates the following components (or risk entities):-

a) assets;

b) a "review boundary", within the limits of which is everything which could impinge on the security of the system;

c) threats;

d) asset inter-dependencies; and

e) impacts and secondary impacts, which arise from a combination of a threat occurrence and an asset vulnerability.

Data on these components are collected from completed questionnaires. From that data CRAMM determines the levels of risk which have to be managed, by valuing the assets and measuring the order of threats and vulnerabilities. (At this stage, it is assumed there are no existing counter-measures.) The dependency of the system on groups of assets is also evaluated. The questionnaires are "scored", to indicate high, medium or low threats and vulnerabilities.

The combination of asset values and threats/vulnerabilities is used to calculate a security requirement number, on a scale of one to five, for each of four possible impacts (i.e. disclosure, modification, unavailability and destruction.

Counter-measures are grouped into seven types, namely

Avoidance	A
Transfer	T
Reduction of Threat	RT
Reduction of Vulnerability	RV
Reduction of Impact	RI
Detection	D
Recovery	R

It is noted that the more effective counter-measures, A, T and RT, are easier to consider during the design stage of an installation.

CRAMM then considers how identified security needs can be met, by selecting counter-measures from a library, which is referenced by security aspect and type (i.e. A, T, RT, etc.). If the review is of a current installation, its existing counter-measures are now recorded. A list of recommended counter-measures is next produced, with likely costs taken from the library. The prioritisation of counter-measures, a "what-if" and a back-tracking facility are included.

It is clear that CRAMM is a thorough and structured methodology, which meets the full set of requirements set by the UK government. It is well able to support the whole risk management process, which needs to include the identification, analysis and management of risks. The library of counter-measures has been assembled using advice from IT security specialists; and previous concerns about evaluating vulnerabilities accurately have been overcome.

Interestingly, during a demonstration and discussion of CRAMM at CCTA, two officials indicated that for a next version, they would elect to use KBS technology.

2.4 Knowledge - Based Systems (KBS) Technology

The terms artificial intelligence (AI), Knowledge Based Systems (KBS) and Expert Systems (ES) are all in current use. The expert system concept evolved from research in artificial intelligence which is a branch of computer science concerned with the automation of intelligent behaviour. Expert systems are interactive programs that encapsulate the knowledge and skills of human experts gained from many years of experience in a narrow field. Expert systems are in growing use

throughout industry and commerce. They can preserve knowledge, make scarce expertise widely available, offer more consistent decision making, improve staff and equipment utilisation provide spin-off training and organisation image. Despite the increase in the use of computers, all recent experience with expert systems shows that the benefits of this technology can be maximised if organisations take some care in its introduction.

Early systems were usually called expert systems. Recently most knowledge engineers refer to their systems as knowledge based systems . This technology is still a new field. There is the tendency for leading authors to use different terms for the same "construct". One may have to track down terms and definitions by consulting many sources. Another indicator that reflects the technology's youth is that many experts disagree about various aspects. Although the recent literature related to the subject might seem quite voluminous, technical issues are rarely or only addressed with superficial examples (toy Problems). Expert systems have been extensively discussed (Michie, 1979; Michie, 1982; Hayes-Roth et al, 1983; Weiss and Kulikowski, 1983; Alty and Combs, 1984; Forsith, 1984; Harmon and King, 1985; Johnson and Keravnou, 1985; Sell, 1985; Simons, 1985; Jackson, 1986; Frost, 1986; Walker and Miller, 1986; Waterman, 1986; Hu, 1987; Keller, 1987; Kriz 1987; Harmon et al, 1988; Bielawski and Lewand, 1988; Luger and Stubblefield, 1989; Jackson, 1990; Turban, 1990) and need not be expanded here.

2.4.1 Early Systems

In the mid and late 1970's, much of the work in expert systems was centred around the development of medical consultation systems. Expert systems have also been developed in geology, computer configuration and engineering. They have been

applied with great success in diagnosis as well as monitoring planning, fault finding and design. Some of the recent UK practical KBS developments are outlined in chapter 5.

2.4.2 Material Covering the use of KBS Technology in IT Risk Management

Referring back to the third milestone some researchers have begun to adapt a KBS approach for the development and implementation of risk management practices. Some of the more significant presentations which was described in the Computer Security Risk Management Model Builders Workshop are described below. (The concept of KBS technology and the more detailed considerations for a system which facilitates the implementation of computer disaster prevention policies which is the aim of this research are reported in chapter 5).

2.4.2.1 Definition and Identification of Assets as the Basis for Risk Management

In Definition and Identification of "Assets as the Basis for Risk Management" (Mayerfield, 1988) the author also pointed to the lack of an accepted procedure for evaluating and minimising risk, compounded by confusion over the underlying concepts and issues. He reviewed some existing risk management techniques and methodologies, and compared them. He went on to present a "new, comprehensive conceptual model of the risk management process". The model had three constituents: informational assets, system components and threat agents.

The paper also described a knowledge-based system based on the model, which "... will also provide the basis for ... enhancing [his] methodology ... ".

Familiar risk estimation techniques (e.g. Annualised Loss Exposure) were discussed. It was commented that many of the differences between the various techniques advocated reflected the respective perspectives of the analysts: those who concentrated on the assets to be protected, looked first at asset vulnerabilities; while the viewpoint of the "threat analyst" was more concerned with the value of the assets to threat agents, than to the assets' owners. It was concluded that a combination of these perspectives would yield optimal results.

The knowledge-based system, which emerged as still under development, used a "shell". Its knowledge base contained representations of detailed class/object structures describing assets, system components and threats. Rules were used for reasoning about relationships between these entities, for inferring risks, and to control the operation of the system. Plans for later development, including interfaces to external databases; and for the use of statistical and fuzzy reasoning techniques, were outlined.

This paper adds no new knowledge of the project domain, and the knowledge-based system references are too superficial to allow detailed study, perhaps for commercial reasons. To the limited extent that it mirrors work involved in this project, it may be taken to confirm the validity of this research. The author's conclusion that the combination of the perspectives, previously adopted in a polarised way for risk assessment, would give better results, was recognised in the methodology developed as part of this research.

2.4.2.2 LAVA: An Expert System for Risk Analysis

In using an expert system approach for Risk Analysis Smith (1988) describes an approach called LAVA, which is an acronym for "Los Alamos Vulnerability and Risk Assessment Methodology". The Los Alamos National Laboratory had developed a methodology for risk analysis which could be tailored to address a variety of subject systems' requirements. LAVA is based on hierarchical systems theory, event trees, fuzzy sets, natural-language processing, decision theory and utility theory. It includes a PC-based system, where the user sees an interactive questionnaire which elicits, for example, data on safeguards, potential consequences of a threat occurrence. Tailored applications "have been used to model risk management for computer safeguards systems, physical protection systems, transborder data-flow security systems, contract awards systems, nuclear safeguards systems, embedded computer systems, survivability systems, weapons systems security, and property management systems". LAVA was reported to be especially effective in modelling systems that include a dependence on human actions.

Users were not required to be expert risk analysts: the mathematical and analytical expertise was part of the methodology's general software system. Expert knowledge about safeguards and security systems was part of each specific application model. In modelling assets for a LAVA application, the developer created asset categories, which treated similar assets as one. Desirable sets of safeguards were modelled as an automated interactive questionnaire, which elicited information on the presence and quality of existing safeguards. A database was used to derive vulnerability values, which were reported at various levels to guide those responsible for reducing vulnerability levels.

LAVA was thus presented as a two-tier system, where the lower tier or "kernel" handled the mathematics of basic risk evaluation, while an upper tier had to be written to accommodate the analysis of data specific to individual scenarios.

KBS technology was used in LAVA's development, and the underlying approach has been, to an extent, followed in developing the Prototype Knowledge Based Decision Support System which forms part of this study. In the latter case, the use of a two-tier approach has not been found necessary.

The LAVA methodology stresses a team approach for conducting risk assessment. The team should be composed of people with a broad spectrum of backgrounds and expertise to ensure a thorough assessment. It is recommended that a consensus among the group be reached before entering an answer to any of the questions and, in some cases, this may be the most difficult part of administering the software (Gilbert, 1992).

In the methodology developed in the present research, an essential element is to use KBS to encapsulate relevant expertise, so that the methodology can be used by an IT manager (as with CRAMM), without the need for a committee approach.

In the Prototype System, all input data needed for the risk identification phase can be collected or supplied direct by the IT manager. The specialist knowledge about interactions and inter-dependencies between risk entities is embodied in the knowledge base, to ensure that the IT manager has no need to seek out this information.

2.4.2.3 An Expert Systems Approach to the Modelling of Risks in Dynamic Environments

Bonyun and Jones (1988) described an approach, again using expert system (or KBS) technology, inspired by the Canadian Department of National Defence and Communications Security Establishment. Earlier work had produced a software package known as IPSATA (I. P. Sharp Associates Threat Analysis), which was described as a theoretical model of threat analysis. This presentation outlined how KBS technology had been applied to that model, and in two developments, namely:-

MAPLESS, or Mixed paradigm APL Expert System Shell; and

a "supershell" called KEEPER: Knowledge Engineering applied to the Evaluation of Potential Environmental Risks.

A mathematical model of risks, the model incorporated in KEEPER, was also presented, as were the reasons for the original selection of a KBS approach and for the conclusion that a KBS should entirely replace the older system.

MAPLESS was described as a software tool for constructing APL-based expert systems; and a knowledge base as a data structure representing knowledge about a field of expertise, consisting of the following parts:-

a set of frames;

a set of association units; and

an uncertainty calculus.

Frames represented entities and their attributes. An association unit represented an association: a collection of similar relationships between conceptual entities. An

uncertainty calculus consisted of a set of uncertainty measures, and operations on those measures. MAPLESS provided the following functions:-

> an interactive editor for constructing, querying and maintaining knowledge bases; and
>
> an environment and a set of utilities for constructing, testing and maintaining programs (inference methods and knowledge base applications).

It had been found that the use of IPSATA involved "a fair bit of human activity", which was acceptable in a strategic tool which would have only infrequent use. "If, on the other hand, the environment ... is ... dynamic ... massive human effort can no longer be permitted. The way to remove or reduce human interaction with a system is to give ... access to an expert system ... ." These conclusions led to the development of KEEPER.

KEEPER was described as a "supershell", with the following explanation:-

> "Just as an expert system shell can be used to make many different expert systems, each based on different collections of knowledge, so, likewise, can the supershell be used to create different end systems (really different versions of the same system).
>
> "Since the systems differ only in terms of the specific data (or knowledge) they contain, and not in their basic structure ... , it appeared ... appropriate to think of the commonality as a shell that had already begun the process of becoming instantiated as an expert system."

The functional aspects of KEEPER were described under the following headings.

The Taxonomies of the object entities, i.e. assets; threat agents ("entities that might be the perpetrators of misadventure on the assets"); groups of risks; and possible safeguards.

The Associations necessary to maintain the taxonomies, i.e.

hasSubclass	hasMember	hasPrototype
inheritsAll	hasImpactArea	isPerpetratedBy

Generic Attributes, of assets, threat agents, groups of risks, and possible safeguards. Wherever assets possessed attributes which are significant to the expert system, they must be present, with values. The exact attributes which needed to be collected could not fully be determined until the required evaluative procedures had been defined. Examples of attributes were the annual frequency, average severity and costing formula for risk entities.

Taxonomies of Rules, about

Risks, to indicate the effects of a threat occurrence;

Choosing Safeguards, to determine which Safeguards are appropriate; and

Safeguards themselves, to indicate the effects of introducing new Safeguards.

Activities, which were either

automatically performed as demons, to keep the knowledge base well-organised and internally consistent; or

user-invoked, such as recording changes to entities and attributes.

The authors' main conclusions included the following.

a) Basing a risk assessment system on an expert system structure was desirable for both dynamic, and more static or strategic, situations.

b) The expert system approach was desirable because, once constructed, the risk assessment system could be operated and maintained by people with less specialised knowledge.

MAPLESS and KEEPER were still undergoing development, and had not yet reached the working prototype stage.

It is clear, however, that the reasons for selecting an expert systems approach included some of those which were identified during this study.

Because, in principle, the same approach was used, then some methodological similarities are inevitable. Again, as with LAVA above, it has not been found necessary in this study to adopt the kind of two-tier structure for the system which MAPLESS and KEEPER represent.

To develop the Prototype System which is combined with this thesis, a KBS development toolkit (Logic Programming Associates' ***flex***^{tm}) was used. Unlike shells, which are typically a previous application from which the rules have been "stripped out", this toolkit includes the full underlying programming language (in this case Prolog). It therefore retains the full functionality of that underlying language. The reasons for selecting ***flex***^{tm} for this development are described in chapter 5.

2.5 Summary of the Literature Review

It was not until the early 1970's that a strong, widespread perception developed of the need for organisations to address the security of their IT installations.

This realisation was spurred by

> the sharply increased dependency which government, commerce and industry had on IT facilities, to support day-to-day operations; and
>
> recognition of the profound effects which even the temporary loss of those facilities could have on
>
> > services provided,
> >
> > profit levels, and even
> >
> > business survival.

As a result, several methodologies were developed which were intended to guide the implementation of computer disaster prevention policies.

By the early 1980's, two main further sets of development were under way, namely:-

> a search for some measure of agreement and standardisation in place of the fragmented and often incomplete approaches which had emerged; and
>
> the automation, using computer systems, of some of the approaches and conceptual methodologies which had been developed.

By the middle 1980's, these early automated developments were becoming more refined, and some developers had begun to adopt KBS methodology to allow the expertise necessary to implement successful disaster prevention policies to be put in the hands of non-technical managers.

The picture now is one which comprises two main elements, namely

some methodologies which excel in some, but not all, of the required areas, and

a continuing lack of one methodology which handles all risk management aspects well, and of an agreed standard which would form the basis of such a comprehensive methodology.

2.5.1 The Significance of the above Summary for the Current Research

Reviewing the situation described above alongside the requirements of this research, two main sets of points emerge. From the published information studied, previous work has not met the objectives of this research in that:-

a) the complex area of interactions between risk entities has still to be fully addressed; and

b) there is still a lack of a full methodology for computer disaster risk management which can be implemented by an IT manager.

Specific illustrations of how the methodological requirements at a) and b) above were met in this research are given in chapters 3 to 6 below. These chapters report

in detail the results of this research and show the contribution of each major development stage (as outlined in chapter 1).

Chapter 3

THE FRAMEWORK FOR IT RISK MANAGEMENT

3.1 Introduction

Since computer disaster prevention is a form of risk management, it is necessary to consider the established concepts of risk management, and to examine their applicability to computer disaster prevention. In the previous chapter, several risk management methodologies covering a wide spectrum of computer security risks have been reviewed. Because of the differences in scope and purpose, and the lack of a unified approach to the systematic application of computer disaster prevention measures, none of the methodologies in isolation were found suitable to serve the objectives of this research. As was stated in Chapter 1, one main objective of this research has been to develop a structured methodology for managing the risks of computer disasters. In this chapter, a structured framework for computer disaster risk management is presented, to form the foundation for the methodology described in chapter 4. As may be expected, this structured framework has also been used in the overall design of the Prototype Knowledge Based Decision Support System (KBDSS) which forms part of this project's submissions. This KBDSS is reported in chapter 6.

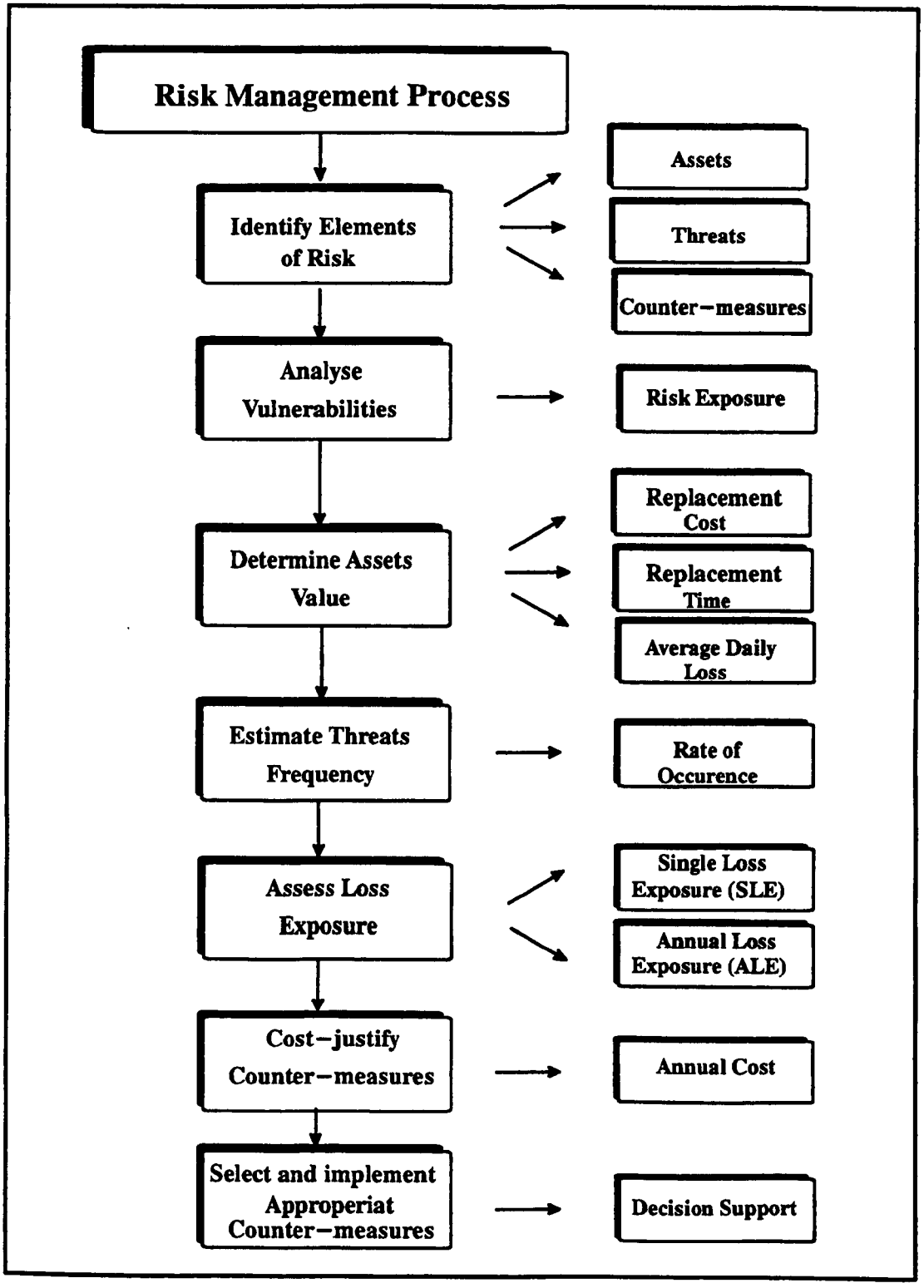
3.2 The Concepts of Risk Management

The suggested computer disaster prevention risk management framework is based on the concepts of risk management. Risk Management (RM) involves the analysis and assessment of risks, and can enable decision makers to take corrective actions regarding security improvement. The process of analysis and assessment is evaluative in nature. It focuses not only on the range of threats to which an IT installation is exposed, and the loss potential due to those threats, but also on the controls currently in place, and their effectiveness. The outcome is information suitable for allowing the selection of appropriate additional controls. This information is based on justifying the cost of required investments in protective measures against the likelihood and costs of breaks in service. RM is therefore the process which addresses the whole spectrum of activities leading to cost-effective counter-measures against harmful events (Troy, 1988; Katzke, 1985; Gilbert, 1989). Figure 3.1 illustrates the process of RM, as it is used in the Framework described immediately below.

3.3 Elements of the Framework - an Outline

The power and effectiveness of risk management as a systematic conceptual aid in dealing with the risk of computer disasters has already been acknowledged in this thesis. As part of the current research's contribution, the incorporation of risk management processes into a framework for the implementation of computer disaster prevention policies was organised into several phases. These phases have been specifically structured to reflect the sequence of actions required in identifying, analysing, assessing and managing the risks of computer disasters. It should be pointed out that the organisation of these phases developed cumulatively, as a result of identifying much other work, among which some

Figure 3.1 - Risk Management Process



general guidance on the IT risk management process is scattered. Thus, collating that data into a single source has been a major contribution, towards the development of the recommended solution. The following suggested phases for a structured framework provide a step-by-step approach, to ensure that the entire risk management process is handled thoroughly, thus allowing the achievement of computer disaster prevention requirements.

The Risk Identification phase (The risk entities data collection phase)

1. Asset Identification and Valuation
2. Threat Identification
3. Counter-measure Identification

The Risk Analysis phase (The risk exposure determination phase)

1. Vulnerability Analysis
   - levels of risk exposure
   - types of risk exposure

The Risk Assessment phase (The loss exposure calculation phase)

1. Single Loss Exposure (SLE)
2. Annual Loss Exposure (ALE)

The Control Management phase (The cost-benefit analysis phase)

1. Ascertain Cost-justified Counter-measures
2. Select and implement the most Cost-effective Counter-measure(s)
3. Implement Review procedures to ensure that the disaster prevention policies are maintained.

3.4 Contributions of the Framework Phases

The incorporation of the above phases into the Framework has been significant in ensuring the required rigorousness of risk management; and the contribution of these phases, in turn, was mainly to ascertain the following.

1. The risk identification phase must first identify and record in detail the associated elements of risk (or risk entities) which are needed for the later risk analysis, assessment and control management phases. Such risk elements are assets, threats and counter-measures.

2. The information gathered about risk entities in phase 1 needs to be analysed, to determine how their interactions would influence:-

   the level of risk exposure (e.g. which assets would be impacted by a disaster, taking account of existing counter-measures); and to distinguish

   the type of impact resulting (e.g. in this case destruction of assets, and/or denial of service).

3. The level of risk exposure needs to be assessed to ascertain what potential loss would result if a computer installation is exposed to such risk, and how this level of risk exposure can be reduced.

4. The result of the assessment can support decisions on the selection and implementation of cost-effective counter-measures which will reduce risks to an acceptable level.

3.5 Explanation of the IT Risk Management Terms Used

Before entering detailed discussion of the above phases, it may be helpful to explain the terms which have been used, so that the reader has a clear understanding of the relevance and use of these terms within the context of the methodology which has been developed. For the purpose of this study, the table below explains the terms which are used in the Framework and Methodology presented in this thesis.

Table 3.1 - Explanation of terms used in IT risk management

Term	Explanation
Risk	The effect(s) of threat occurrence, measured in terms of the costs or losses to an organisation.
Risk Identification	Is the process of identifying the elements of risk (risk entities), such as assets, threats, counter-measures, and recording that information for the Risk Analysis, Risk Assessment and Control Management phases.
Asset Identification	Involves recording information about the location, inter-dependencies etc. of all components of an IT installation which contribute to the service which it provides, such as premises, hardware, software, personnel, the value (including revenue) of the service, counter-measures etc. This information is used at the Risk Analysis phase, which determines levels and types of risk exposures.
Asset Valuation	Involves recording information on each asset's replacement cost, and the value of its contribution to the service provided by the installation. This information is used at the Risk Assessment phase, which determines the potential costs of loss exposures.
Replacement Costs	The costs of replacing or re-constructing destroyed assets. This information is used at the Risk Assessment phase, where the type of risk exposure is shown to be destruction.

Table 3.1 - Continued

Replacement Time	The time required to replace an asset, so that a normal level of service can be restored (expressed in days). This information is used at the Risk Assessment phase, where the type of risk exposure is shown to include denial of service.
Average Daily Loss	The average daily revenue which accrues from the service provided, which would be lost in the event of denial of service. This information is used at the Risk Assessment phase, where the type of risk exposure is shown to include denial of service.
Consequential Loss	The sum of a) losses due to denial of service, e.g. loss of goodwill, revenue, customers, etc. (usually derived from average daily loss); and b) losses caused by destruction of assets, in terms of replacement costs.
Denial of Service	Results from direct damage to IT facilities, e.g. buildings, plant rooms, computer equipment etc.; or indirectly from loss of electrical power, water supply, communications, key staff and access to working areas.
Threat Identification	Is the process of ascertaining all of the existing or potential intentional, accidental or natural hazards to which an IT installation and its assets are or may be exposed. Information is also required on the frequency of each threat occurrence.
Threat Agent	An entity which may initiate a threat occurrence.
Frequency of Threat Occurrence	Is the number of occasions when a loss-causing event is expected to happen during a given period (in this case, annually).
Identification of Counter-measures	Is the process of recording information about safeguards or controls, which provide some degree of protection of assets against threats.
Risk Analysis (Vulnerability Analysis)	Is the process of analysing the interactions of assets, threats and counter-measures, to determine an installation's exposure to threats (e.g. which assets would be impacted by a threat occurrence, and with what likely result - destruction of assets and/or denial of service).
Exposure Classes	The type of impact which a physical threat occurrence may have upon an asset, i.e. destruction and/or denial of service.
Risk Assessment	Is the process of calculating an installation's loss exposure (see below).
Loss Exposure Calculation	Assesses the direct and indirect financial consequences (including costs and denial of revenue) of a threat occurrence.
Single Loss Exposure	The expected financial consequences (as above) of a single threat occurrence.
Annual Loss Exposure	The projected loss (as above) which can be expected in any one year, which is calculated by considering the expected frequency of threat occurrence.

Table 3.1 - Continued

Measurement of Risk	Assessing the effect(s) of threat occurrence, which may involve quantitative and/or qualitative approaches.
Quantitative measurement	Computation in precise numerical terms, where information on threat frequency and risk is obtainable or can be mathematically derived.
Qualitative measurement	Assessment from subjective or descriptive information where precise data is unavailable or difficult to obtain (usually expressed as very likely/probable/ improbable frequency of occurrence; or high/medium/low risk).
Control Management	Is the process of ascertaining which counter-measures are cost-justified, and selecting and implementing the most cost-effective.
Cost Benefit Analysis	Compares the investment costs of counter-measures with the reduction in Annual Loss Exposure which will result from that investment. This is also referred to as Trade-Off Analysis and Value Analysis.
Costs of counter-measures	The level(s) of investment required to provide relevant protection to those assets or groups of assets which have been identified as being at risk.

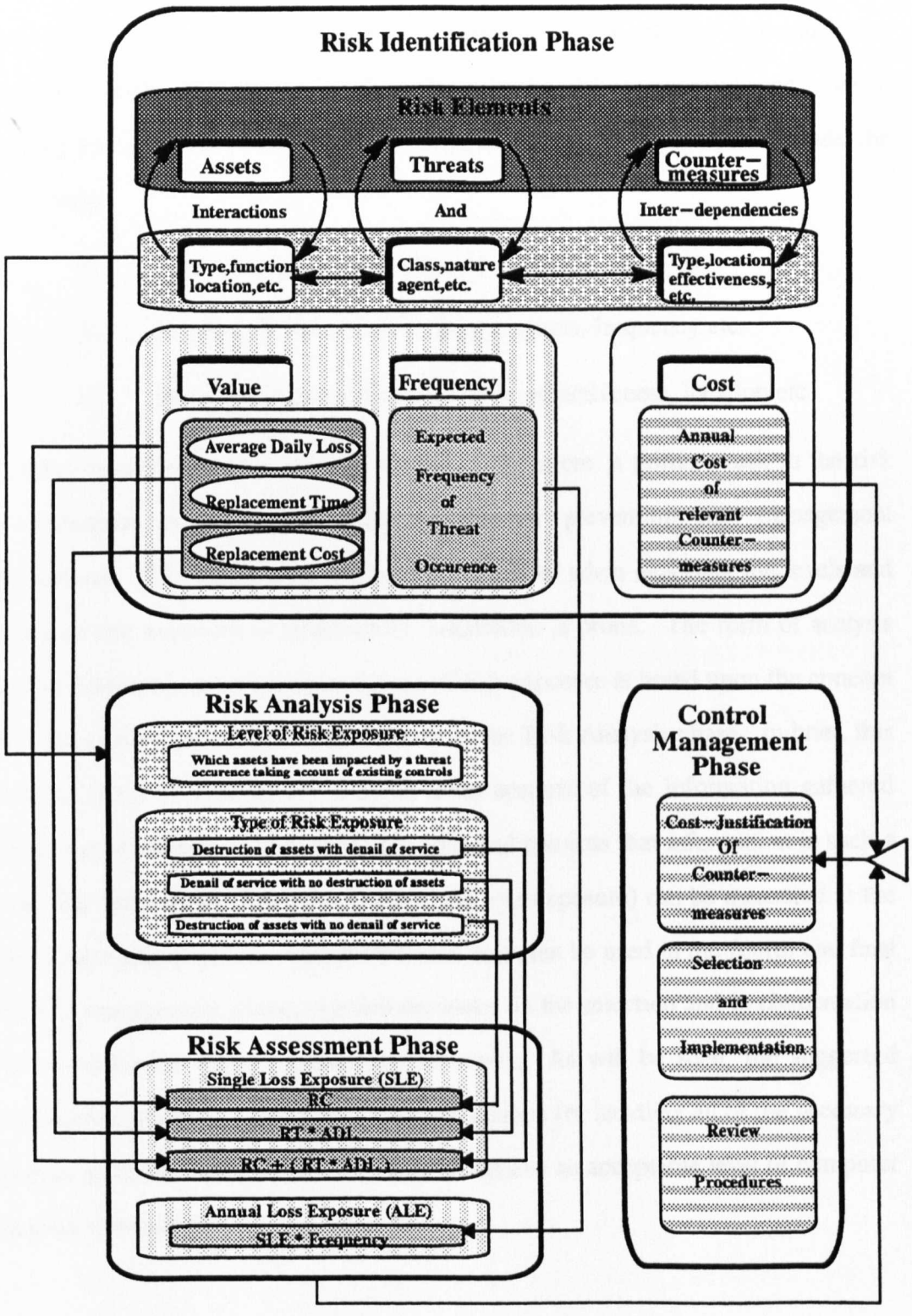
3.6 Discussion of the Framework's Phases

As a reminder, this research has identified four principal phases which need to be worked through in an IT Risk Management exercise, as follows.

1. The Risk Identification phase (risk entities data collection)
2. The Risk Analysis phase (risk exposure determination)
3. The Risk Assessment phase (loss exposure calculation)
4. The Control Management phase (cost-benefit analysis)

Figure 3.2 below illustrates the systematic and logical sequence of all of the phases involved in risk management, and the contents of those phases. The arrows indicate the relationships between the phases and elements, to show how and where information is needed and used.

Figure 3.2 - The Framework - Risk Management Phases



3.6.1 Overview of the Phases in the Framework

Risk identification is the first important phase of risk management. It is the process of gathering and recording information, about all the individual risk entities, which is needed for the risk analysis, risk assessment and control management phases. The information gathered may, for example, include the following.

a) For assets, their types, function, location, value etc.

b) For threats, their classes, nature, agents, frequency etc.

c) For counter-measures, their types, effectiveness, location etc.

Collecting and recording this information is, therefore, a primary task in the risk identification phase of this computer disaster prevention risk management framework. The information gathered is used later when analysing the levels and types of risk exposure to which an IT installation is prone. The form of analysis which determines these levels and types of risk exposure is based upon the concept of vulnerability, which is discussed below in the Risk Analysis phase. In brief, this second phase of vulnerability analysis takes account of the information gathered about assets, threats and counter-measures, and presents that information in such a way that the potential cost of a disaster (the loss exposure) can be measured at the third, assessment, phase. This information can then be used in the fourth and final control management phase, to reach decisions on the selection and implementation of cost-effective disaster prevention measures. As will be seen, the suggested framework provides a structured, logical sequence for handling all of the necessary phases of IT risk management, in order to achieve an acceptable level of computer disaster prevention.

The level of information required for the analysis should first be identified, to determine the scope of individual risk management exercises. Most risk identification methods first require the recording of information about assets and their values. After this initial step, however, each of the earlier methods reviewed proceeds differently. Some focus on asset vulnerabilities (Fordyce, 1982); others on threat identification (Smith, 1986); and still others on controls that can be applied to ensure the protection of assets. The use of a check-list or questionnaire approach to aid data collection has been adopted by many IT risk managers to record the information about risk elements which is required for risk analysis (for example, Browne, 1979). As part of the approach adopted in this research, the questionnaire method is refined to ensure that data about assets, threats and counter-measures is collected in a highly structured manner. This refinement is achieved by using the "three environment" approach, developed as part of this research, and described in chapter 4. This approach allows a structured examination for all of the involved risk entities to be evaluated and considered thoroughly.

3.7 Detailed Discussion of the Phases in the Framework

The following sections provide detailed discussion of all the phases considered in risk management.

3.7.1 Phase 1 - Risk Identification (Risk Entities Data Collection)

The conduct of a proper and effective RM exercise is critically dependent upon an understanding of all of the components of an IT installation and its environment, including the elements of risk and the relationships between them. The risk

identification phase involves the identification of risk entities (e.g. assets, threats and counter-measures); and identifying their inter-relationships. A three environment approach, as described fully in chapter 4, has been developed to ensure that all necessary data and knowledge about risk entities is recorded by their location. (This approach allows an analysis of vulnerability to be established later, with regard to the relationship and risk entity interactions.)

3.7.1.1 Elements of Risk (Risk Entities)

Since the risk identification phase involves the identification of risk entities, it is important first to describe and define these entities in terms of the following:-

a) an asset will be defined in terms of its value to the organisation;

b) a threat event in terms of its nature and effect; and

c) an existing counter-measure in terms of its effectiveness to reduce the impacts of threats upon assets.

These are further explained below.

Assets

Researchers in the IT risk management field tend to define assets as any thing or resource that is of use or value to the organisation (Carroll, 1984; Parker, 1981). The identification and valuation of assets is considered as the most important task in the management of risks. At a common-sense level, it makes Management aware of situations where there is a need for counter-measures; or it may point out that there are no assets of substantial value, and therefore no protective measures are required.

For the purpose of this study, assets include personnel, premises, hardware, software, the value of the service including revenue etc. (see Table 4.1 in chapter 4). Each asset needs to be recorded in detail, so that the risk manager is able to go on to analyse and assess the risk exposure of the installation. The types of data which need to be recorded for each asset, to permit this later analysis and assessment, are discussed in chapter 4. Broadly, the information needed for the risk analysis phase includes the following:-

an asset's type, location, function and dependencies; and

the information needed for the risk assessment phase includes:-

an asset's replacement cost, replacement lead-time, and the loss to the organisation which would result from its non-availability (expressed as an average daily loss).

In order to analyse the risks to which assets are exposed, the identification of an individual threat and its potential impact upon these assets should next be considered as another important risk element.

Threats

A threat is an essential element in modelling risk, since without a threat a risk does not exist. Several classifications of threats, and of agents which may initiate their occurrence, have been identified in the literature. According to (Parker, 1981; Carroll, 1984), a threat may be intentional, e.g. from espionage or other malicious motives; accidental, e.g. from mistakes or lack of training; or natural, e.g. earthquake, flood, windstorm etc. More recently, the UK National Computing Centre (Elbra, 1992) distinguished between threats which affect the integrity and confidentiality of system software and data, and threats which affect the hardware

components and their accommodation. He related the effects on data and software to threats which could happen logically, and called them logical threats (although logical threats may be initiated by physical agents, e.g. intrusions); and related the effects on hardware and accommodation to threats which could happen physically, and called them physical threats. This relation is unique, because it helps to identify more accurately whether the required analysis needs to focus on data, or on the installation's requirements. Assets to be investigated for relevant effects or exposures can therefore be identified. For example, if the analysis focuses on data, the relevant potential effects might be disclosure, modification or destruction of data. (Elbra, 1992)'s view had previously been supported by (FIPS PUB 65, 1979; SDC, 1979; DOA, 1977).

Because this study deals with computer disasters which could result in damage to IT facilities, and service interruption losses, it is important to identify the associated threats which relate to those effects (see table 4.2 in chapter 4). Such threats can be classified as intentional, e.g. acts of sabotage; accidental, e.g. fire spreading from an adjoining building; or natural, such as flooding caused by rainstorm. These threats usually arise first outside the immediate accommodation of the IT facility, for which previous research has not provided adequate computer disaster prevention guidance or analysis. The identification of these threats requires proper investigation for their potential impact upon assets, to determine their likely effect upon the service provided by the IT installation. Available methodologies (reviewed in chapter 2) do not, in the main, provide clear analysis to show how a particular threat may interact with an asset and other risk entities. For example, risk is seen as being influenced by the nature of a threat to the system, and how vulnerable the system is to such a threat. But how threat and vulnerability is combined to give an indication of risk is unclear (Clark, 1989).

Each threat needs to be recorded in detail, again so that the risk manager is able to go on to analyse and assess the risk exposure of the installation. The types of data which need to be recorded for each threat, to permit this later analysis and assessment, are discussed in chapter 4. Broadly, the information needed for the risk analysis phase includes the following:-

a threat's class, nature, agent, and likely extent of impact; and

the information needed for the risk assessment phase is the threat's expected frequency of occurrence.

Again, as will be seen, the "three environment" approach provides a useful framework for identifying threats, and helps in analysing assets' exposure to them, by forcing a structured examination for all the related interactions and influences.

Counter-measures

This next element in risk identification is to identify and record any counter-measures which have already been provided. (The question of whether to provide new or enhanced counter-measures is first dealt with during the later Control Management phase of this Framework.) Counter-measures may include controls, policies and procedures. They are those safeguards which protect assets against threats, which may cause destruction and/or the denial of service, for example:-

physical access controls against intrusion;

fire protection plans against accidental or intentional acts;

physical or structural barriers against natural and environmental disasters.

Table 4.3 in chapter 4 provides examples of the kinds of preventive counter-measures which have been considered in this study.

As has been noted, data on existing counter-measures is collected in this risk identification phase. During an IT risk management exercise, data on new counter-measures may also be needed. This data, therefore, also needs to be collected. The types of data which need to be recorded for counter-measures are discussed in chapter 4. Broadly, the information required includes its type, location, function, dependencies and cost. In an automated computer disaster prevention system, it is to be expected that a library of available counter-measures (as well as the other classes of risk entity) would be created, as with CRAMM, to avoid the need for repetitively entering large amounts of data of this kind.

The presence of counter-measures (which are considered as assets), if they are found to be effective, may greatly influence the risk exposure of an installation and its component parts. One advantage of the "three environment" approach, is that it provides for the systematic recording of individual counter-measures by their location (and in relation to each other). Thus their relative positions are determined, so that a clear picture of which counter-measures protect which assets from which threats can be deduced later in the risk analysis phase.

3.7.2 Phase 2 - Risk Analysis (Risk Exposure Determination)

In this phase, which is the corner-stone of risk management, the information gathered about risk entities is analysed. This is done to determine whether the interactions and inter-dependencies among assets, threats and existing counter-measures do, in fact, produce a positive level of risk exposure, as a result of which assets are exposed to impact by a disaster, taking account of existing counter-measures. If such a positive level of exposure is identified, the analysis goes on to distinguish the type of impact to which the installation is exposed (i.e. destruction of assets, and/or denial of service).

Therefore, risk analysis is a fundamental step in determining which assets are vulnerable to impact from which threats, by analysing the interactions among risk entities. This has also been called the exposure analysis stage, which determines the assets potentially exposed to a threat, and also analyses their contribution to the IT installation. For example, as a result of the interactions among risk entities, the impact may be upon assets that are irrelevant to the availability of service (e.g. vehicles or warehouses). Equally, the impact may be upon assets that are relevant to the provision of the service (e.g. buildings, plant rooms, computer equipment, electrical power, personnel, etc.). This information can be used later at the risk assessment phase, which requires the calculation of the loss exposure that may result due to the risk exposure of these assets.

The form of analysis which determines these levels and types of risk exposure is based upon the concept of vulnerability, which is next considered.

Vulnerability

Understanding vulnerabilities to threat occurrences is an important aspect of assessing potential losses (Gilbert, 1989). Researchers have recognised the importance of the concept of vulnerability in modelling risks (Moses and Glover, 1988). A workable definition of vulnerability is, however, difficult to achieve (Ottwell and Aldridge, 1989). In general, it has been treated in one of three ways:-

a) as an attribute of assets;

b) as a relationship between assets and threats; or

c) as the absence of safeguards.

Those who treated vulnerability as a system attribute defined vulnerability as weaknesses in the safeguard systems that allow threats to compromise the security

of an asset (Mayerfield, 1988). Thus, while vulnerabilities are viewed as properties of the system, they are considered only in the context of threats that might exploit them. (Bonyun and Jones, 1988) more clearly recognised the relevance of vulnerability exploitation by a threat, and define vulnerabilities by assertions of the relationships between assets (e.g. entities of the internal environment of a system) and a threat agent. This method has been criticised on the grounds that an external factor - such as a threat, change or threat agent - has to be hypothesised. Yet previous authors have not provided a measure, qualitative or quantitative, for vulnerability (Ottwell and Aldridge, 1989).

Those who treated vulnerability as a missing safeguard tended to define vulnerability as an absence of safeguards or controls that would prevent security violations (Gilbert, 1989). Thus, vulnerability in this respect indicates the lack of controls or safeguards; or conversely, controls or safeguards reduce vulnerability. While authors such as (Smith, 1988; Lewis, 1988; Mosleh, 1988) might appear to define vulnerability as the absence of controls, definitions used depend on a prior determination of threats and system properties (Otwell and Aldridge, 1989).

Those who treated vulnerability as a threat/asset relation recognised that a characterisation of vulnerability depends upon both the information system (including safeguards), and some understanding of the threat environment. This view, for example, can be found in work done by (Moses and Glover, 1988; Katzke, 1988; and Schmidt, 1988).

Although earlier approaches have the same general perception of vulnerability, each proceeded differently on how it can be specified and measured. These approaches have a common theme: that the risk to a system cannot be determined without knowledge of how vulnerable the system is to potential threats. It would

seem that a carefully designed combined perspective, examining all risk entities, would yield optimal results.

In this research, vulnerability has been defined as a relationship among the following involved entities.

1. The extent of a threat level at a specified location, taking account of factors contributing to its level or severity (e.g. proximity of hazards, boundaries, elevation, or meteorological and geological conditions).

2. The relative location of inter-dependent assets and controls (including "support" assets, such as the building, staff access, electrical supply etc.).

3. The existence and effectiveness of counter-measures which may reduce the impact of threats upon assets.

These entities must all be recognised in the analysis of risk exposure (i.e. when identifying the impact upon assets, from which destruction or denial of service may result). This analysis is aided by the use of the "three environment" approach, which allows vulnerability to be measured qualitatively, in order to determine levels of risk exposure.

The results of this risk analysis phase are carried forward, along with the risk assessment data collected at the risk identification phase, to the next phase, where the installation's loss exposure is calculated.

3.7.3 Phase 3 - Risk Assessment (Loss Exposure Calculation)

Having identified the levels and types of risk exposure from which destruction of assets or denial of service may result, it is now appropriate to assess the potential

loss which would result if a computer installation is exposed to such risks. The kind of loss which may result may require assessing the direct loss which may result from damage to IT facilities and equipment, and the consequential loss which may result from denial of service. The sum of losses caused by a destruction of assets and the losses due to denial of service is a single loss exposure (SLE), which is the potential cost of a single threat occurrence. Although this information is an important logical step in loss exposure calculation, it does not support decisions about the required investment in cost-effective counter-measures. The concept of risk is often defined in terms of annualised loss exposure (ALE), which is calculated by multiplying the expected loss (the SLE) and the frequency of threat occurrence.

The concepts of SLE and ALE are important factors used in the insurance industry, and are also useful in IT risk management (Palmer and Potter, 1989). How the SLE and ALE are used in this Framework is described in chapter 4.

3.7.4 Phase 4 - The Control Management phase (Cost-Benefit Analysis)

The previous three phases have involved the preparatory work which needs to be done, as a foundation for the taking of well-informed (or supported) Risk Management actions. These actions have to do with the selection and implementation of counter-measures which will create an acceptable level of risk exposure.

IT Risk Managers have historically been faced with difficulty in deciding on the level of investment required for counter-measures, and how they can be cost-justified. Resources have been expended on threats which are not worth controlling, while other major threats (such as computer disasters) receive little or

no control (Gilbert, 1989). The kind of counter-measures selected will depend upon the functions of the assets and their values. For example, in business installations which run high-value services, prevention measures against service interruption may be of primary concern, while recovery measures may play an important role if the cost of prevention measures is so high as to be unjustifiable. Insurance has also been considered as a option in those situations where the implementation of preventive counter-measures cannot be cost-justified. Examples of the cost-benefit analysis of preventive measures, recovery services and insurance are given in chapter 6, where a prototype solution is provided.

The function of this final phase in the Framework is in three parts, as follows.

3.7.4.1 Part 1 - Ascertaining Cost-Justifiable Counter-Measures

The first part involves ascertaining what new counter-measures are cost-justifiable. Cost-justifiability is determined by considering:-

a) the Annual Loss Exposure identified at phase 3 (risk assessment); and

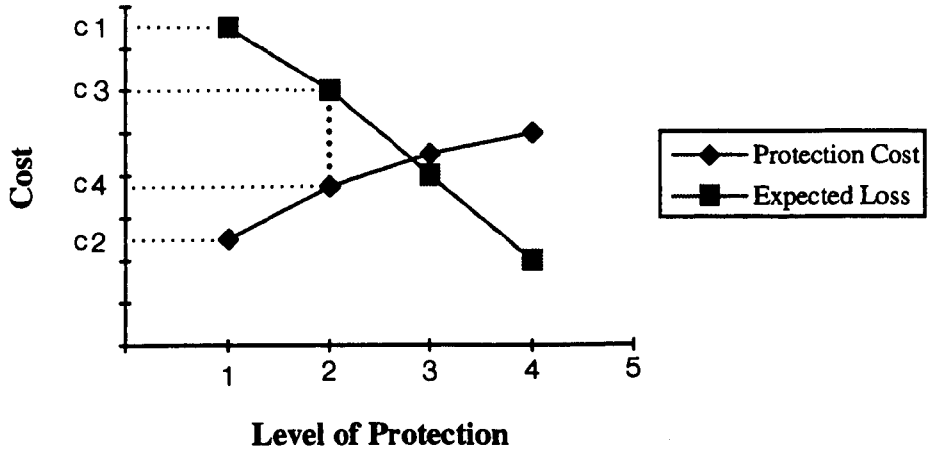
b) the annualised cost (from the risk entity library referred to in phase 1 - risk identification) of providing counter-measures which will reduce that Annualised Loss Exposure.

A counter-measure which is cost-justifiable is one whose annualised cost does not exceed the value of the reduction in Annual Loss Exposure which it will provide (see Figure 3.3 below).

To enable IT Risk Managers to develop a justification for the acquisition of counter-measures, a safeguard cost-benefit analysis is helpful. Cost-benefit analysis is also referred to as trade-off analysis and value analysis. The trade-offs

between security and effectiveness are well known to professional insurers and risk managers. Likewise, trade-offs arise between the costs of control and the benefits of risk reduction. As the expenditure on controls increases, incremental benefits arise in reducing loss exposure (Saltmarsh and Browne, 1983; Gilbert 1989). The optimum point is to spend money on controls only up to the crossover point between the cost of controls and the level of protection. Figure 3.3 below illustrates this concept.

Figure 3.3 - Level of Protection versus Cost



As mentioned above, controls provide benefits to the organisation by reducing its loss exposure. This reduction can be calculated and used to produce a new ALE, the "loss exposure after control implementation". The difference between this new ALE and the original ALE is the "benefit" of the control. If the benefit of a control is higher than its costs, management should consider implementing the control. If the cost of a control is higher than its benefit, then the control should probably not be implemented. Instead, a search should be made for another control to reduce the loss exposure.

3.7.4.2 Part 2 - Selecting the most Cost-Effective Counter-Measures

The second part deals with the selection, from the list of cost-justifiable counter-measures produced at Part 1, of those counter-measure(s) which are calculated to be the most cost-effective, and the implementation thereof. The most cost-effective counter-measure is the one which requires the lowest level of investment, while providing the required level of protection. At this stage, it may be found that the total cost of implementing all of the most cost-effective counter-measures exceeds the available budget. In this case, the recommended cost-effective counter-measures will be listed in a descending order of their contribution to ALE reduction, so that those which provide the best levels of protection, up to a total cost which does not exceed budget provision, can be identified. It will necessarily be for management to decide whether to act on this list, or to adopt a different selection.

3.7.4.3 Part 3 - Implementing Review Procedures

The third part involves the implementation of review procedures which will ensure that the disaster prevention policies are maintained. The common-sense reason for this is to ensure that the acceptable level of disaster prevention which an organisation has achieved at the end of this phase is monitored. Failure to do this may allow that level of protection to be eroded over time, as risk entities change.

3.7.4.4 Phase 4 - Conclusion

This phase sees the IT risk manager enabled to make decisions on counter-measures. Management decision-making is supported by the results of the work in the previous three phases, by the end of which clear information is available on where the installation is susceptible to disaster impact, and on what cost-effective counter-measures will provide protection against such impacts. The "three environment" approach, as will be shown, provides a rigorous basis for this management support. It may also, depending upon how much of the required investment in counter-measures can be afforded within available budgets, provide the basis for yet further guidance. For example, if the optimum mix of counter-measures cannot immediately be acquired, the "three environment" approach will indicate which affordable controls will provide the best achievable level of protection. For example, for most physical threats (such as external flooding) the best levels of protection are likely to be provided by those counter-measures which are outside the premises, since they will protect the premises, including any outside assets, and the building and its contents.

Chapter 4

THE METHODOLOGY FOR IT DISASTER PREVENTION

4.1 Introduction

The previous chapter described a structured Framework to reflect the sequence of actions required in the identification, analysis, assessment and management of computer disasters. Each of the accepted risk management phases and entities were incorporated in the Framework. This chapter describes the Methodology which was formulated based on that Framework.

This methodology is needed so that a structured computer disaster prevention policy can be constructed. One fundamental aim in its development has been to help the IT risk manager in performing computer disaster risk management exercises. Another principal reason has been to ensure the methodology would provide a basis for the later development and implementation of a structured knowledge based decision support system (KBDSS) which is described in chapter 6. Thus, in order to help meet the above objectives, the features of this methodology had to ensure that the following sets of requirements were satisfied.

4.2 Specific Requirements for the Methodology

First, this methodology needed to provide the IT risk manager with an aid for the capture and recording of all of the data and knowledge concerning risk entities which should be considered in performing a computer disaster risk management exercise. This information has already been briefly discussed in the description of the Framework, and is further discussed in this chapter under the headings Asset Identification and Valuation; Threat Identification; and Counter-Measure Identification.

Second, this methodology is needed to aid the analysis applied to the interactions and inter-dependencies between assets, threats and counter-measures, to determine an installation's overall vulnerability. This analysis needs also to consider the factors which could contribute to the level of risk exposure. These factors have not previously received adequate attention, and have already been outlined in chapter 1. Specific illustrations of how these factors may contribute to the extent of risk exposure are considered in this chapter, and are further explored in the prototype solution which is described in chapter 6.

Third, the methodology also needed to help the IT risk manager in the assessment of loss exposure after determining the level of vulnerability. This assessment requires information which also needs to be collected at the risk identification phase. The methodology should support the identification and collection of this information. Such information concerns asset values and threat frequencies. How the methodology supports this assessment, and performs the necessary calculations, is explained in this chapter under the heading Loss Exposure Calculation; and how these calculations are processed to support decisions is shown in chapter 6.

Fourth, as a final important requirement which brings us back to the central theme of this study, the methodology also needs to support decisions on the selection and implementation of appropriate disaster prevention measures. This particular requirement has emphasised the need to search for a suitable technology which could implement the methodology. and provide it to IT risk managers in a usable form. This technology is reported in chapter 5, and its ability and suitability to provide the implementation "vehicle", and to prove the validity of the methodology, are reported in chapter 6. How this requirement is met in the methodology is described under the heading Deciscion Support for Control Management .

The above requirements need to be satisfied during the development of a computer disaster prevention methodology, if it is to be successful. How these requirements can be achieved in practical terms is next considered.

4.3 The Elements which enable a Structured Methodology

A "three-environment" approach has been developed to provide the IT risk manager with a structure for identifying and collecting the necessary information about risk entities, and to help in the analysis, assessment and management of the risks of computer disasters. As will be seen in the following paragraphs, this "three-environment" approach provides powerful facilities:-

a) for ensuring that the basic data about risk entities is collected in the first place; and far more importantly

b) for enabling the complex interactions and inter-dependencies between the risk entities to be fully taken account of during the analysis and assessment phases.

As a further refinement which may only need to be used in some cases, a hazard exposure zoning method was also developed. It is described in detail in this chapter after the "three-environment" approach, but briefly its function is to allow the user to address all of the extreme variations of threat severity which can arise across some situations. As will also be seen in chapter 6, this additional facility was needed when examining external flood risks, because several types of exposure zone can be encountered (as illustrated in Figure 6.1).

4.4 The "Three-environment" Approach

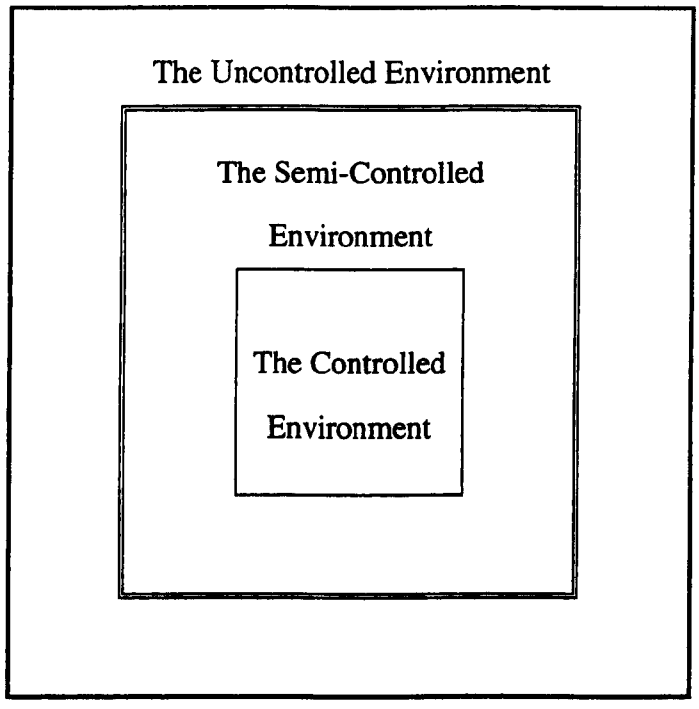
This approach considers the computer installation as residing within three "concentric" environments, as shown in Figure 4.1, namely:-

a) a controlled environment, which is the immediate area occupied by the computer installation;

b) a semi-controlled environment, which is the premises containing the computer installation; and

c) an uncontrolled environment, which is as wide a surrounding area as contains features which influence the degree of threat to which the installation is exposed.

Figure 4.1 - The Three-Environment Approach



This approach allows the IT risk manager to determine the boundaries for each environment, so that features and risk entities which influence risk exposures within it can be identified and analysed. It also assists considering how to protect the installation, by using a "three main lines of defence" strategy to ensure service continuity.

4.4.1 The Role of the Three-Environment Approach

As stated, the central theme of this study has been to address more fully the protection of IT services from interruptions caused by computer disasters. These interruptions are sometimes referred to as denial of service, and can result from

direct damage to IT facilities, such as buildings, plant rooms, computer equipment etc. There are also indirect effects which could, for example, result from the loss of electrical power, water supply, communications, key staff, access to working areas etc. These types of direct and indirect impacts have not received adequate previous attention, and need more thorough identification and analysis than has previously been achieved. For example, a generator located in a semi-controlled environment exposed to a high risk of flood will be a poor or totally inadequate safeguard, depending on whether mains electricity is supplied from the uncontrolled environment using overhead or underground lines. As a further example, the fact that a semi-controlled environment is sited on land very prone to flooding is of very great, or no, significance, depending upon whether man-made means (e.g. stilts) have been used to elevate the structure.

It is recognised that these direct and indirect impacts are influenced by the interactions and inter-dependencies of the following risk factors or entities.

a) the nature of threats to the installation, whether arising in the IT installation's immediate accommodation (e.g. fire in a plant room); in the building (e.g. leaks in roof); or in the surrounding areas (e.g. flood from a river).

b) The locations of assets, and their respective contributions to the IT service (e.g. personnel working in the computer room are key to service continuity, as are utilities serving the building and its operations, staff access to the building, etc.).

c) The location and effectiveness of existing controls which protect against direct damage to IT facilities (e.g. flood barriers outside the building, construction standards for the building itself, fire walls in the computer room etc.); and controls to protect against indirect

impacts (e.g. a standby generator for the computer room, an emergency lift in the building, safe refuge outside the building etc.).

The interactions and inter-dependencies of these risk entities determine to a great extent which assets would be impacted by which threats, taking account of available counter-measures. The results can then be used to identify which types of impact (or risk exposure) may affect the service (e.g. destruction of assets and/or denial of service).

It is clear from the above that the potential effects of a threat impacting an IT installation and its service cannot be determined without knowledge about which risk entities are involved, where and how. It is also clear that the relative location of inter-dependent assets and controls in a computer room is a different matter than for those located elsewhere in the building, or in the surrounding areas. Similarly, the impact of threats arising in the IT installation's immediate accommodation may also be different from those which arise externally. These conclusions have, however, emphasised the need for considering and developing the three-environment approach.

The contribution of the "three environment" approach is that it helps in achieving the specific requirements of a computer disaster prevention methodology which were described above. It is structured to assist IT risk managers to make full use of the recommended framework phases which are described in chapter 3, and helps in the achievement of practical computer disaster prevention policies. The details of the form which this contribution takes are discussed in detail below.

4.5 More Detailed Discussion of the "Three-Environment" Approach

As noted in chapter 3, the Risk Identification phase of the Framework is in three parts. These are as follows.

4.5.1 Asset Identification and Valuation

The use of a check-list or questionnaire approach to Asset Identification is well established. As part of the approach adopted in this research, and outlined in chapter 3, the questionnaire method is refined to ensure that data about assets are collected by "environment". (See Table 4.1)

Table 4.1 - Examples of Assets

Uncontrolled Environment	Semi-Controlled Environment	Controlled Environment
Physical: vehicles storage	The Building	Physical: hardware furniture
Backup Utilities	Building Contents	Personnel
Structural: refuges access	Utilities supporting the Computer Site: HVAC water supply power supply storage facilities	Data and software: system s/w application s/w data files Manuals other Records
Utilities supporting the Building: power water cabling	Building Utilities: lifts lighting electrical mechanical	Disposable: stationery magnetic media The Service

This ensures that each asset's location is accurately recorded, and that the IT risk manager is provided with data in sufficient detail for rigorous analysis. It also, by definition, means that each asset's location is recorded by "environment". Moreover it means that, in analysis, the overall effect of a disaster occurrence can be measured. Equally, the "contribution" of each asset to the service provided by the IT centre is comprehensively evaluated, taking informed account of the interactions and inter-dependencies which can occur between assets, and between "environments", in the event of a threat occurring. Thus, the following necessary data needs to be collected and provided for each asset:-

What is it? - *e.g. buildings, equipment, staff, replacement costs, the service provided, revenue.*

Where is it? *In which Environment?*

Is it duplicated? *e.g. standby generator, back-up HVAC.*

What is its replacement cost? *i.e. the cost (including delivery, commissioning and other extra costs) of a new unit.*

What is the average daily loss to the organisation due to its non-availability? *i.e. what is the daily revenue from those parts of the organisation's services to which the asset makes an irreplaceable contribution?*

What "lead time" is required for replacement? *i.e. the time needed to "make ready", and the supplier's time to deliver and commission the new unit.*

On what other assets is it dependent for support? *e.g. a unit may be dependent upon the electrical supply, HVAC; the computer suite upon protected premises.*

What other assets does its functioning support? *e.g. a barrier's effectiveness protects a building, loss of personnel is essential to the service provided.*

How essential is it to the service provided? *e.g. the building is 100% essential; a warehouse or vehicles may not be.*

On the last point, an asset may make a unique (i.e. irreplaceable) contribution, in which event the service is lost. Its contribution may be removed because of an impact on another asset. Or there may be an asset which could take its place in sustaining the required service.

Asset Valuation must be done accurately and thoroughly before proceeding to the next stage, if full confidence is to be had in any conclusions reached about disaster prevention measures. An important benefit of the "three environment" approach, when applied to asset valuation, is that it enables the more accurate assessment of the impact on individual assets of a threat occurrence. This is because, for example, a given flood level in an outer environment may impact assets located in it, but have no direct effect on assets located in an inner environment. There may, however, be an indirect effect. This consideration is also relevant, of course, at the Threat and Existing Counter-measure Identification stages, which are discussed below. Because, in many cases, conclusions reached about disaster prevention measures will call for investment in counter-measures, those conclusions need to be fully justifiable. The "three environment" approach provides a rigorous structure to ensure that all relevant facts about assets are elicited and available.

4.5.2 Threat Identification

Because of the sheer breadth of the topic of Computer Disaster Prevention, it was decided in this study to concentrate on threats to computer installations which arise

outside the immediate accommodation of an IT installation. In Table 4.2, examples of threats are given which emphasise the type of disasters that this research addresses.

Table 4.2 - Examples of Physical Threats

Name of Threat	Class of Threat	Causes/Agents	Which Environment First affected
Flood	Natural	Heavy rain, hurricanes, thaws	Uncontrolled
	Accidental	Leaks in roof	Semi-controlled
	Intentional	Dam destruction nearby	Uncontrolled
Fire	Natural	Lightning strike	Uncontrolled
	Accidental	Electrical fire in external supply	Uncontrolled
	Intentional	Terrorism, vandalism	Semi-controlled
Windstorms	Natural	Meteorological conditions	Uncontrolled
Earthquake or Volcanic Eruption	Natural	Geological conditions	Uncontrolled
Falling Objects	Natural	High wind removing tree branches	Uncontrolled
	Accidental	Aircraft crash	Uncontrolled
Traffic Hazards	Accidental	Tunnel subsidence	Uncontrolled
	Intentional	Use of explosives vehicle to attack IT facility	Uncontrolled
Building Defects	Accidental	Poor design, materials, construction, floor loading beyond design limits	Semi-controlled

The "three environment" approach aids Threat Identification, and assessment of the potential impact of threats for assets. To ensure, therefore, that threats are also identified with sufficient thoroughness to enable the complete consideration of IT

disaster prevention requirements, the following types of necessary data need to be collected for each threat, again by each of the three environments:-

What is the threat? *e.g. flood, fire, explosion, windstorm, earthquake.*

What is the class of threat? *i.e. natural, accidental, intentional.*

What is the threat agent? *i.e. the entity which might initiate a threat occurrence, e.g. river for flood; electricity for fire; terrorist attack for explosion.*

What is its initial extent likely to be? *e.g. how high a flood level can be expected.*

How often has it historically occurred to that extent? *i.e. consult local records or experts for frequency.*

Are there any known recent factors which would reduce or increase the extent or frequency, and if so to what new extent? *e.g. new local authority flood management measures; newly built adjoining hazard sources.*

Which other environments can the threat also be expected to impact? *e.g. is the level of flooding likely to spread from the uncontrolled environment to the semi-controlled environment?*

Are there already counter-measures which will prevent the threat to the expected extent impacting neighbouring environments? *e.g. flood barriers, building elevation by stilts or earth-fill.*

To continue with the example of flooding, as a threat it can in principle arise in any of the three environments, even though this study, as noted, considers only threats which arise in the external environments. In practise, data concerning the threat therefore needs to be collected for each environment. For illustration:-

a) natural flooding from meteorological causes will usually occur first in the uncontrolled environment; and

b) water damage from an internal water tank will usually occur first in a semi-controlled environment; and

c) water damage from an air-conditioning system will usually occur first in the controlled environment.

Up to that point, only assets in the respective "first" environments may be impacted. Thus, collecting data by environment enables the first point of impact to be identified. And because assets have similarly been identified by environment, the immediate effects of that first impact can also be assessed. Subsequent or indirect effects of the first impact, as they in turn impact on other environments, can also be identified by considering data, already collected, on the operating interactions and inter-dependencies of assets between environments. Over and above this, however, having identified first the environment initially impacted by a threat, one can go on to ascertain whether neighbouring environments are likely to be impacted; and the "three environment" approach forces such a structured examination. For example, the natural flood which arises first in the uncontrolled environment may also be a threat to the semi-controlled environment, depending on the level of water expected. It will not, however, be a threat to the semi-controlled environment if an asset called a barrier of sufficient height exists between the two environments. Similarly, a flood level of a given height in the uncontrolled environment may not directly affect assets in the controlled environment. But the absence of staff access above the water level would produce a real indirect effect.

4.5.3 Counter-measure Identification

A primary objective for the "three environment" approach is to allow the evaluation of the counter-measures currently in place; and also to help identifying any additional counter-measure which may be appropriate for the reduction of risk. The methodology should consider several categories of controls, and some of these are shown in Table 4.3.

Table 4.3 - Examples of Counter-Measures

Type of Counter-Measure	Uncontrolled Environment	Semi-Controlled Environment	Controlled Environment
Zone Selection	Site location within specified Zone, considering geology, meteorology, boundaries, proximity of hazards, elevation.	Building location within site, to reduce exposure to neighbourhood threats.	Computer location within building, to further minimise risk exposure, e.g. for flood, the basement is least desirable.
Safety Regulation	Avoidance of adjoining hazardous activities or operations. Adequate Emergency Services. Development Planning Policies.	Use of space for materials, equipment and working areas. Access and escape routes. Safety training and supervision.	Safe refuge from fire, flood, etc. to protect staff as key to service continuity. Safety training and supervision.

Table 4.3 Continued

Physical Protection	Site structural barriers. Site Drainage.	Building construction standards. Building Drainage. Emergency Facilities. Detection and Alarm Systems.	Computer Room construction standards. Computer Room Emergency Facilities. Computer Room Detection and Alarm Systems.
Backup and Standby Facilities	For Utilities serving the building, main power supply, water supply, telephone lines.	For Utilities serving the Computer Room, Emergency Lighting with independent power supply. water supply tanks.	HVAC systems. Computer. Room Emergency Lighting. Standby Power.

For each counter-measure, we need to record the following necessary data:-

a) What is it? *i.e. against which exposures does it protect, e.g. counter-measures against destruction of assets; counter-measures to protect continuity of service.*

b) Where is it? *i.e. in which environment is it located?*

c) Against which threat(s) does it provide protection? *e.g. fire, flood, explosion.*

d) Up to what level of threat does it provide protection? *e.g. does it protect one or more environments?*

e) On what other controls does it depend for protection itself? *e.g. the integrity of a building may depend on several structural features,*

such as a computer suite's HVAC which depends on a water supply or electricity.

f) Which other controls depend on it for protection? *e.g. a barrier's effectiveness may protect the building's integrity.*

Having the answers to these questions alongside the information on assets and threats, we can proceed to analyse the levels and types of risk to which an installation is vulnerable.

4.5.4 Vulnerability Analysis

To summarise the above, we have recorded information on assets, threats and existing counter-measures, and interactions and inter-dependencies among them, using the "three environment" approach. We know where we have assets and counter-measures, where there are threats, and we can therefore infer which assets are potentially exposed to which threats, in spite of any existing counter-measures. In addition, because we already know about the interactions and inter-dependencies among assets, threats and counter-measures, we can infer or produce an accurate picture of an installation's overall vulnerability. The following paragraphs describe, and chapter 6 illustrates, how this inferencing is accomplished.

The Vulnerability Analysis phase is to determine the levels and types of risk exposure to which the installation is vulnerable. The level of risk exposure shows which assets, in each of the three environments, will be impacted by which types of disaster occurrence. This information provides part of the basis for the later calculation of the potential cost of this risk exposure. For example, if a flood is not

prevented from entering the semi-controlled environment, then all assets located there will be impacted.

In addition, from knowledge of the level of risk exposure (i.e. which asset groups will be impacted), and the type of threat, we can infer the type of risk exposure. The type of risk exposure (see list below) is another part of the basis for the calculation of the potential cost of this risk exposure. Several types of risk exposure classification method, as described in the Literature Review, have been used by other researchers. In this study the following classifications of risk exposure have been adopted:-

a) Destruction of assets, with no denial of service, where the effect is not critical to the availability of service (e.g. vehicles or storage may be replaced by hiring), but the assets will need to be evaluated for an overall loss exposure assessment.

b) Destruction of assets, but with denial of service, where the effect is significant and relevant to the availability of service (e.g. buildings, plant rooms, computer equipment etc.).

c) Denial of service, but with no destruction of assets, where the effect is only relevant to the availability of service (e.g. loss of power, loss of key staff, lack of access etc.).

These categories are mutually exclusive, which helpfully avoids confusion or duplication. By providing for the situation at a) to be recognised, the IT risk manager's attention can be drawn to the relative unimportance of the assets which are at risk. Thus, in reaching later decisions about investment in counter-measures, which may well involve the judgement of priorities, he or she is helped to identify those possible counter-measures which only merit a low priority.

At b), on the other hand, it is clear that physical assets are at risk which are vital to service continuity. It follows that their adequate protection demands a high priority.

In the situation at c), other types of asset are shown to be at risk, not of destruction, but of being unable to carry out their normal functions. However, it is clear that their function is of critical importance for normal service provision, and that therefore their adequate protection also demands a high priority.

These risk exposure types help later in determining the loss exposures which may result as effects on the installation. (Loss exposure is the term which denotes the potential cost of a given disaster occurrence.)

As a summary, the function of Vulnerability Analysis is to draw together the information which has already been gathered on assets, threats and counter-measures, and "present" that information in such a way that the next stage of Loss Exposure Calculation can be executed. As has been noted, the key feature in developing the methodology has been the "three environment" approach. That is what provides a common structure within which all of the relevant considerations about Assets, Threats and Counter-measures can be captured, analysed and assessed. It is new in providing such a structure, which further allows the many types of interactions and inter-dependencies among these entities to be considered before Disaster Prevention measures can be selected.

Again, as was noted in chapter 3, Vulnerability Analysis is the second phase of Risk Management. We now progress to the Risk Assessment phase.

4.5.5 Loss Exposure Calculation

Up to this point, the development of the "three environment" approach which was involved in the Risk Identification and Analysis phases of the structured framework for IT risk management has provided for the collection of much information on the Assets, Threats, Counter-Measures, and their Interactions and inter-dependencies for each installation. This information is necessary input for the calculation of a site's overall loss exposure. In other words, it is now possible, having identified the complete picture of which assets are exposed to which threats, to ascertain

a) what potential loss exposure costs or consequences will continue if no counter-measures are installed;

b) what investment in counter-measures will enable those exposure costs to be reduced, and by how much; and

c) whether the investment in counter-measures will be cost-justifiable.

Previous researchers have used a variety of techniques to "drive" methods of Loss Exposure Calculation (see chapter 2). These techniques have included both qualitative and quantitative methods. When quantitative methods are used, loss exposure calculation is computed in precise numerical terms, a technique which has its origins in the insurance industry. When qualitative methods are used, it is because description rather than calculation is the more practical way of dealing with some risks.

As described in detail in chapter 5, this study has concluded that Knowledge Based System (KBS) technology provides the best platform for developing a Decision Support System for Computer Disaster Prevention. KBS technology supports the use of both qualitative and quantitative methods. The facility for supporting qualitative values has not, however, been used in the Prototype Knowledge Based

Decision Support System because, as stated, it does not set out to deal with logical threats and their effects on data and software. It is expected that data on the frequency of disaster occurrence, and the loss resulting from these occurrences, can be provided by the users.

The first set of activities within Loss Exposure calculation is, logically, the calculation of a Single Loss Exposure.

4.5.5.1 Single Loss Exposure (SLE)

The Single Loss Exposure calculation shows the potential cost of a single occurrence of each identified threat. The result of this calculation has no value in its own right for Decision Support purposes, beyond its use later in the Annual Loss Exposure calculation, which is described in the next section. Information on the following factors is used or derived. The SLE is calculated using the formula

SLE = RC + (ADL*RT)

where

RC = costs of replacing destroyed assets (or groups of assets),

ADL = the average daily loss to the organisation from the non-availability of the assets, and

RT = the time required to restore a normal level of service (expressed in days).

The SLE is calculated using this formula, which is able to accommodate the mutually exclusive Loss Exposure types described above.

Information at RC is only required in situations where assets are exposed to potential destruction, whether denial of service results or not (e.g. loss of vehicles

will not cause interruption of service; while the loss of computer hardware or personnel does). If there is no asset destruction, but denial of service is caused by a power interruption, for example, then the information at ADL and RT is still needed, so that the potential loss from denial of service can be calculated.

The requirement to input asset replacement costs is unlikely to present the user with difficulty. Other required input data, for example, on the consequential losses due to denial of service, is more complex to derive. However, it is accepted practise that service users, rather than IT staff, should provide this information (Pinder and Hover, 1990).

4.5.5.2 Annual Loss Exposure (ALE)

The Annual Loss Exposure calculation annualises the Single Loss Exposure costs, so that informed decisions to invest in counter-measures can be achieved. In other words, the calculation shows how the Disaster Prevention measures recommended later in the Decision Support System can be justified.

The ALE is calculated using the formula

ALE = Freq * SLE

where

Freq = the expected annual frequency of a threat occurring, and is an important element in deriving the ALE. It is expected that this information will usually be available to the user from historical records (FIPS PUB 31, 1974).

SLE = the calculated cost of a single threat occurrence, which has already been calculated as shown above.

The Annual Loss Exposure values then become vital information for use in the final, Control Management, phase of Risk Management.

4.5.6 Decision Support for Control Management

The development of a Decision Support System to enable IT Risk Managers to implement Disaster Prevention policies was the aim of this research. The description of the Prototype System in chapter 6 covers in detail how the IT Risk Manager's selection of counter-measures is guided, taking account of:-

a) what investment in counter-measures is needed to protect against the exposures which have been identified; and

b) what level of investment in counter-measures will be cost-justifiable.

Control Management deals with the selection and implementation of counter-measures. The most cost-effective counter-measure is the one which requires the lowest level of investment, while providing the required level of protection. It may be that the total cost of implementing all of the most cost-effective counter-measures exceeds the available budget. To cater for this situation, the recommended counter-measures will be listed in a descending order of their contribution to risk exposure reduction. It will naturally be for management to decide whether to act on this list, or to adopt a different selection.

The "three environment" approach has provided a rigorous foundation for this decision support. It may also provide the basis for yet further guidance. If all of the suggested counter-measures cannot be afforded, the "three environment" approach will show which will provide the best achievable level of protection. For example, for external threats the best levels of protection are likely to be provided by counter-measures in the outer environments.

4.6 The Hazard Exposure Zoning Method

The "three-environment" approach enables the evaluation of risk exposures in the individual areas (environments) surrounding an IT installation. For external threats, extreme variations of threat severity may typically be caused by geographic, topographic, architectural and engineering factors. The following examples illustrate this point.

a) The severity of the threat of external flood varies substantially depending on a site's proximity, and elevation in relation to, large bodies of water; and on the availability and location of natural or man-made barriers.

b) The degree of susceptibility to windstorm damage will vary depending upon the site's own elevation, and any protection available from surrounding hills, buildings, trees etc.

In order to ensure that the Methodology can cater for such interactions and inter-dependencies among risk entities in differing types of area, the method of hazard exposure zoning was introduced. The method allows:-

a) threat levels,

b) the location of assets, and

c) the efficiency of existing counter-measures

to be modelled and assessed in such potentially "interactive" situations. This research covers new ground, by representing the interactions and inter-dependencies of these risk entities, and allowing counter-measures to be rigorously evaluated. The IT risk manager needs access to expert knowledge (e.g. on

flooding) when identifying risk exposures. This research has needed to address the question how to represent the specialist knowledge in a meaningful form.

One way of describing the form of representation of risk entities provided by the "three-environment" approach is to compare it to a map which has boundary lines (in this case, the boundaries of the individual environments) and shows the positions of the risk entities. If a threat exists within a "boundary" it can be recorded as such. The hazard exposure zoning method also shows the potential severity of a threat, and where its area of impact overlaps the (environment) boundary lines. A cartographer would use contour lines to show hills, valleys etc., and the function of hazard exposure zoning is the same. It allows the user of the methodology to model the potential severity, and area of impact, of a threat by representing degrees of severity as contour lines represent hills and valleys. The combination of the "three-environment" approach and the hazard exposure zoning method thus allows all of the factors which influence each installation's risk exposure to be clearly modelled. Figure 4.2 below is an illustration of some of the kinds of geographical, topographical and engineering factors which the hazard exposure zoning method is designed to represent. In the case shown, the relative positions of the IT installation and nearby

a) hazards in the form of a river, railway, highways, chemical plant and airport; and

b) counter-measures in the form of the elevation of the building and its access; a fire station and electricity generating station.

Data derived from the structured approach described above can be processed in a decision support system for Risk Management, which guides the user toward correct decision-making by:-

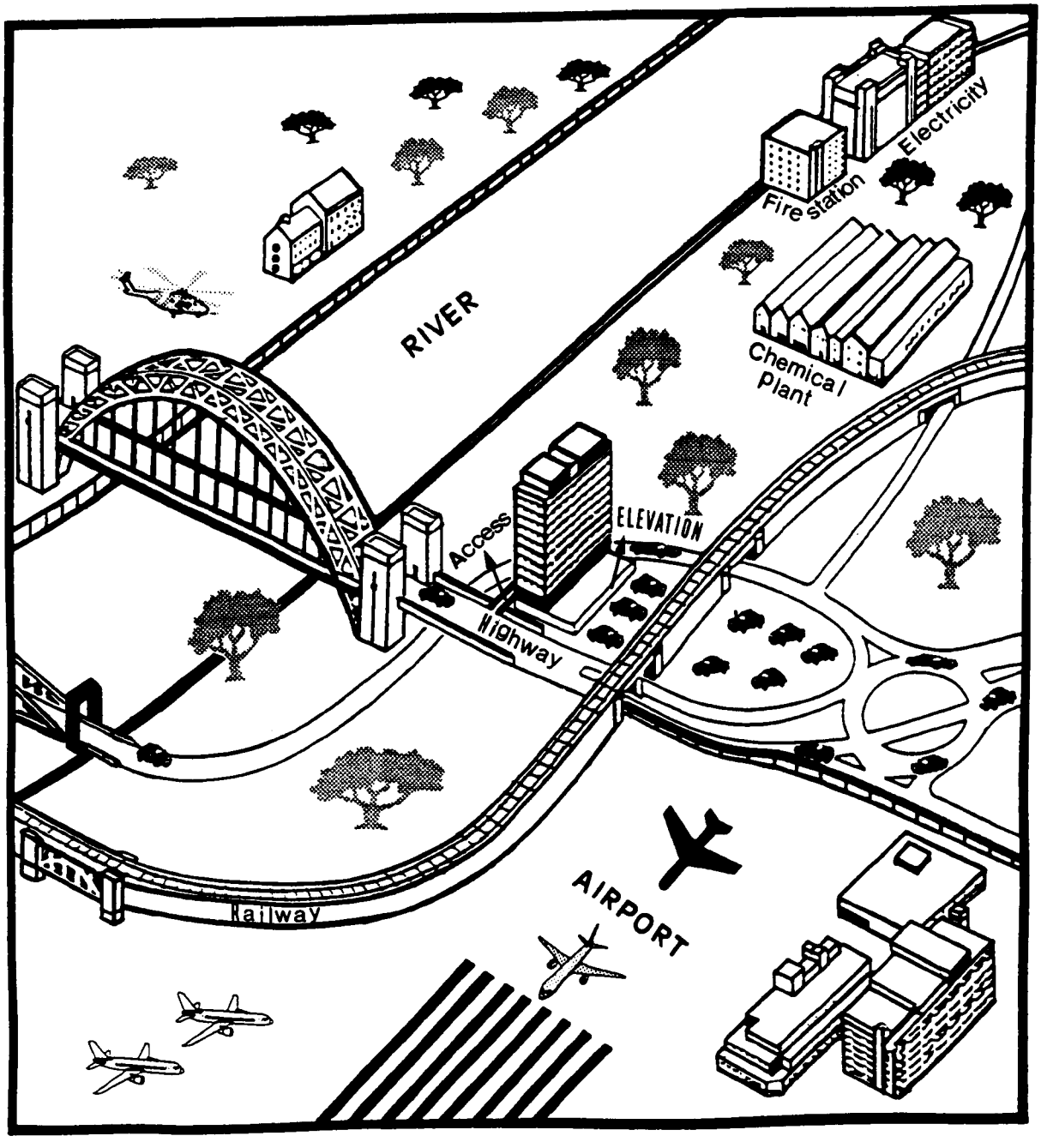
a) collecting the variable data which is peculiar to a particular situation;

b) comparing this variable data with that from generically similar situations; and

c) producing recommended decisions which result from analysing and comparing the data by applying heuristics supplied by domain experts.

What is new in this research, and offers significant benefit for the IT community, is the much wider scope which can be given to IT Risk Management; and the opportunity for practical disaster prevention policies.

Figure 4.2 - An Illustration of the Geographical, Topographical and Engineering factors represented by the Hazard Exposure Zoning Method



Chapter 5

KNOWLEDGE BASED SYSTEMS (KBS) TECHNOLOGY

5.1 Introduction

As has been stated, the principal objective of this study has been to provide IT risk managers with a solution which can be used to implement a computer disaster prevention policy. The Framework and Methodology which have been developed have each represented a significant step towards that goal. The next step which was required, which this chapter addresses, was to identify which technology best supports the development of a system which would facilitate the construction of computer disaster prevention policies. That system would need to encapsulate the Framework and Methodology in a system which would be suitable for use by IT risk managers. The more detailed considerations at this stage of preparing to build the system are described below, under the following headings.

The functionality required.

The reasons for providing a decision support system.

The suitability of KBS to provide a solution..

Concepts of KBS technology.

KBS development languages and tools.

Building a KBS.

5.2 The Functionality Required

The functionality required of a technology which will support the development and delivery of the required solution was analysed under the following headings.

5.2.1 The data to be stored and processed

The data which an IT risk manager needs to consider has, almost exclusively, to do with risk entities. These include assets, threats and counter-measures. Each of these will have a number of attributes. For example, an asset's attributes include its location, replacement cost, service contribution and dependencies on other assets. A threat's attributes will include its probable frequency, its agents, its likely impact. A counter-measure will have its location, its effectiveness and its investment costs. Assets have shared attributes, as well as dependencies and other interactions and inter-dependencies among themselves, and with threats and counter-measures. This is correspondingly true of threats and counter-measures. A requirement of the technology to support the delivery of the required solution will, therefore, be that it allows the risk entities to be represented as objects with attributes, and with rules which govern their interactions and inter-dependencies.

5.2.2 Domain Expertise

The range of threats which may cause computer disasters is wide, as has been noted. The detailed understanding of each class of threat, its agents, probabilities, impact and appropriate counter-measures, is in itself a specialist domain. For example, an expert in flood protection is not likely to be able to advise on fire

prevention; manuals and regulations on flood protection are unlikely to contain any advice on precautions against theft. The expertise across the range of threats involved is not likely to be directly accessible to IT risk managers, and may well be scarce. The designated IT risk manager is not likely to be expert in any of these other fields, yet in order to reach properly based decisions on these matters, he or she needs access to such knowledge, regulations etc. He or she may, in addition, not be fully conversant with the full range of concepts and practices involved in Risk Management. A further requirement of the selected development technology, therefore, is that it supports the representation of specialist knowledge, regulations, heuristics, etc. The ability for a non-expert to be guided to correct selections and decisions aided by specialist expertise is a key requirement of the solution.

5.2.3 Consistent Quality

As has also been noted, the selection of computer disaster prevention controls and actions has to be made among a number of high-value factors. The investment values of large IT installations will be high, as may be the significance of the services they provide, and the costs of counter-measures to protect them. For these reasons, it is critically important that decisions on disaster prevention controls are consistently correct. The technology selected to deliver the solution will, therefore, need to provide an assurance of consistently correct decision support.

5.2.4 Decision Support

It was clear that a fundamental requirement of the solution to be developed was that it should provide effective assistance, or support, for the decision-making process involved in computer disaster prevention. There is already an established class of IT systems which are termed Decision Support Systems (DSS). DSS have been described as follows.

> "Decision Support Systems (DSS) couple the intellectual resources of individuals with the capabilities of the computer to improve the quality of decisions. They are computer-based support systems for management decision makers who deal with semi-structured problems."
>
> The following definitions indicate the four major characteristics of DSS:-
>
> > DSS incorporate both data and models.
> >
> > They are designed to *assist* managers in their decision processes in semi-structured (or unstructured) tasks.
> >
> > They *support*, rather than *replace*, managerial judgement.
> >
> > The objective of DSS is to improve the *effectiveness* of decisions, not the *efficiency* with which decisions are being made. (Turban, 1990)

5.2.5 Ease of Use of the Solution

A key element of the objectives of this research has been that the resulting solution should be readily usable by an IT risk manager. This means that the system, while

meeting all of the criteria described immediately above, should be developed to be "user-friendly" in a few special senses. For example, since an IT risk manager is likely to be a relatively expensive employee who also has other responsibilities, the system should be effective and efficient in use. Its HCI facilities should allow for the fast entry of the minimum amounts of input needed. The use of hypertext, for example, can save time spent on referring to documents, by allowing explanation facilities to be "called down" by one key-stroke. The suppression of questions which can be inferred to be irrelevant from previous data input will save time wastage. The use of "point-and-click" methods and multiple-choice menus can also save time at data input. The production of reports which only contain output which it is necessary for the reader to study will also lead to the effective use of time. A system which encourages, rather than discourages, use will be more readily accepted. Such "performance" aids are also a factor for consideration in selecting the best technological mix for the delivery of the intended computer disaster prevention system.

5.3 The Reasons for providing a Decision Support System (DSS)

In brief, there are two main reasons for seeking to provide a DSS for IT risk management. These are:-

1. The sheer significance, complexity and width of the subject area's issues.

2. The objective of supplying a solution which can be used by a designated IT risk manager, who cannot be expected to have expertise in all of the subject areas, but who needs to be able to reach decisions of a consistently high quality.

The significance and complexity of the subject matter referred to becomes apparent when one considers that a successful computer disaster prevention policy can only be fully implemented after the thorough consideration of all of the following.

a) All of the phases (described in chapter 3), and rigorous examination, involved in a risk management exercise.

b) The potentially wide range, and investment value, of the risk entities which are likely to be present at any major IT installation (including, for example, a range of services with differing operational and service value requirements; a complex and expensive hardware configuration, with varying inter-dependencies among its components.)

c) Equally, the wide range of potential risks/counter-measures which threaten/protect an IT installation, has already been discussed. As reported in chapter 4, it was found necessary to develop a "three environment" approach. This forces the rigorousness needed, but it also reflects the complexity of the risk entity issues.

d) The thoroughness and wide scope of the expert knowledge which the IT risk manager needs to access if correct assessments of risk exposure, appropriate selections of counter-measures etc., are to be made.

e) The application of cost-benefit analysis - a discipline in its own right - which is needed to ensure that decisions about major investments in controls are a soundly balanced judgement of risk exposure, on the one hand, and asset values, on the other.

As can be seen from all of the above, computer disaster prevention is not just a complex area: it is a complex set of complex facets. A manager who is responsible

for implementing a computer disaster prevention policy needs access to all of the kinds of specialist knowledge referred to at a) to e) above, if high-quality decisions are to be made. It became clear, during this research, that a system which would meet the requirements was very close to the descriptions of a DSS in (Turban, 1990). The use of systems which give the manager access to computer power, for handling large amounts of information, is also well established.

For these reasons, it was decided that a computer-based DSS was the best basis upon which to develop the solution which was the objective of this research.

The above paragraphs, without needing to be a detailed requirements specification, illustrate the key processing and "user" criteria for the selection of a suitable technology for the development of the solution. It became clear to this researcher at an early stage that a key requirement of the system to be developed was that it should be able to encapsulate large amounts of domain-specialist knowledge. This led naturally to the conclusion that a development approach based upon Expert Systems, or Knowledge Based Systems (KBS), methods would be the most suitable.

5.4 The Suitability of KBS to Provide a Solution

The following check-list, from Guidelines for the Introduction of Expert System Technology (DTI, 1990), sets out the criteria for judging whether a task justifies a KBS solution. If the job has some of the following attributes, a KBS could be relevant. *(Notes on how a computer disaster prevention system relates to these criteria are in italics.)*

The task is rule based

Regulations, company guidelines and rules of thumb can be encoded. From these, each time, the relevant ones are used by the KBS in its operation. Because of the separation of the rules and reasoning mechanism in a KBS, it is easier to debug and maintain than a conventional program. *(A Risk Management study is carried out within a rule-based structured approach; the counter-measures for individual threats are in the knowledge of the appropriate experts.)*

The task involves analysis

A situation occurs for which there are several possible explanations, or a range of options must be placed in order of desirability. *(Cost-benefit analysis is used to determine what is the best choice of investment in counter-measures after analysing what reduction in threat exposure level will be provided. As explained in detail in the Framework and Methodology chapters, it is also necessary for the system to analyse very complex risk entity interactions and inter-dependencies.)*

The rules change frequently

Some applications are beset by ever-changing rules for which traditional programming techniques are unsuitable. The separate grouping of the rules in a KBS makes modification easier to deal with. *(Rule-sets for the selection of counter-measures may change every time that a supplier changes the specification of the controls which he or she supplies.)*

The task is performed infrequently

Mistakes easily occur in tasks which are undertaken rarely, and simple errors can multiply. KBS can lead infrequent or novice users through

unfamiliar territory, and explain their conclusions at any stage. *(After the initial implementation of a disaster prevention policy at an individual IT installation, subsequent reviews may not be needed more than once per year.)*

The task requires more rapid decision making

Sometimes the task rules are too complex for a speedy human decision. In a narrow field, a KBS can process the necessary information very fast, often in minutes instead of hours, with much wider access. *(Because of the complexity of the range of threats, of risk entity interactions and inter-dependencies, and of counter-measure effectiveness in varying situations, the implementation of a disaster prevention policy at an individual IT installation is too complex for speedy human decisions.)*

The task involves inexact data.

Apart from some aspects of science, engineering and finance, the bulk of day-to-day decision making is qualitative. It is advisable to avoid applications involving uncertainty, but if it is impossible to circumvent it, KBS may be able to deliver a result which takes this into account. *(Many of the considerations involved in decision making on computer disaster prevention need to be expressed qualitatively, e.g. the level of vulnerability, or risk exposure.)*

The task uses dynamic data

For applications in which the input data is changing from moment to moment, early awareness of underlying trends can be vital. KBS operating in real-time are able to use rules to rapidly validate the input data and respond immediately. *(To enable the IT risk manager to see the*

comparative effects of selecting varieties of counter-measures, this system needs to be able to work interactively during consultations.)

Repeated decision making is required

Tasks of this kind, where decisions are required often but at a fairly trivial level, can lead to boredom and error. By removing the repetitious elements from the work, the KBS allows the expert to concentrate on the exceptional cases. *(At the level of individual hardware assets and determining their inter-dependencies, the number of decisions to be made can be high, and erroneous decisions could have significant consequences.)*

Thus, if the form of the task is broadly any of these:-

- Advisory, classifying a static situation.
- Diagnostic, interpreting data.
- Design analysis, critically examining structures or relationships.
- Scheduling, organising resources.
- Monitoring/controlling, using real-time data.

and if some of the following characterise the task, then a KBS is likely to be appropriate:-

- Meets a real business need.
- Is part of an essential activity.
- Occurs in a stable environment.
- Takes small amounts of time.
- Occurs only infrequently.

- Subdivides easily.
- Has existing/easy links to other systems.
- Offers a tactical solution.
- Involves immediate decision making.
- Involves repeated decision making.

(DTI, 1990)

It is clear, from the comparison of the requirements of a system for computer disaster prevention with the above criteria, that the problem is suitable for solution by the use of KBS technology.

5.5 Concepts of KBS Technology

Ed Feigenbaum of Stanford University, one of the leading researchers in expert systems, defined an expert system as:

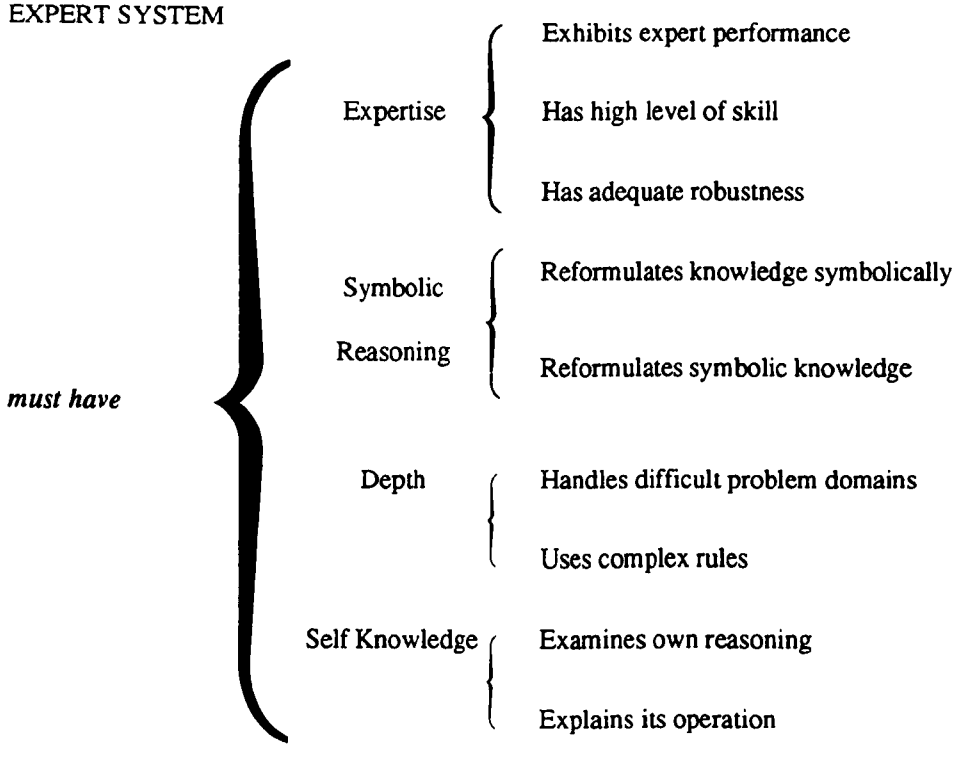
> An intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution. Knowledge necessary to perform at such a level, plus the inference procedures used, can be thought of as a model for the expertise of the best practitioners in the field. (Harmon and King, 1985)

The term "expert systems" is unfortunate because expert systems are a species of KBS, although the two terms are often used interchangeably. Currently most knowledge engineers refer to their systems as KBS (see Appendix D for KBS terms and definitions).

5.5.1 Features of KBS Developments

Key features which differentiate KBS from conventional systems have been identified as shown in figure 5.1.

Figure 5.1 - Features of KBS



(Waterman, 1986)

Whereas a conventional program is algorithmic in nature and requires a complete set of data to produce a unique solution, a KBS is conceptual in nature, can function with an incomplete set of data, and may produce several solutions

(Bielawski and Lewand, 1988). Table 5.1 below is a further illustration of the basic distinctions between KBS and conventional programs.

Table 5.1 - Basic Distinctions between KBS and Conventional Programs

Conventional Program	Knowledge Based System
Requires a complete set of data	Can function with an incomplete set of data
Uses algorithms	Uses heuristics or rules of thumb
Produces a unique solution	May produce several solutions
Generates results that are certain	May generate uncertain results
Lends itself to a top-down approach to development	Accommodates a bottom-up development methodology

(Bielawski and Lewand, 1988)

Additionally, KBS are distinguished from conventional programs by their method of development. Traditionally, most software engineers have adopted a top-down approach to software development: the project is broken down into several smaller projects, each of which may in turn be further modularised. An effective methodology, the top-down approach requires a vision of the overall structure of the problem at hand and an awareness of the relationship among the various modules that work together to solve the problem. In KBS development, this vision is often blurred. Frequently, it is not until all the expert's knowledge is entered into the system that the structure of the knowledge becomes evident. Experts may be able to offer advice without consciously being aware of the organisation of the knowledge they possesses which enables them to give the advice. For this reason,

the top-down approach to KBS development is often less effective than it is in the development of conventional software (Bielawski and Lewand, 1988).

5.5.2 The Architecture of a KBS

KBS are, therefore, interactive computer programs that incorporate the knowledge and skills of human experts, gained from many years of experience in a particular field, to assist users to solve problems in that field (in the case of computer disaster prevention, such as flood protection or fire prevention). Two of KBS main components are referred to as the knowledge base and the inference engine. The knowledge base stores all the knowledge and rules to solve the problem. It is a collection of facts and rules, as for example the US Flood-Proofing Regulations referred to in chapter 6. The inference engine compares information supplied by the user with knowledge contained in the knowledge base, and deduces whatever conclusions may logically follow. The inference engine is generally domain-dependent.

5.5.3 The Functional Elements of a KBS

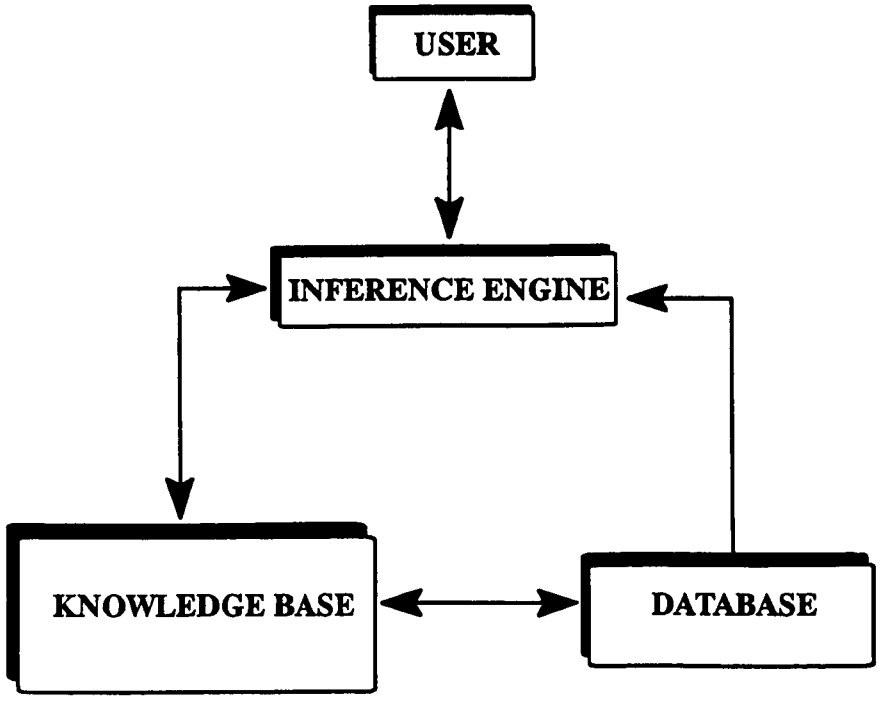
Once the knowledge is represented using rules, an inferencing mechanism must be chosen to organise and control the steps taken to solve the problem. For IF ... THEN ... rules there are two ways of reasoning: backward-chaining and forward-chaining. (The relevance of backward- and forward-chaining to this study's objectives is explained in chapter 6.)

In the backward-chaining reasoning style, the system starts from a hypothesis (or goal) and works backwards, through the rules in the knowledge base from their

THEN parts to their IF parts, toward a conclusion which might or might not be supported. This reasoning strategy is often referred to as goal-driven.

In forward-chaining, the inference mechanism compares the information in the database with the IF parts of the rules in the knowledge base. If a comparison reveals a match between information in the database and an IF part of a rule, that rule *fires*, which means that the THEN part of the rule has been added to the database. This process is repeated until no matches occur between facts in the database and the IF parts of the rules in the knowledge base. This inference strategy is referred to as data-driven, because a system using this strategy starts by gathering all the facts pertaining to the problem, and then considers the rules in the light of this data collection. Figure 5.2 shows the structure of a KBS.

Figure 5.2 - KBS Structure



5.6 KBS Development Languages and Tools

KBS can be built in almost any programming language. Expert System shells, which are KBS emptied of their knowledge base, provide another means for building KBS, but the constraints imposed by their design usually make them applicable to a single class of problem.

Symbolic manipulation languages, such as Lisp and Prolog, are designed specifically for artificial intelligence applications, and have many advantages over conventional, procedural languages such as Fortran and Pascal. It is widely recognised that the majority of serious KBS are written from scratch, rather than built using an expert system shell. The question of which language is more suitable for KBS development had therefore to be considered.

Lisp is the best choice according to the American view. Prolog is best from the European view. Perhaps the best conclusion that can be drawn at this stage is that any artificial intelligence language will be better for building a KBS than a conventional language.

Prolog is usually implemented as an interpreter, which means that changes in a program can be made and tried rapidly. It is also quite easy to build data structures with Prolog. It provides powerful means of searching, pattern matching and list manipulation, and can handle very complex tasks. Backtracking is built into Prolog, and recursion is an important feature of the language.

As an implementation language, Prolog is preferred to Lisp for rule-based systems, because its code is more compact (an important consideration for a personal computer application). According to (Patterson-Hine and Koen, 1987), PATREC was first coded in PL\1. Lisp and Prolog were at first not available to the author. More recently, PATREC has been implemented first in Lisp, and then in Prolog.

(Cuadrado, 1985) described the experience of using two languages, first Ada then Prolog, while working on data-flow models for high-performance signal-processing systems. The Ada code was around 50 pages long, and ran reasonably well. Its Prolog counter-part was only 5 pages long, and surprised the author by running faster than the Ada version.

The procedures required in the solution to be delivered from this research are very easily implemented using the Prolog artificial intelligence programming language. Prolog is an abbreviation of PROgramming in LOGic, and is based on the principles of first order-predicate logic. It was invented around 1970 by Alain Colmerauer and colleagues at the University of Marseilles. The first Prolog system is credited to Colmerauer and Robert Kowalski (Imperial College), both involved in research in logic programming and automated theorem proving. Prolog is a declarative language. It allows a developer clearly to describe the problem to be solved, but not necessarily how it is to be realised computationally, and the language itself will be able to provide the solution. It can also be used procedurally. The suitability of Prolog as an expert system building tool was demonstrated by (van der Gaag, 1986). Prolog's acceptability was underlined by the Japanese decision to base their ICOT Fifth Generation Computer Systems program on Logic Programming in 1982.

5.6.1 The Selection of a KBS Toolkit

The particular suitability of KBS technology for the development of the solution required had therefore become clear. Further, it became apparent that Prolog was potentially the most suitable programming language, because the functionality which it offers is a close match to this research's requirements. Prolog itself is a programming language in its own right, and a requirement to learn it would have

added impracticality to the time scale for this project. It was therefore decided to seek a toolkit, which would ideally combine the powerful functionality of Prolog with ease of use.

In proceeding to identify what KBS toolkit would be the most appropriate, it was noted that most KBS development tools fall into three general categories, namely:-

1. rule-based tools, which require the rules to be entered in an if ... then ... form;
2. inductive shells, which allow the input of a descriptive set of examples, from which the tool will construct a set of rules; and
3. hybrid tools, which combine rules and induction, and add other features to help structure knowledge.

(Bielawski and Lewand, 1988)

From their combination of rule-based and induction techniques, hybrid tools offer the more flexible and powerful building environments, but because the methods used to accomplish this varies among vendors, it becomes difficult to talk about such tools generally (Bielawski and Lewand, 1988).

Before the emergence of KBS shells and other toolkits, the process of building a KBS was a formidable three-stage task, as illustrated below.

Stage 1

Establish the Structure of the System:

a) learn a high-level programming language;

b) design a knowledge base and an inference mechanism;

c) devise a method of handling uncertainty;

d) implement explanation facilities.

Stage 2

Capture the knowledge:

a) interview the person whose knowledge you intend to duplicate;

b) fill the knowledge base with rules.

Stage 3

Develop an interface.

Today, an alternative to labouring with Stage 1 is available: use a shell or toolkit (Bielawski and Lewand, 1988).

For all of the above reasons it was decided to use a KBS toolkit for the development of the Prototype Knowledge Based Decision Support System for computer disaster prevention which accompanies this thesis. The specific toolkit selected was the **F**orward **L**ogical **EX**pert system (***flex***^{tm}) from Logic Programming Associates (LPA) of London, UK, for the following reasons.

flex^{tm} is a hybrid expert system toolkit which:-

a) supports frame-based data representation, sophisticated full rule-based programming, and procedural programming;

b) has its own English-like Knowledge Specification Language (KSL) which enables the development - without the need to learn Prolog - of knowledge and rule sets which are easy to read, validate and maintain because knowledge is stated in a natural manner;

c) also includes

i) a frame hierarchy with multiple, specialised and negative inheritance;

ii) interleaved forward and backward chaining rules;

iii) an automatic question and answer sub-system;

iv) explanations;

v) data-driven programming via daemons; and

vi) an intelligent syntax analyser; and

d) seamlessly includes the full underlying Prolog language, unlike most shells which have only a limited sub-set of some underlying language.

flex^{tm}, as with all of LPA's current products, is portable across 386 PCs running MS DOS or Windows and Apple Macintosh platforms, and has been ported to a number of UNIX platforms.

Although the development of a Prototype Knowledge Based Decision Support System was sufficient to satisfy the objectives of this research, it was kept in mind that one of the useful functions of any prototype system is to form the basis of a later full production system. Therefore, although the prototype system did not use all of ***flex***$^{tm'}$ functionality (no daemons, constraints, launches, synonyms or watchdogs are used), it was prudent to select a toolkit which had all of the features which might be needed to take the prototype that large step further. The features of ***flex***^{tm} which were of value to the prototype, as described in chapter 6, were:-

frame-based data representation, and hierarchies with multiple, specialised and negative inheritance,

rule-based programming,

KSL, for easy-to-read knowledge and rule sets,

forward and backward chaining rules, and

explanation facilities.

The check-list below is a summary of the criteria for selecting a KBS development tool. After a thorough examination, it was found that ***flex***^{tm} satisfied all of these criteria.

1. Fit of the tool to the problem.
2. Effectiveness of the developer interface.
3. Effectiveness and friendliness of the user interface.
4. Integration capability with existing programs and databases.
5. Run-time licensing for delivered systems.

(Bielawski & Lewand, 1988)

5.7 Building a KBS

The term Knowledge Engineering was first adopted by Feigenbaum and his colleagues at Stanford University while developing their first expert system (Harmon & King, 1985). It was used to denote the process through which an expert system can be developed. They also started using the term Knowledge Engineer to describe the person who develops such systems. At that time, it was assumed that the only way to develop an expert system was by obtaining knowledge from recognised human experts. Recently, however, many systems have dealt with difficult decision-making situations, and are hardly just the equivalent of a human expert. To avoid suggesting that such systems only capture the knowledge of human experts, the term Knowledge Based System is a more appropriate designation, and is replacing the term Expert System.

Knowledge engineering involves the following activities.

1. Identification and refinement of the knowledge needed to solve a particular problem, referred to as Knowledge Acquisition.
2. Organisation of the acquired knowledge into a knowledge base which represents the system developer's understanding of the domain referred to as Knowledge Representation.
3. System implementation, using an appropriate programming language or development tool.

5.7.1 Knowledge Acquisition

Knowledge of a domain may take many forms. It is rarely formulated in a way which permits its simple translation into a program, which means that it is not initially in a form usable by the developer. The process of translating the knowledge needed to solve a problem, obtained from an expert or other source of expertise, and converting it into a program, is an important and difficult task.

Potential sources of knowledge include human experts, textbooks, journal articles, technical reports, conference proceedings, databases and the system developer's own experience. It has been stated that the bottleneck in building expert systems is knowledge acquisition (Hayes-Roth et al., 1983; Hart, 1986)

The objective of knowledge acquisition is efficiently and thoroughly to extract rules, facts and the necessary data for the construction of a knowledge base. If knowledge involves how and why experts approach a problem in a certain way, knowledge acquisition includes the process of encoding the relevant rules, facts and data, so that they can be used by the KBS.

5.7.2 Knowledge Representation

In any type of KBS, a key issue is how the knowledge will be represented. The way that this knowledge is structured in a program is referred to as knowledge representation. Several techniques for representing knowledge in a knowledge base are in use. Three methods which are employed frequently are semantic nets, frames and production rules (Waterman, 1986).

Selecting a suitable representation for the domain problem is one of the first problems to be encountered in building a KBS. The Prototype Knowledge Based Decision Support System developed as part of this study uses both frames and production rules, and examples of these are shown in chapter 6 and Appendix A. Production rules are conditional statements that describe actions which are to be performed when specific conditions are true. Rule-based representation is probably the closest to the way a human expert approaches solving a problem. The representation is deliberately simple and uniform, to facilitate the reading and manipulation of the knowledge base.

5.7.3 System Implementation

One very helpful feature of KBS programming languages and toolkits is that they encourage the use of an incremental prototyping approach to system development. This feature is particularly useful when domains of some complexity are to be represented. The developer does not, during the prototyping stages, attempt to transfer to the system the entire body of the knowledge which will eventually be represented, but only representative samples. This prototyping approach has at least four advantages, as follows.

1. A first prototype enables the developer to judge whether the system is feasible. A prototype thus provides the opportunity to "prove the concept", or test the validity of the solution to the problem.

2. A prototype allows the developer to test the suitability of the development tool which has been selected. It may, for example, be revealed that the knowledge representation scheme, the control or inference mechanisms of the tool, or its user-interface mechanisms are inadequate.

3. The first prototype will provide a guide to the amount of time needed to build the entire system. This, in turn, will assist in determining the cost-justifiability of the full system.

4. Prototypes can be helpful in gaining the commitment and support of the full system's ultimate users. The fact that they are able, at an early stage, to see a prototype system actually working is a more convincing argument for the full system than any amount of cogent and well-prepared written or oral presentation. As a result, the knowledge engineer is likely to receive their willing assistance in acquiring the full range of knowledge to be represented in the completed system, in designing effective user interfaces, etc.

To complete the development of a full KBS, after an appropriate number of prototypes, the knowledge engineer's work needs to include the following.

a) Building, and refining after comments by the domain expert(s), the complete knowledge base.

b) Refining the user-interfaces, after use of the prototype and comments on it by the intended users.

c) Developing any required database, communications etc. interfaces which may be required.

d) Documenting the system, both for users and to assist the subsequent maintenance and up-dating of the system.

e) Training the users, and those who are to be responsible for the subsequent maintenance and up-dating of the system.

As can be inferred from all of the descriptive paragraphs above, a working KBS is only achieved after a number of elements of work, each of which requires its own set of specialist knowledge. The Knowledge Engineer is now recognised as a specialist in his own right, who has had to acquire skills in knowledge acquisition and representation. The programmer who has to implement the knowledge engineer's solution (whether in Prolog or using some other development tool) is a second acknowledged professional. It follows that completing this research, which included building the Prototype Knowledge Based Decision Support System for computer disaster prevention, the researcher was faced with the added burden of acquiring this additional range of skills.

5.8 Concluding Remarks

The considerations involved in, and the reasons for, selecting Knowledge Based System technology as the basis for developing a Decision Support System for Computer Disaster Prevention have been described above. In brief, having analysed the Framework and Methodology for the solution (as described respectively in chapters 3 and 4 above), it was clear that conventional, or procedural, programming languages would be unsuitable because of their limitations. It would, for example, have been inefficient and unnecessary to

develop knowledge representation and inferencing mechanisms in those languages, because they are already available in KBS toolkits.

In the next chapter, the Prototype Knowledge Based Decision Support System is described from two main viewpoints: the factors involved in actually developing it, and how it provides decision support for the IT risk manager.

Chapter 6

DESCRIPTION OF THE PROTOTYPE KNOWLEDGE BASED DECISION SUPPORT SYSTEM (KBDSS)

6.1 Introduction

This chapter describes the Prototype Knowledge Based Decision Support System which is the tangible product of this research. The chapter is written in three parts. In the first part, the selected domain - the risk of flood from external sources - is considered. Most of the illustrations are quotations from available literature on the subject of the risks, including flooding, to which IT installations are potentially exposed. The second part of this chapter takes a detailed look at the US Flood-Proofing Regulations, and shows how this document's contents were used, and supplemented, in the development of the prototype KBDSS. The third part provides an overview of the prototype KBDSS which includes a description of the system from a user point of view and explanation of how the system performs the inferencing and analysis tasks with examples of the rules and other coded content of the system.

6.2 The Risk of Flood from External Sources

This topic was selected as the domain for three principal reasons.

First, the risk of external flooding is one which has been shown to warrant serious consideration from IT managers. In 1974, the US National Bureau of Standards (NBS) reported as follows in FIPS PUB 31.

> "Tropical storm Agnes, which swept through Pennsylvania in June 1972, caused severe flooding. Newspaper accounts reported that hundreds of computer systems were submerged in mud and water. The resulting damage appeared to depend largely upon location, and the reported time to recover ranged from two days to two months. The Pennsylvania Bureau of Management Information Systems reported its large computer under six feet of water. The entire reserve supply of certain forms used weekly, 45 million in all, was lost by another computer facility, leaving only a one-week supply in hand. A number of computer centres lost data files which were not backed up.
>
> "This experience points up two things. First, if an IT facility is located in a basement in a low-lying area, disruptions from flooding are almost inevitable. Second, careful planning for back-up operation can greatly reduce the time required to restore normal operations after an emergency."

In chapter 1 of this thesis, further examples of the costly consequences of flooding have been quoted, mainly from the insurance industry. It is clear, then, that there is already undeniable evidence that the risk of external flooding is one which IT managers should take as seriously as, for example, fire or unauthorised intrusion. The implications of an external flood can be substantial in terms of direct financial costs, and of loss of service. This research discovered no evidence that the

domain receives that level of attention. It is, however, a matter of historical record that flood is the next major cause of disasters after fire (see Figure 2.1).

The second principal reason for selecting the risk of external flooding as the domain was the availability of authoritative, documented specialist knowledge in the areas of risk identification and effective flood prevention measures. The publication "Flood-Proofing Regulations" (Dept of the Army, US, 1972) has been described by the NBS in FIPS PUB 31 as "... excellent guidance ... in the form of a model building code ... for minimising flood-related hazards of building occupancy and for protecting structures against flood damage." This availability of authoritative documented expertise ensured that the knowledge encapsulated in the prototype system could be a basis for sound decision support to users.

In spite of the fact that this document had existed since 1972, this research found no evidence that its contents had been used in any comprehensive approach to risk management by IT managers.

An objective of this research, therefore, became the representation of this sound basic advice into a prototype knowledge based system which would form an effective decision support tool. To achieve that, the researcher had to add significant value to the contents of the original document. This required the risk-entity interactions inherent in the domain to be identified and analysed. For example, there are interactions between the elevation of assets and their location, to handle which the hazard exposure zoning method described in chapter 4 above was developed. The three-environment approach also described in chapter 4 allows, for example, the interactions and inter-dependencies between assets and existing counter-measures to be plotted. These methodological refinements made it possible to incorporate the handling of the risk-entity interactions within the

system's inferencing mechanism. How this was achieved in practice is described in the Overview of the Prototype System below.

It will be noted that the document quoted here (FIPS PUB 31) is more than 20 years old. Its underlying principles, which stem from a concern to ensure the continuing availability of critical IT services, remain valid. But FIPS PUB 31 refers to back-up, i.e. steps to enable recovery from a disaster occurrence after the event. The most which such steps can achieve is the minimisation of the delay between the interruption of a service, and its resumption. However, the document also refers to the Flood-Proofing Regulations, and by definition "proofing" is aimed at protection from, or prevention of, a threat. This brings us to the central theme of this thesis: that disaster prevention measures will ensure that an IT installation's continued operation is safe-guarded against interruption by a flood occurrence.

The final principal reason for selecting external flood as the threat domain was the pervasiveness of the threat when it occurs. This imposed on the research a requirement for the rigorous treatment of the subject if a successful outcome was to be achieved. More detailed examples and explanations were given in chapter 4 above, but essentially by nature a flood will affect a wide but definable area. The critical arbiter of whether assets will be directly affected is their elevation in relation to expected flood water levels. Physical counter-measures may exist or be introduced to provide protection from a direct threat upon an asset. The determination of risk exposures involves considering risk entities, including counter-measures, assets and their locations, and their interactions and inter-dependencies, as described in chapter 4. There are also potential indirect effects to consider. For example, an IT installation may itself be sufficiently elevated to be proof against damage by flooding, but its ability to continue operations will be

compromised if there is no stand-by electrical power supply, or if staff are denied access.

Recognition of these potential interactions and inter-dependencies demanded the development of a solution which incorporated means of allowing them to be handled. These approaches enable the systematic and thorough analysis of the external flood threat, its direct and indirect effects, and appropriate counter-measures, as will be described in the Overview of the Prototype KBDSS below.

6.3 The US Flood Proofing Regulations

This document was published by the Office of the Chief of Engineers, US Department of the Army, Washington DC in June 1972. In its Preface, the following statements are included.

> "Existing building codes and regulations do not provide the special flood-proofing requirements and minimum standards of design and construction that should be met for buildings and structures susceptible to flood damage. A need for such standards has long been recognised at all levels of government and in the private sector. However, little, if any, work has been done to develop or assemble information on flood-proofing into a workable set of standards that could have national application.
>
> "This publication specifies the flood-proofing measures and techniques that should be followed to regulate private and public building construction in riverine flood hazard areas.
>
> "We have taken the first step ..."

In preparation for developing the Prototype System, a lot of time was spent on searches and enquiries to, for example, UK Water Authorities. The objective was to find expertly derived material to encapsulate in the Prototype System's knowledge base. However, no other material was found which offered the thorough treatment of the subject which these Regulations contain. Some examples below will illustrate this point.

As has been stated, the key value of the Regulations is that they provided the basic fund of "knowledge" which needed to be encapsulated in the Prototype KBDSS. This material not only enabled the system to have an authoritative knowledge base, but also allowed the accuracy of its inferencing mechanism to be validated.

That said, this material had to be supplemented during the research. Two brief examples, which are elaborated elsewhere in this thesis, illustrate this need for "added value". First, the Regulations themselves were primarily written for use by architects, civil engineers etc. The aim of this research was to produce a tool which would be "user-friendly" to IT managers. Thus, the technical content of the Regulations had to be re-presented in suitable terms. Second, although an engineer's knowledge would enable him intuitively to relate the Regulations to individual cases, the Prototype System needed to assist the IT manager by ensuring that, as each consultation proceeds, he or she is only presented with questions and conclusions which are relevant to the situation he or she is considering.

The following paragraphs draw upon parts of the Regulations, to show how they were used and built upon in creating the Prototype.

6.3.1 Use of the US Flood-Proofing Regulations in the Prototype KBDSS

The Regulations present and explain some practical aspects of flood-proofing. The following are some examples of:-

relevant extracts from the Regulations; and

how those extracts were built upon, or simplified, for use in the Prototype System.

Flood Hazard areas are classified by the Regulations as

Primary, which is land which would be covered by water during a flood which is representative of large floods which have occurred, or can be expected, in the area; or

Secondary, which is land outside the Primary Flood Hazard Area, which could nonetheless be affected by higher floods.

Any other land can be expected to be flood-free.

Because any IT installation is likely to be dependent upon assets, such as power and staff access, from outside the computer room, it was found necessary to develop a method for enabling the impact of such dependencies to be assessed. For example, electrical power is needed, and in the absence of a stand-by generator continued operation is at risk if any part of the supply route passes through a flood hazard area. Further, staff are needed to operate the installation, and if no safe refuge is available, or access to the building is exposed to the flood risk, then continuity of operation may also be at risk. In order to

a) simplify the handling of these considerations within the Prototype System;

b) permit a necessary finer distinction of risk exposure than only two area classifications can allow; and

c) enable some inferencing which will be described,

a hazard exposure zoning method was developed. It functions as follows.

Zones 1 and 2 are within the Regulations' definition of a Primary Flood Hazard Area. Two zones are used in the Prototype System, to distinguish between low lying areas (Zone 1) and

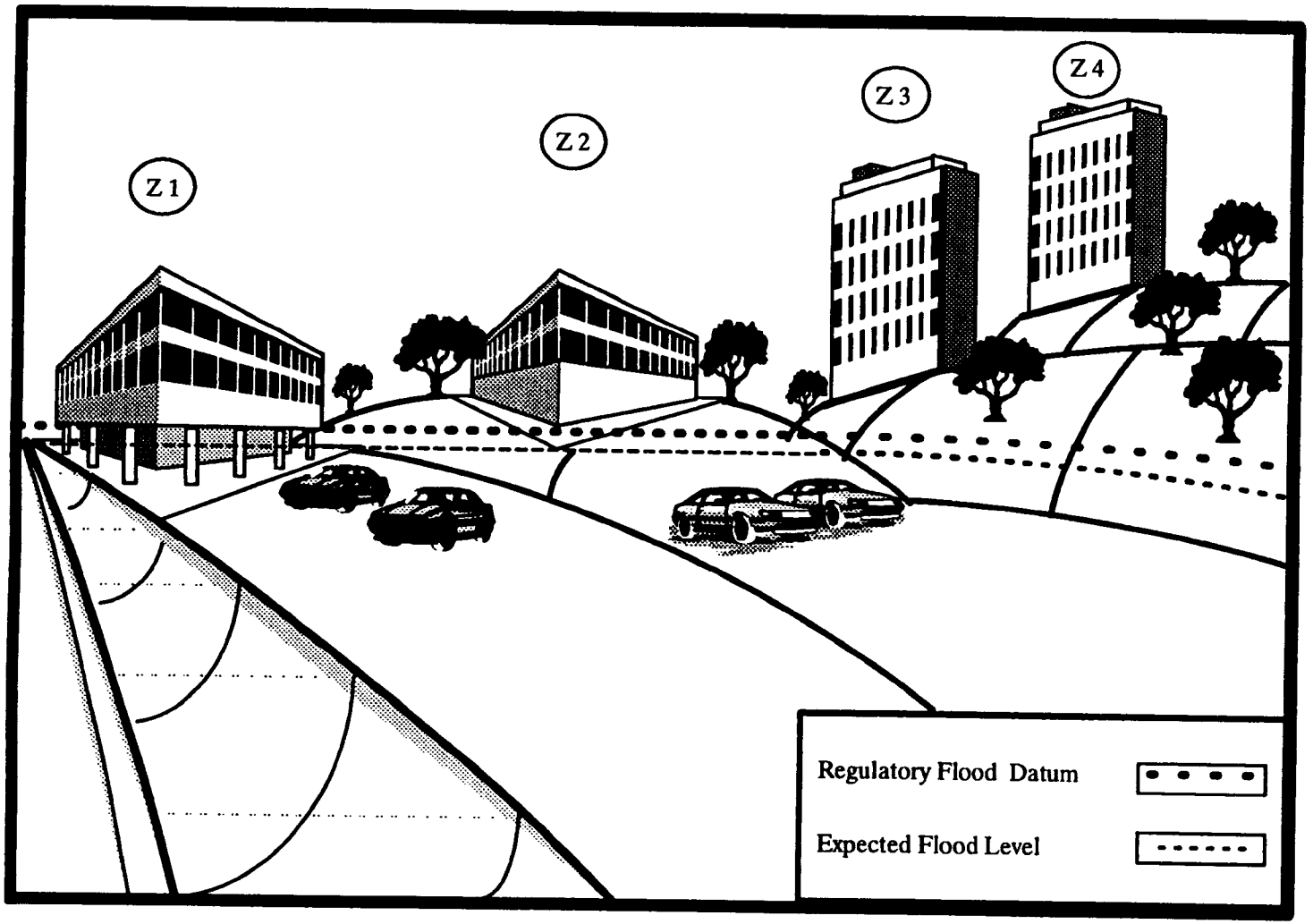
those which are naturally elevated by undisturbed terrain and the terrain is entirely in the flood plain (Zone 2).

Zone 3 is used to describe an area at the runout line of the flood plain naturally elevated above sea level by a natural and undisturbed terrain, and the terrain is partially in the flood plain.

Zone 4 is clear of any expected flood level, and is above sea level (see figure 6.1).

The Zoning approach was used to enable the following inferences to be made by the Prototype System, where appropriate.

Figure 6.1 – River Flood Hazard Exposure Zones



For Zone 1 the system takes all assets to be at risk, if there are no existing counter-measures. The basis of this inference is the Regulations' assertion that no buildings should be constructed in such a location, unless counter-measures (see below) are provided.

For Zone 2 outside assets, access and power are taken to be exposed if there are no counter-measures. The Regulations state that these assets should be specifically protected.

For Zone 3, only electrical power is considered potentially at risk.

For Zone 4, no assets are assumed at risk.

Four main classes of flood counter-measures for the full protection of buildings are described in the Regulations, namely

i) building on natural terrain above the Regulatory Flood Datum (RFD) which is the height above sea level up to which flood - proofing measures must be provided;

ii) building on fill, where elevation is achieved by the use of approved materials;

iii) elevation of the building by the use of stilts; and

iv) the use of dykes or flood walls.

Further counter-measures are suggested for the protection of assets, including the fabric of the building, in situations where the above counter-measures are not provided or would not protect all assets. For example, elevating a building does not in itself guarantee access or an uninterrupted power supply. The following are some examples of the specific counter-measures which the Regulations advocate.

If the building is located in a Primary or Secondary Flood Hazard Area, elevation of it by earth-fill or stilts provides protection. Barriers, which may be flood walls, dykes or levees, are also approved counter-measures.

The Regulations use a Building Type classification to denote the degree of protection afforded by the building's type of construction.

A Type A construction is completely impermeable to flood water and water vapour.

A Type B construction is substantially impermeable to flood water, but may admit flood water and water seepage.

A Type C construction is not water-proofed against flood water, and is taken by the system to be unsuitable for IT accommodation.

Yet further types of control may provide protection within a building whose location or construction otherwise provides less than ideal flood protection. Examples of such controls are siting IT facilities within a building, above any expected flood level; providing a personnel refuge above that level; having a stand-by generator above the flood level; and ensuring that there is an access road above that level.

The Prototype System's knowledge base contains these measures, and, from the zoning classification, the topographical and other physical factors which determine their suitability in particular cases. The system is thus able to make recommendations which take account of the degree of threat to which assets are potentially exposed, the extent of protection which they already have, and authoritative advice on which counter-measures are appropriate for the amount of "net" risk exposure remaining.

As a further necessary refinement, to enable the system to take full account of the complex interactions and inter-dependencies between assets, threats and counter-measures, the three-environment approach described in detail in chapter 4 was also used. In brief, the environments are defined as:-

controlled, or the computer room;

semi-controlled, or the building housing the installation; and

uncontrolled, or the surrounding area.

Given data on these "siting" factors, as well as on asset, threat and counter-measure interactions and inter-dependencies, the system is able to infer and recommend appropriate risk management actions as will be seen later.

Table 6.1 below illustrates the above points in regard to the selection and effectiveness of individual flood-proofing prevention measures. The protection given should only be read off for individual counter-measures. It uses a simple Yes/No annotation to show where risk exposure of assets does or does not exist, by relating the assets' and counter-measures' locations to the flood Hazard Exposure Zones and to the "three environments". By reading across, it shows which assets are, or are not, protected from the threat of flood by individual counter-measures. Examples of the use of the Table are given below. This table is represented in the Prototype Knowledge Based Decision Support System's rule-base, so that the user is presented with the results of the provision of any one counter-measure, or combination thereof.

In the table, assets can be "read off" as follows:-

a) outside assets located in the uncontrolled environment, such as vehicles, external storage (*);

b) the building itself, the outer walls of which are the boundary of the semi-controlled environment (*†);

c) the computer room, which is the controlled environment, including all of its contents (*†);

d) personnel, located in the controlled environment, and on whom the IT service is dependent (*†);

e) access, to permit the above staff to enter their place of work (†); and

f) electrical power, on which the continuity of IT service is also dependent (†).

Notes:

Classes of risk exposure are denoted above as:-

* indicates where physical destruction of assets is a risk or any additional operating costs that may result (e.g. in the case of personnel loss).

† indicates that denial of service would result.

Table 6.1 lists the counter-measures taken from the Regulations, and shows to which assets those counter-measures will provide effective protection. This takes account of the degrees to which the individual assets are exposed to the threat of flood, which in turn is influenced by the data on the zone and environment in which they are located.

Table 6.1 - Protection provided by Disaster Prevention Measures

Environment	Disaster Prevention Measures	Control in Place	Assets Exposed if there are no Countermeasures															
			in Zone 1						in Zone 2			in Zone 3						
			Outside	Bldgs.	Comp. Room	Pers onnel	Access	Power	Outside	Access	Power	Outside	Bldgs.	Comp. Room	Pers onnel	Access	Power
Uncontrolled Environment	Flood Barriers	Y	N	N	Y	Y	N	Y	N	N	Y						
	Building Elevation	Y	Y	N	N	Y	Y	Y									
	Outside Asset Elevation	Y	N	Y	Y	Y	Y	Y	N	Y	Y						
Semi-Controlled Environment	Type A Building Const'n	Y	Y	N	Y	Y	Y	Y									
	Type B Building Const'n with AFS*	Y	Y	N	Y	Y	Y	Y									
	Type C Building Const'n	Y	Y	Y	Y	Y	Y	Y									
Controlled Environment	Comp. Room Elevation	Y	Y	Y	Y	Y	Y	Y									
	Personnel Safe Refuge	Y	Y	Y	Y	N	Y	Y									
	Access to Building	Y	Y	Y	Y	Y	Y	Y	Y	N	Y						
	Stand-by Generator	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N

*Automatic Flooding System

The table can be used to "read off" the effects of risk management decisions, as the following further examples illustrate.

Example 1.

As noted above, if the site is in Zone 1, and there is only a flood barrier in the uncontrolled environment, the assets at risk from flooding are the computer room, personnel and power. This is because the barrier provides protection for the outside assets, the building and the access. If the site were in Zone 2, with the same counter-measure, the only asset exposed to flood risk becomes power. This is because the computer room and personnel are protected by the higher elevation. If the site were in Zone 3, only a stand-by generator is required to ensure that no assets are at risk, because the yet higher elevation ensures protection of all other assets including access.

Example 2.

If the site is in Zone 1, and the elevation of the building in the uncontrolled environment is the only counter-measure, the assets at risk from flooding are the outside assets, personnel, access and power. This is because the elevation of the building provides protection for the building and the computer room. If the site is in Zone 2, the only counter-measures required are those which can provide protection to outside assets, power and access, because the building and its contents are already elevated above an expected flood level.

The above examples relate only to counter-measures provided in the uncontrolled environment. Let us now consider examples where counter-measures in the other two environments are included.

Example 3.

If the site is in Zone 1, and Type A (or B with an Automatic Flooding System) building construction is provided in the semi-controlled environment, the only protection is to the building itself: all other assets are at risk unless additional counter- measures are provided. The semi-controlled environment counter-measures are irrelevant to sites in Zones 2 and 3, because those latter sites will always be above the water level.

Example 4.

For sites in Zone 1, consideration of counter-measures in the controlled environment is worthless unless effective controls have been installed in the uncontrolled environment, because only the IT centre staff can be protected by action exclusively in the controlled environment (i.e. by the provision of a safe refuge).

Table 6.1, therefore, is an illustrative aid to modelling these risks. The Prototype Knowledge Based Decision Support System itself enables the consideration of the kinds of "compound" scenario inherent in addressing risk exposure. As a result, an IT manager is supported in making decisions on risk management measures.

6.3.2 Conclusion - US Flood-Proofing Regulations

As has been illustrated, the Flood-Proofing Regulations have provided an authoritative basis for building the Prototype System, but considerable additional development has been needed to convert that basis into a fully working Disaster Prevention Decision Support System.

6.4 Overview of the Prototype KBDSS

KBS technology is capable of handling the expert heuristics, relations and "rules" which need to be considered in Risk Management. For example,

a) it allows the representation of the knowledge of domain experts (e.g. of the influences and interactions and inter-dependencies among asset locations, threat types and counter-measure effectiveness through their identification in the three environments and hazard exposure zones) to identify risk exposures;

b) given this information on potential risk exposures, it goes on to assess the associated potential loss, using additional input data as required for the calculation; and

c) uses the resulting potential loss value as one factor, alongside others such as cost-effectiveness, in advising on the selection of appropriate counter-measures.

The Prototype Knowledge Based Decision Support System makes extensive use of the three-environment approach and hazard exposure zoning method already described in chapter 4, and addresses the threat of Flood. The Prototype System demonstrates how an IT Risk Manager can be provided with a Disaster Prevention Decision Support tool, as part of an overall Disaster Prevention policy.

A key objective of any Prototype Knowledge Based System is to prove that the technology is capable of delivering the required solution. Unlike a full production system, a prototype does not set out to

a) address more than a sub-set of the problem domain, or

b) provide ideal facilities for Human Computer Interfaces, data storage or output reporting.

Clearly in a full production system, such omissions would be unacceptable. For a prototype, on the other hand, simply to be able to answer the question "Is this technology suitable?" is sufficient. It is also sufficient to demonstrate the feasibility of the type of solution recommended in this project.

However, since the result of this research has been to prove that Knowledge Based System technology is suitable for use within an overall Computer Disaster Prevention policy, it may be helpful to note that in a full production system, the following additional facilities would be provided.

a) Coverage of all threats, including their equally thorough analysis, their counter-measures, etc.

b) The comprehensive recording of all assets.

c) More refined Human Computer Interfaces and output reporting.

It is clear from its name that a Knowledge Based System holds and processes Knowledge, in other words a "special" form of data which is the information, expertise and heuristics which a specialist would apply to a problem in his domain in order to solve it. In the prototype, questions elicit information on the extent of previously recorded flooding at a location, on the availability of relevant counter-measures and their effectiveness, and on the interactions and inter-dependencies of threats and counter-measures. Some of this information may be "pre-loaded" into a Knowledge Based System before the system is given to a user, or the user may be required to provide it after consultation with meteorologists, water authorities,

architects etc. The types of domain knowledge which need to be held and processed by the prototype KBDSS are shown in figure 6.2. These types of domain knowledge are encapsulated into the knowledge base, and processed using frames and production rules. The arrows show how the prototype KBDSS processes its knowledge following the same logical sequence of actions required for risk management process. This leads to the solution of selecting and implementing cost-effective flood prevention measures

How the system actually works can perhaps best be described by explaining its use step by step. By further describing how the system processes knowledge and data, the reader is given a clear picture of both the structure and working of the Prototype KBDSS. The content of each screen is described, to aid the reader in visualising the system in actual use. Typical examples of the Input / Output screens are also represented in Appendix B. The Prototype KBDSS, which is screen driven, consists of two main modules, namely:-

a) Knowledge Acquisition and Data Collection; and

b) Inference Mechanism, Analysis and Results.

6.4.1 Knowledge Acquisition and Data Collection

The system prompts the user by asking questions. The first question asks "In which zone is the site?" The user is asked to select Zone 1, 2, 3 or 4. Definitions of each type of zone are in Table 6.2 below.

Figure 6.2 - The Types of Domain Knowledge Held and Processed by the Prototype KBDSS

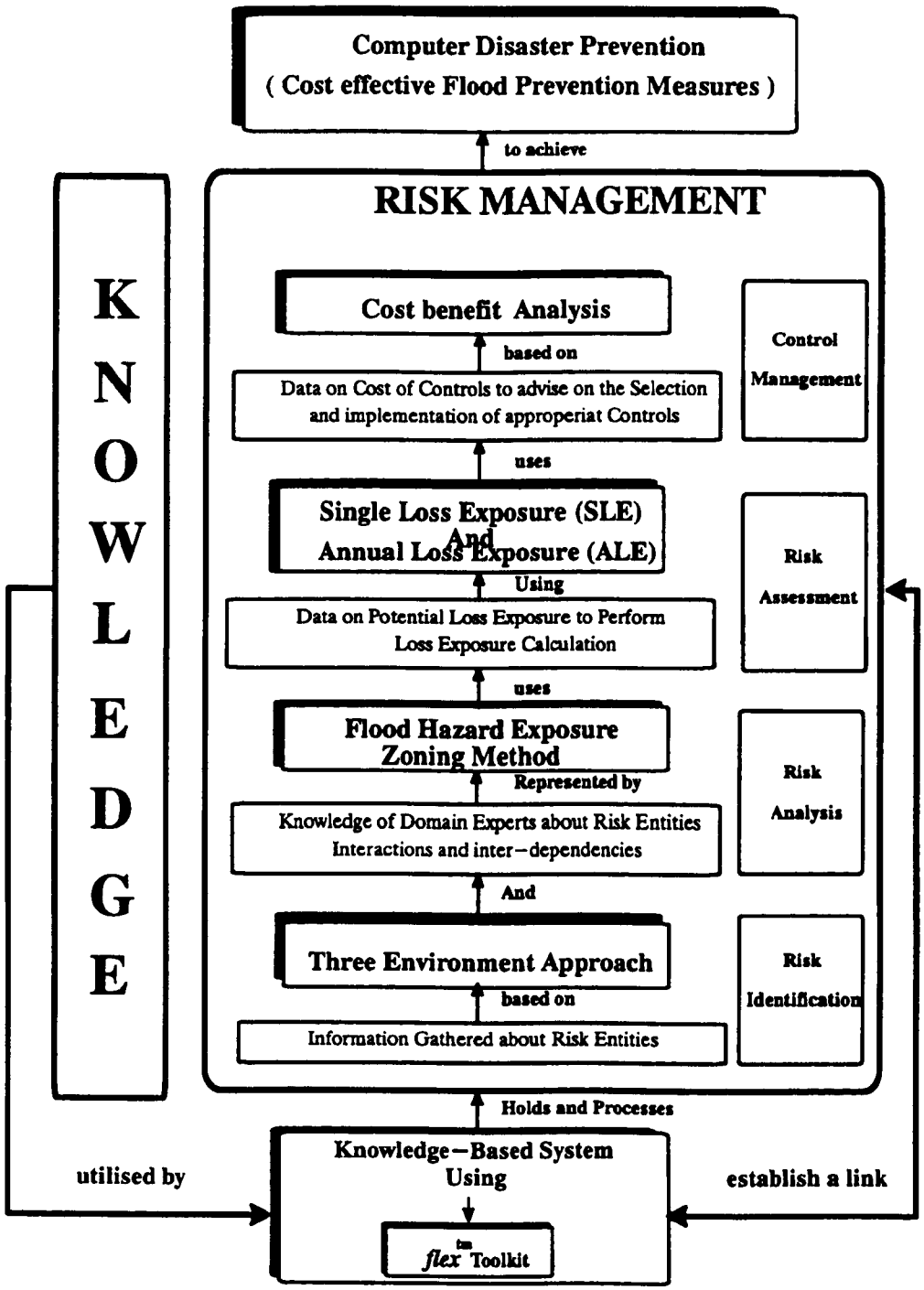


Table 6.2 - Hazard Exposure Zones

Zone 1	An area in the floodway of the flood plain adjoining a river and below sea level. (The system associates outside assets, access and power; and the building itself, personnel and the computer installation as being exposed to threat of flood, if there are no existing counter-measures.)
Zone 2	An area in the floodway fringe of the flood plain, naturally elevated above sea level by natural undisturbed terrain, and the terrain is entirely in the flood plain. (The system associates outside assets, access and power as being exposed to threat of flood, if there are no existing counter-measures.)
Zone 3	An area at the runout line of the flood plain naturally elevated above sea level by a natural and undisturbed terrain, and the terrain is partially in the flood plain. (The system associates power as being exposed to threat of flood, if there are no existing counter-measures.)
Zone 4	An area beyond the runout line of the flood plain and above sea level. (The system associates no assets as being exposed to threat of flood.)

This form of hazard exposure zoning is used for ease and unambiguity of reference, and in the Prototype System is based on the U S Flood - Proofing Regulations. Clearly, in a production system, locally available standards would be used.

Further questions are only asked if the site is in Zone 1, 2 or 3, because the "knowledge" that a site in Zone 4 is above any expected flood level has been

recognised by the inferencing mechanism. (If the user "enters" Zone 4, the system reports that no losses will arise, and asks if a further consultation is required.) Similarly, the questions that are asked as the user proceeds are also selected by the system's inferencing mechanism, so that users are only required to answer questions relevant to their location. The subject of interactions and inter-dependencies among environments, threats, counter-measures and assets has already been discussed fully in chapters 3 and 4.

Specifically, if the site is in Zone 1, 2 or 3, questions will next be asked about what relevant counter-measures are already installed in any of the three environments.

Subsequent input screens may ask about some of the following, for example.

a) Controls which exist in the uncontrolled environment, namely whether the site is elevated or there is a barrier.

e.g. i) elevation may be by earth fill or stilts; and

ii) barriers may be flood walls, dykes or levées.

It may also be relevant to ask what other controls exist in the uncontrolled environment, such as the elevation of outside assets which may be vulnerable to damage. Even though that need not lead to loss of service, information on all loss exposures is needed for an overall loss exposure assessment. Answers input allow the inferencing mechanism to process this "knowledge" later, in determining what further questions to ask, and what recommendations to make at the end of the consultation.

b) Controls which exist in the semi-controlled environment, namely the type of building construction, and whether there is an automatic flooding system*.

Table 6.3 - Building Types

A.	Completely impermeable to flood water and water vapour.
B.	Substantially impermeable to flood water, but may admit flood water and
C.	Not waterproofed against flood water. Not suitable for IT accommodation.

* This is a protective measure. By deliberately flooding the interior of a building to the same level as the water outside, exterior and interior pressures on the structure can be balanced.

c) Controls which are relevant to the controlled environment, namely

Is the Computer Room elevated above any expected flood level?

Is there a personnel refuge above any expected flood level?

Is there a standby generator above any expected flood level?

Is there an access road above any expected flood level?

The purposes of these controls are respectively to ensure that

the computer service is not lost because the equipment itself is in an area which is liable to flooding;

staff have a safe place to occupy during a flood emergency;

service can be continued even if a flood disrupts the main electrical supply; and

staff and other necessary access is available even during a flood.

From answers given by the user up to this point, the system is able to ascertain and report the site's and its assets' exposure to threat, after taking account of any counter-measures which are already in place. What is involved here, within the inferencing mechanism, is the consideration of the data already input, and what it means for the actual vulnerability of the service provided. The site's hazard exposure zoning, the presence or absence of counter-measures in all three environments, the computer room's location etc. all need to be considered in relation to each other. The process is similar to modelling. The following table illustrates some of the interactions and inter-dependencies which are possible for sites by type of zone, if there are no counter-measures in place.

Table 6.4 - Asset Exposures

Assets Exposed	Zone 1	Zone 2	Zone 3
The Building	+		
Outside Assets	+	+	
Computer Room	+		
Personnel	+		
Electrical Power	+	+	+
Access	+	+	

+ indicates which asset groups are at risk if there are no counter-measures in place. (Nothing is at risk of external flood if the site is in Zone 4.)

Of course, the Prototype Knowledge Based Decision Support System will also consider data on the presence of counter-measures, to refine its reporting of actual exposures (see table 6.1).

A few further input screens then ask for data on the replacement cost and time for those groups of assets which the system has identified as exposed to the threat of flood, and on the average daily loss from the non-availability of those assets. After processing this data, the system reports:-

a) the longest down-time given;

b) the revenue loss from denial of those assets for that period; and

c) the aggregate replacement cost.

The system then asks for the expected annual frequency of a flood occurring. This data is used to convert the one-time values for down-time, revenue loss and replacement costs to annual values.

Further questions elicit the costs of providing disaster prevention measures which will give protection to those asset groups which have been identified as being at risk. Yet further questions will be asked, depending upon what risk exposures have been identified by the system, as shown in table 6.5 below.

Table 6.5 - Further Counter-measure Costs

Type of Risk Exposure	Questions on Costs of:-
Destruction with Denial of Service	Insurance against loss of service, against damage; Disaster Recovery Service
Destruction with no Denial of Service	Insurance against damage
Denial of Service only	Insurance against loss of service; Disaster Recovery Service

The costs of providing counter-measures are then compared with each other, and with the potential costs or losses due to their absence. The system then recommends the provision of cost-effective counter-measures. For example, if the cost of providing a counter-measure is greater than the loss without it, the counter-measure will not be recommended. If more than one counter-measure would provide the required level of protection, the system reports the choices available, and selects and recommends the least expensive. Insurance or disaster recovery measures may only be recommended if preventive counter-measures cannot be cost-justified. Examples of the kinds of results which cost-benefit analysis may produce are given later in this chapter.

6.4.2 Inference Mechanism and Analysis

In the above paragraphs, the reader has seen an explanation of how the Prototype Knowledge Based Decision Support System works, but mainly from the viewpoint of what the user is required to input, and what he or she will see as results. For the purposes there, any description of what is actually happening within the system was necessarily brief. In the following paragraphs is a more detailed explanation of how the system performs the inferencing and analysis tasks.

The reasons for selecting a KBS solution and ***flex***^{tm} toolkit have already been described in chapter 5. How this technology has enabled this research to address and solve the problems of developing a Prototype Knowledge Based Decision Support System for Computer Disaster Prevention can be explained by looking more deeply at the key facilities the toolkit provides for Knowledge Representation and Inferencing. As an aid to understanding, table 6.6 below is a glossary of LPA

flex^{tm} of the terms which are more frequently used (for more definitions see Appendix C)

Table 6.6 - Terms of LPA *flex*^{tm}

Term	Explanation
action	Name of defined goal. Contains 1 test and 1 definition.
constraint*	A procedure which is triggered for validity checks on updates.
demon*	A procedure which is triggered after specified slot values are set.
frame	Used to hold data. Is composed of attributes, which correspond to data fields.
group	Contains the names of the rules to be considered.
inheritance	The means by which slot values may be acquired from parent frames
launch*	A procedure which is triggered at the creation of new instances.
question	Acquires data from user.
relation	Name of defined goal. Can contain tests and have multiple definitions.
rule	Forward chaining production rule, infers conclusions from data.
ruleset	Forward chaining control structure.
slot	Data is held in slots which have values corresponding to the data.
synonym*	Allows use of text for specifically defined values.
template	Allows program to be more like natural language.
watchdog*	A procedure which is triggered before slot values are accessed.

* These facilities did not need to be used in the Prototype System.

To represent the knowledge which drives the Decision Support process, use has been made of Frames and Relations. The part of the Inferencing Mechanism which produces Disaster Prevention recommendations is rule-based. A few examples of the Frames, Relations and Rules actually used in the Prototype Knowledge Based Decision Support System will show how these facilities operate together. These examples will also show how the interactions and inter-dependencies between

threats, locations, counter-measures and assets described conceptually in chapters 3 and 4 are handled in the actual development of the prototype solution.

In the description immediately above of how the system interacts with the user, it was noted that the user who has input his location as zone 4 is asked no further questions, because the system is able to infer that all of his assets are above any expected flood level. This example characterises the Prototype Knowledge Based Decision Support System, in that all subsequent questions are selected by the inferencing mechanism, which takes account of previous answers when selecting next questions. For example, the system only asks for data on the value, cost and replacement time for assets which it has identified as exposed to the hazard. Further, it only asks for costs of counter-measures if they are relevant to the type of exposure, and if they are not already provided (see table 6.5).

6.4.2.1 Frames

In a Knowledge Based System, a frame is used to represent an object, and frames can include slots to represent the attributes of that object. A slot can refer to another frame or object, and thus provide for the inheritance of attributes from one object to others in an hierarchical fashion. It is also possible, within a rulebase, to "block" inheritance where knowledge or data indicates that that should be done. In the Prototype System, for example, the vulnerabilities of any site shown to be in zone 1 will be inherited as slot values from a model zone 1 site. The relevant counter-measures for all zone 1 sites are the same. But if one or more relevant counter-measure is shown to exist already, then the inheritance of default vulnerabilities and of appropriate new counter-measures is modified. At set-up

time, it is often appropriate to give slots zero default values, which are modified at run time by answers to questions about individual objects.

Example A

```
frame site

default losses is nil

and default delays is zero {0}

and default total_cost is 0.
```

At setup each is given a slot value of nil for potential losses (default losses is nil), for time required to restore normal level of service (default delays is zero {0}) and for costs of counter-measures (default total_cost is 0). If the site is in any zone and has all relevant counter-measures in place, these values remain unaltered. Otherwise the user is asked to input a slot value (i.e. price) for each relevant new counter-measure the sum of which is written into the slot total_cost.

The "firing" of Rules (see later) selects relevant new counter-measures as questions on hazard exposure zoning, existing counter-measures, revenue losses and asset costs are answered.

Example B

```
frame building

default flood_system is nil.
```

Again, at setup time a building is assumed to have no automatic flooding system (flood_system), and this assumption "holds" for sites in zones 2, 3 and 4. The slot

value is modified for sites in zone 1 if the user indicates that an automatic flooding system is in place.

Example C

```
frame uncon_envir

default controls are nil.
```

Similarly, the assumption of no counter-measures existing (default controls are nil) in the uncontrolled environment (uncon_envir) is of no concern if the site is zoned 3 or 4, and the slot value may be modified for a site in zone 1 or 2 depending on users' answers to questions.

6.4.2.2 Rules and Inferencing

The LPA *flex*^{tm} toolkit used to develop the Prototype System allows the forward- and backward-chaining of Rules. Forward chaining allows the inferencing to be data driven - for example, what action to recommend is indicated by analysis of the data input. In the Prototype System, a recommendation to install a barrier will be made if consideration of data on location, existing counter-measures, losses and costs show that to be appropriate.

Backward chaining allows the inferencing to be goal driven. In the Prototype Knowledge Based Decision Support System, the goal is to ensure that sites have adequate protection against threats. When answers to questions, represented as slot values in Frames, indicate a level of risk exposure, the user is made to consider the installation of appropriate counter-measures.

The key function of the Rules, and of the Rulebase which they collectively become, is to represent the domain expert's heuristics. At first they may appear too simplistic in structure - i.e. if .... and.... and .... then .... - to achieve this satisfactorily. But by also using Frames, Relations, Templates, Groups and Actions, a Knowledge Based System can not only encapsulate the heuristics used by a domain expert, but also use those heuristics to drive an inferencing process which produces reasoned conclusions. Part of that reasoning process, in the Prototype Knowledge Based Decision Support System, takes account of the interactions and inter-dependencies between locations, existing counter-measures, losses and costs discussed earlier.

It has become clear in this research that a rule-based approach to the problem of how to ensure adequate protection against threats has not previously been used to address the full spectrum of Disaster Prevention issues. It is successful because of two features of Knowledge Based System technology. First, the technology allows expert heuristics to be represented and used comprehensively and economically. Second, it imposes a disciplined, structured approach on the user. This ensures that important considerations are not overlooked. This combination of thoroughness and the application of expert knowledge means that the users can have a high level of confidence in the recommendations that they are given to consider. That this much has actually been achieved at a practical level is a clear indication that the research has not only been into previously unexplored possibilities, but that it has also been accurately targeted and its results correctly interpreted. The research indicated, and the results proved, the value of Knowledge Based System technology in solving Computer Disaster Prevention problems, which is an area of key concern to IT Risk Managers. It was clear from the literature review that some areas of risk exposure and prevention have already been researched. It was also clear that some narrower research had been

addressed at the use of Knowledge Based Systems in Risk Management. What this study has done is to show how all aspects of risk exposure can be addressed. It also shows how Knowledge Based System technology is capable of accommodating the heuristics needed to cover the full spectrum of issues involved in Computer Disaster Prevention.

Here are a few examples of rules.

a) rule zone1_1

if the location of the site is zone1

and there are no controls

then site's losses become {building and computer_room and outside and power and access} .

This rule asserts that if a building is in zone 1 (see figure 6.1 and Table 6.2 above), and there are no counter-measures, then the building and its equipment and personnel; its external assets; its electrical power supply and access route are all potential losses. The system would then go on to ascertain what counter-measures are appropriate for protecting against those losses. This is a simple illustration of how the system infers, from the interaction of information on the site's location, and on the absence of counter-measures, what potential losses arise. It is a "worst-case" scenario, and shows how a Knowledge Based System can accommodate the kind of scenario-based analysis propounded by some previous researchers.

b) rule zone1_3

if the location of the site is zone1

and the type of building is 'A'

then the site's losses become {outside and power and access} .

Here the system looks to see if the building has been declared to be type "A" (see figure 6.1 and table 6.2). If it has, the system asserts that the potential losses are outside assets, power and access. In other words, because a "Type A" building is known to have good protection against structural damage from flooding, the system can infer that the building itself and assets in the semi- and controlled environments are adequately protected against flood damage. Nonetheless, a further rule checks whether the computer installation itself is located above an expected flood level. If it is not then elevating the computer installation above an expected flood level is evaluated as a potential counter measure.

c) rule zone2_1

if the location of the site is in zone2

and the uncon_envir has a barrier

then the site's losses become {power} .

In this case, the rule asserts that the potential loss is only power, because there is a barrier in the uncontrolled environment. In other words the system infers

i) from the location in zone 2 (see figure 6.1 and table 6.2) that the building and its contents are naturally elevated above an expected flood level; and

ii) that the existence of a barrier protects the building's outside assets, including access.

Additionally, because of the elevation implied by the zone 2 categorisation, the system suppresses any questions about counter-measures in the semi- controlled environment, since that environment is adequately protected.

d) rule zone2_2

if the location of the site is in zone2

and the uncon_envir does not have a barrier

then the site's losses become {outside and power and access) .

Here there is no barrier therefore, the rule asserts that the potential losses are to outside assets, power and access. From the zoning (see figure 6.1 and table 6.2) the system can automatically infer only the building itself is elevated, and that therefore its outside assets are exposed to threat.

e) rule zone3_1

if the location of site is in zone3

then the site's losses become {power} .

Here, the system automatically asserts that only the power supply is potentially threatened, because it is known from the zoning that all other assets are sufficiently elevated above an expected flood level. It can be seen from figure 6.1 that not only the site itself, but also the surrounding area, are naturally elevated, protecting access and outside assets.

f) rule zone4_1

if the location of the site is zone4

then the site's losses become nothing .

In this case, the system asserts that there are no potential losses, again by inference from the hazard exposure zoning, this is not an insignificant conclusion. It should be pointed out that one aim of this study has been to permit the evaluation of potential computer sites, as well as of existing ones. In comparing sites available for development, the scenario may arise where a zone 4 site has a much higher purchase cost than in zone 1. When using this kind of system to evaluate the two sites comparatively, however the true cost (including the costs of providing required counter measures) of the zone 1 site is available. In this way, the actual and full costs of both can be compared on an equal basis. As a further point, the site in zone 4 is already well protected against flood, and one might argue that that fact is obvious without the detail of analysis undertaken. It should be remembered that the Prototype System only deals with flood risks, whereas a full production system would address all risks. It is quite possible that the zone 4 site, fully protected against flood is, however, very exposed to lighting, terrorism or other threats. Therefore, in a full system it is intended to assess each site's vulnerability to all risks, even if analysis showed it fully proofed against one of them.

g) rule z1refuge_1

if the site's location is in zone1

and the type of building is 'B'

and the building's flood_system is not effective

and the answer to site refuge is no

and the answer to uncontrolled_measure is nothing

then include personnel in the site's losses

and remove refuge from the sites controls .

At this point, the system considers the safety of personnel. If data indicates that

i) the building's location;

ii) the building's structure, which is not adequately flood-proofed;

iii) the absence of an effective automatic flooding system;

iv) the absence of a personnel refuge; and

v) the absence of other relevant counter-measures in the uncontrolled environment.

are a cause for concern, the system infers that the safety of staff is at risk and reports personnel as a potential loss among other associated losses such as the building and its contents. At a later stage, the system will assess the provision of a personnel refuge as a potential counter-measure.

This personnel issue is a particularly important one for thorough consideration, most obviously because of a proper concern for human safety. Additionally, any IT Risk Manager faced with a potential loss of personnel, will later in the system be made to consider the significant costs of replacing trained staff, and the delay in providing an effective computer service which such a loss will also impose. These factors are clearly very significant, as are the considerations for any employer's occupational health responsibilities.

It should be reported here that some fragmented work has been done on staff safety issues. There has, however, been no previous success in achieving the thorough assessment of personnel risks, and potential losses from them, and the identification of needed counter-measures.

The above paragraphs have set out only to illustrate some of the rules used in the Prototype System. Much of the program code for the whole system is reproduced in Appendix A.

It is now appropriate to explain how the system goes on, from having identified the site's risk exposures, to consider appropriate counter-measures. What are appropriate counter-measures can be identified using a Cost-Benefit Analysis approach, which is a way of evaluating what counter-measures are justified by

a) the level of threat exposure identified;

b) the loss exposure identified (i.e. replacement costs and revenue losses); and

c) the costs of appropriate counter-measures.

The system uses two sets of calculations to perform the assessment. Data for use in the calculations is taken from users' answers to questions on service values, and asset replacement times and costs. After the first calculation, the system reports the longest down-time; aggregate asset replacement costs, and revenue losses. At this stage, the report treats the losses as the result of a one-time occurrence.

Further calculation converts the one-time occurrence values to annual values, using data provided by the user on the expected frequency of flood occurrence. User-provided data on the annualised costs of appropriate counter-measures is then

considered, so that the best-value counter-measures can be selected and recommended. The following is an example of a rule used at this stage.

h) rule cb3

if the site's location is zone2

and the site's losses include {outside and power and access}

then new_slot (case, global, wcsz2)

and wcs1z2 becomes barrier_cost + power_cost

and the site's wcso1z2 becomes {barrier and power}

and wcs2z2 becomes outside_elevation_cost + power_cost + access_cost

and the site's wcso2z2 becomes {outside and power and access}.

Here the system asserts that data on the annualised costs of appropriate counter-measures should be provided by the user so that proper control justification can be achieved. In this particular case, the system suggests two control sets and selects the one that is cost-effective and provides the required level of investment (see table 6.7 below).

Table 6.7 - Control Set

Counter measures required for a site in zone 2	First control set	Assets protected after control implementation			Second control set	Assets protected after control implementation		
		outside assets	access	electrical power		outside assets	access	electrical power
Flood barriers	♦	♦	♦					
Outside assets elevation					♦	♦		
Access to building					♦		♦	
Standby generator	♦			♦	♦			♦

Although both control sets may provide adequate protection against an external flood risk, their annualised costs may not be the same. Further rules ensure that, where there is a choice of counter-measures of equal effect, the least expensive is recommended. Clearly, where the cost of losses is less than the cost of preventive-measures, then insurance and disaster recovery measures may be considered. The following section illustrates how the prototype KBDSS considers and justifies the costs of appropriate counter-measures.

6.5 Cost Benefit Analysis

If the system reported that the only exposed asset will be the computer room, due to its location below the flood level, the loss exposure would be the sum of the

asset replacement cost and the loss resulting from the denial of service, caused by the unavailability of the system.

If:-

a) the replacement cost of the damaged equipment is £1,000,000;

b) the average daily loss to the organisation from the denial of service is £30,000; and

c) the time required to restore a normal level of service (maximum down time) is 25 days,

then the Single Loss Exposure (SLE) is

£1,000,000 + (£30,000 x 25) = £1,750,000.

If the frequency of flood occurrence is 0.2, then the Annual Loss Exposure (ALE) is £1,750,000 x 0.2 = £350,000.

If we were to apply a counter-measure by raising the computer room above the expected flood level, we should not spend on that an annual amount which exceeds the ALE. This of course would protect the equipment from damage, and also reduce the downtime period to zero days, thus preventing the loss resulting from the denial of service. If, on the other hand, we wish to justify the cost of this preventive measure against the cost of insurance and disaster recovery measures, we need to compare all of these costs based on the following sets.

a) The annual cost of disaster recovery measures, plus the cost of the annual insurance premium against property damage, because disaster recovery measures alone can only account for service interruption losses.

b) The annual insurance premium against service interruption loss plus the annual insurance premium against property damage.

c) The annual cost of preventive counter-measures to provide protection against both property damage and service interruption.

The total costs for each of the above sets of measures should be compared with each other, to determine which is the most cost-effective set.

Using the same example to justify these costs, we assume that the annual cost of raising the computer room above the expected flood level has reached the cross-over point (see figure 3.3) at which the expected ALE is equal to the annualised cost of this control, which in this case is £350,000. We know that this counter-measure would provide protection against £1,000,000 as the cost of replacing the equipment, and also against a maximum service interruption loss of £750,000. If we assume that the annual cost of disaster recovery measures, which only provide protection against service interruption loss, is equal to the annual loss through service interruption, which in this case is £150,000, we would still not be covered for physical damage losses, and in this case we may need to consider insurance to deal with the latter losses. The annual insurance premium for physical damage may prove to be higher than the annualised replacement cost of £200,000, which was estimated for risk assessment purposes, because in practice insurers usually consider a loading factor to cover administration costs and a profit margin. Also, insurance underwriters may not fully compensate for the damaged equipment at 100% of its replacement cost. They may apply some form of deduction, so that the amount reimbursed is a depreciated value.

The conclusion is that, even with a combination of disaster recovery measures and insurance, the total annual cost may be higher than the expected ALE. Similarly, if

we combined the annual cost of damage insurance with the annual cost of interruption insurance, we may find that the organisation will be committed to a higher cost, because again the premium for service interruption cover may be higher than the estimated annual downtime loss.

The example above has shown a justification for preventive counter-measures when the annualised cost of raising the computer room above the expected flood level could only be as high as the cross-over point. If this cost exceeds the total ALE, it may be found that neither risk reduction measures nor the transfer of risk to insurance can be justified. In such a case, management may find it necessary to avoid the risk by moving or siting the entire installation in another location which is not exposed to the risk of flood. In some extreme cases it may be found difficult to avoid the risk by moving an existing installation to another location, especially when the cost is shown to be even more than the cost of just raising the computer room above the expected flood level. In this particular case, management may search for alternatives. One alternative is to consider dealing with the risk by accepting some part of the loss, either by transferring some of the risk (to insurance) or by implementing only disaster recovery measures. Another possible alternative is to implement disaster prevention measures, if their annual cost is slightly higher than the ALE, since the benefit may prove to be much higher than the combined benefits of disaster recovery measures and insurance. It is left to management to select the most appropriate action.

In the previous example, the risk was only concerned with one class of risk exposure, namely the destruction of the computer room with resulting denial of service. In other situations, the risk may include several classes of risk exposure. For example, with the loss of the computer room there may also be the loss of vulnerable outside assets, which may not cause denial of service, but which may be

costed and added to the computer room replacement cost. There may also be a denial of service loss resulting from the interruption of the electrical power supply due to flooding, and this interruption would cause a downtime period of five days. In such a case, the prototype system gives full consideration, in calculating the ALE, to each class of risk exposure and its associated loss. This is done by summing the annual replacement costs for all damage, and adding the result to the maximum annual down-time loss, to determine the total ALE.

In justifying the cost of counter-measures for several classes of risk exposure, a similar approach is followed to that used for a single class of risk exposure above. The total cost for each of prevention, recovery and insurance measures is calculated and presented, to be matched with the total ALE, for proper justification.

As a further explanation, using the same figures as for the single class of risk exposure above, we know that the annualised replacement cost from physical damage is £200,000 and that the annual service interruption loss is £150,000, and therefore that the ALE is £350,000. We also know that the investment for raising the computer room above the expected flood level should not exceed the ALE. In fact, this investment may only be considered cost-effective against this particular class of risk exposure alone, but if the risk also includes a loss of outside assets, and a loss resulting from the interruption of the electrical power supply, then the investment may need to be matched with the new ALE, which may be increased as a result of the incremental risk. For example, the new ALE would be the sum of the total annualised replacement costs for the exposed outside asset (in this case, assumed to be £10,000) and the exposed computer room, plus the maximum annual down-time loss. The new ALE would be £360,000. This new ALE should be compared with the total cost of all counter-measures, as discussed above.

In comparing these costs, the prototype system considers the maximum ALE for all classes of risk exposure, and compares it against the total cost of all the recommended counter-measures as one control set. In other words, the annual cost for all disaster prevention measures, which protect against all classes of risk exposure, is calculated in total before it can be compared with the maximum ALE. Similarly, the annual costs of disaster recovery measures and insurance as another control set is compared with the same maximum ALE. It should be noted that, when analysing several classes of risk exposure, the prototype system does not consider the ALE for a single class of risk exposure, or the justification of a single counter-measure, within a specified set, because this can be achieved by consultation when running the system for a single class of risk exposure, thus avoiding repetition.

Chapter 7

CONCLUSION

In order to present a concise view of the conclusions which may be drawn from this work, an overview of the various aspects of the research described in the previous chapters alongside the requirements and objectives of this research is firstly discussed. The value of the research will then become clear from this discussion and will also be reported in this chapter followed by some suggestions of possible areas of further related research.

7.1 Overview

Previous incidents of computer disaster have shown that service interruptions may result from direct damage to IT facilities or from the loss of electrical power, key staff, access to working areas etc. Most of these incidents arise from threats (see table 4.2) which originate outside the immediate accommodation of IT installations from which previous research has not provided adequate Risk Management guidance.

This research has concluded that this area is of key concern to IT risk managers and thus warrant a serious consideration. The implications of these threats can be substantial in terms of direct financial loss and loss of service.

Although IT services have become critical to the operation of almost all organisations, more risk management attention has been given to means of

recovering from disasters (including insurance) than to measures which can be taken to prevent them.

Available literature has recognised that disaster prevention policies can provide more comprehensive assurances of service continuity than recovery measures or the transfer of risks through insurance. These conclusions were, however, reached with only disaster prevention measures in the IT installation's immediate accommodation in mind. Clearly this accommodation is only a subset of the total picture. The risk of flood, for example is most likely to arise externally, where it is also best controlled. Human safety is also an important issue and should be regarded as a key component in service continuity, yet it has also received little previous attention. Some of the counter-measures addressed by this research which could be provided against the destruction of assets and the denial of service are shown in table 4.3.

The approaches taken to minimising the effect of computer disasters upon IT services have been based on Risk Management concepts. Risks to computer installations need first to be defined and analysed, so that suitable preventive measures can be identified. Effective Risk Management in the computer disaster prevention field is critically dependent upon identifying and analysing risk entities such as assets, threats, counter-measures and their interactions which influence risk exposures (e.g. to destruction and/or denial of service). This research has concluded that factors (as illustrated in Chapter 1) which could contribute to risk exposure, need more thorough identification and analysis than has previously been achieved.

Guidance for the identification, analysis and management of risks, as a result has also been lacking, especially in areas outside the immediate vicinity of a computer installation. This has led to the lack of a full methodology for computer disasters prevention which can be implemented by an IT risk manager.

The findings of this research can then be summarised as follows,

1) Little work has been done in the specific area of computer disaster prevention and that no adequate disaster prevention policies were in use at IT installations.

2) Insurance and disaster recovery measures are valid protection schemes, but because of their limitations (see Chapter 1) they may leave an IT installation in a less than ideal position to ensure service continuity.

3) The direct and indirect impacts which result in IT service interruptions and the disaster prevention controls which can be provided against the destruction of assets and the denial of service have not been investigated thoroughly for risk management concerns.

4) The complex area of interactions between risk entities which determine an installations overall vulnerability (or risk exposure) has still to be fully addressed and that the form of analysis which takes account of these interactions has previously been acknowledged to be unclear (Clark, 1989).

Based on the above findings the opportunity was taken in this research to develop and describe a new and adequate approach for implementing disaster prevention policies. This approach offers methods enabling the prevention or reduction of risks to computer installations and providing guidance for reducing risks to the minimum which can be achieved within tolerable costs.

The decision to undertake this particular field of research was prompted by the realisation that

a) substantial consequential losses (as illustrated in Chapter 1) which results from service interruptions are real threats to government, commerce and industry, and that

b) no fully adequate risk management methodology existed for computer disaster prevention which can be used by IT risk managers.

7.2 Requirements and Objectives

The development of a structural computer disaster prevention methodology and its incorporation into a system which provides full support in this important area of decision making became central to this project.

In order to help meet the requirements of this research, it has been necessary to form a comprehensive understanding of existing best practice in the risk management and computer disaster prevention field. It has also been necessary to study in detail available technologies and tools to arrive at a confident selection of the best methods for developing and delivering the required solution.

The main groups of research work undertaken which helped meeting the requirements and objectives of this research are summarised below.

7.2.1 The Framework for IT Risk Management

During the literature review, several risk management methodologies were reviewed and none was found suitable to serve the objectives of this work. There was also a lack of generally accepted standard and of precise measurements, which in turn led to the lack of a unified approach to evaluating risk (see Chapter 2). It was thus difficult to adopt any of these methods for the purpose of this study.

Nonetheless, the concepts of risk analysis and management provided a useful and proven basis for this study. These concepts helped determine a framework (see Chapter 3) which handled all of the risk management phases required for evaluating the risks of computer disasters to IT installations.

Briefly this framework suggests four phases of risk evaluation and these are identifying, analysing, assessing and managing the risks of computer disasters. These phases have been significant in ensuring the required rigorousness of Risk Management and their contribution, in turn, was mainly to ascertain the following.

1) Assets, threats and counter-measures need to be identified for rigorous analysis.

2) The interactions and inter-dependencies between these risk entities need to be analysed to determine an installation's overall vulnerability (or risk exposure).

3) The level of risk exposure needs to be assessed so that the potential loss to which the installation is exposed can be calculated and used for risk management purposes.

4) The required level of investment in appropriate counter measures needs to be based on a cost justification scheme so that the best cost-effective counter-measures can be selected.

This framework has therefore been essential to the successful development of the computer disaster prevention methodology and the decision support system for use by IT risk managers.

7.2.2 The methodology for IT disaster prevention

A significant part of this study, therefore, has involved developing a methodology which considers all of the above risk management phases. The rationale behind developing such methodology was emphasised by the need for advancing a method which would provide practical help for IT risk managers when evaluating the risks of computer disasters. This evaluation would not have been possible unless data and knowledge about assets, threats and counter-measures are captured and systematically analysed. Therefore it was found necessary in this research to design a method which would help IT risk managers to capture and record information about these risk entities so that risk exposures can be analysed and assessed. This method is represented by the use of the "three environment" approach which has been fully described in Chapter 4. In brief, this approach is in the method of considering the influences of the geographic location of a computer installation upon the risks to which it is exposed. It considers the computer installation as residing within three "concentric" environments, namely;

A) The controlled environment, which is the immediate IT installation.

B) The semi-controlled environment, which is the building housing the installation.

C) The uncontrolled environment, which is the surrounding area.

This approach provides a common structure within which all relevant facts about assets, threats and counter measures, and their interactions and inter-dependencies can be captured, analysed and assessed. More significant, it assisted the IT risk manager in making full use of the recommended risk management framework in that

1) All necessary data and knowledge about risk entities can be captured and recorded by their location during the risk identification phase.

2) The complex interactions and inter-dependencies between these risk entities can be fully taken account of during the risk analysis phase.

3) Information needed for the risk assessment phase which had been gathered and recorded during the risk identification phase can be used to determine an installation's overall loss exposure.

4) Data available on the annualised costs of counter-measures can be used alongside the information on loss exposure which has been calculated in the risk assessment phase to help in the selection of the most cost-effective counter measures.

The "three environment" approach therefore has been a key element which enabled the development of a structured computer disaster prevention methodology. It has been successful for the following reasons,

1) It provides powerful facilities to help the IT risk manager to conduct a risk management exercise for computer disaster prevention.

2) It imposes a disciplined structured approach on the analyst ensuring that knowledge about risk entities and their relationships is consistently observed.

3) It allows the IT risk manger to determine the boundaries for each environment so that features and risk entities which influence risk exposures within it can be identified and analysed.

4) It assists considering how to protect the installation by using a "three main lines of defence" strategy to ensure service continuity.

7.2.3 Knowledge Based Systems (KBS) technology

A principal requirement of this research has also considered the development and delivery of a solution which could test the validity of the developed methodology and provide it to the IT risk manager in a usable form.

Again, during this research, many approaches were studied, but after their evaluation it was concluded that a development approach based upon Knowledge based system methods would be the most suitable. In this study well established Risk Management concepts were combined with KBS technology to examine their possible applicability in computer disaster prevention.

As stated, effective Risk Management in the computer disaster prevention field involves all of the phases described in the framework, and can only be exercised if data and knowledge about risk entities and their interaction and inter-dependencies are captured and systematically analysed. KBS technology was found to be well able to handle data and knowledge about risk entities and their interaction and inter-dependencies, and to infer and report on risk exposures. This is because KBS technology is capable of handling the expert heuristics, relations and "rules" which need to be considered in Risk Management. For example,

a) it allows the knowledge of domain experts about the complex interactions between risk entities and factors influencing risk exposure to be represented in a suitable form (e.g. frames and rules).

b) it holds and processes this knowledge to infer and report on the level of risk exposure to which the installation is exposed.

c) it uses the information on potential risk exposures to assess an installation's overall potential loss exposure by using additional input data as required.

d) it uses the resulting potential loss value as one factor, alongside others such as cost-effectiveness, in advising on the selection of appropriate counter-measures.

It is clear from the above that a systematically structured knowledge representation of the elements involved in Risk Management was needed for clarity and to ensure that elements and their relations with other elements were not overlooked.

Because the knowledge based system had to hold and process domain knowledge from many sources in a shared knowledge base, the "three environment" approach to considering the problem area, already described was also adhered to consistently.

This approach has significantly contributed to the building process of the knowledge base. It is assumed that the types of knowledge which need to be considered in dealing with the knowledge acquisition and knowledge representation tasks have already been introduced by the use of this approach.

7.2.4 The prototype knowledge based decision support system (KBDSS)

A fundamental requirement of the solution to be developed was that it should provide effective assistance or support for the decision making process involved in computer disaster prevention. In the development of the decision support system, advantage has been taken of the incremental prototyping approach encouraged by KBS technology. A prototype knowledge based decision support system has been

developed to test the feasibility of the IT disaster prevention methodology suggested.

This system is now a working prototype and has been fully described in Chapter 6. Examples of the underlying code have been given in Appendix A. The discussion that follows describes the prototype from two points of view.

A) How Risk Management concepts were combined with KBS technology to provide a solution, and

B) How that solution was able to produce recommended decisions which result from analysis and comparison of the data by applying heuristics supplied by domain experts.

7.2.4.1 System design and development

The prototype system's purpose is to demonstrate how an IT risk manager can be provided with a decision support system as part of an overall disaster prevention policy. In both the design and development phases of this system, advantage was taken of the incremental style of development permitted by the KBS technology used. For this project, the main gain was to change the merging prototype system's design to incorporate the amounts of information yielded by the study of available literature. The same ability will, of course , be of even greater value in delivering a full production knowledge based decision support system for IT disaster prevention, where a full set of user input on all possible assets, threats and counter-measures (or controls) will need to be processed.

The prototype KBDSS deals specifically with the risk of external flood, and the reasons for selecting this particular threat as the problem domain have already been described in Chapter 6. More significant, because

1) It is a threat that has shown to warrant serious considerations from IT risk managers, and its impacts can be substantial in terms of direct financial losses, and of loss of service.

2) it has not been given the same level of attention as to for example, the threat of fire or unauthorised intrusion.

3) it has not been receiving adequate risk management attention, especially in the areas of risk identification and effective flood prevention measures, even though some published standards covering counter measures were available (e.g. US Flood-Proofing Regulations).

The prototype has been developed using Logic Programming Associates Ltd's ***flex*** product, principally because

a) it is portable across MS DOS 386 and 486 platforms (including Microsoft Windows), as well as Apple Macintosh and UNIX, giving flexibility of choice in delivery vehicle, and

b) its English-Like Knowledge Specification Language (KSL) allows easy-to-read expert systems to be produced; and its frame-based, data-driven and rule-based functionality, fully integrated into a prolog environment, cater well not only for the development tool requirements of the prototype system, but also for the onward development of the prototype system into a full production decision support system.

7.2.4.2 Prototype structure and functionality

The prototype KBDSS uses the structured framework already described. It analyses the interactions and inter-dependencies between risk entities to determine the installation's overall risk exposure. For example, if the user indicates that a flood barrier is the only counter-measure in place for a site exposed to flooding, the system can infer that this counter-measure may only be relevant to certain types of assets (e.g. buildings or access) and therefore may not prevent disruption of the external electrical supply, or the computer room if it is located below the expected flood level. It then reports on the actual risk exposure for the types of assets exposed.

The prototype KBDSS therefore, is capable of handling the interactions and inter-dependencies between these risk entities, and to infer and to report on risk exposures. The synthesis of this knowledge provides a new way of solving the problems in the area of computer disaster prevention. The "three environment" approach ensures that all necessary data on risk elements was recorded by their location, and available to KBS for rigorous analysis.

This approach is new in providing such a structure, which allows the many types of interactions and inter-dependencies among the risk entities to be considered in identifying risk exposures.

The "three environment" approach has therefore been essential to the successful development of the prototype KBDSS for use in managing these risks for Computer Disaster Prevention. It enables the evaluation of risk exposures in the zone immediately surrounding the IT installation. However, this evaluation may not always allow the user to address all of the extreme variations of threat severity which can arise across situations. These variations will typically be caused by geographic or topographic factors. For example, the severity of the threat of

external flood varies substantially, depending on a site's proximity to, large bodies of water.

In order to ensure that the prototype KBDSS could cater for the interactions and inter-dependencies of external flood-risk entities in differing types of areas, the "three environment" approach was further refined by introducing a method of hazard exposure zoning. This refinement was found necessary for examining the risk of flood, arising from several types of exposure zones.

The Prototype System models risk entities, such as the location of assets and existing counter-measures, through their identification in the three environments. The risks of these assets are analysed from data concerning the site's three environments, and from the hazard exposure zone in which the installation is sited. On the other hand,

a) if that review shows an unacceptably high risk exposure with unjustifiable costs for counter measures, or

b) if a new or alternative site is being considered,

the risk exposure level for several different hazard exposure zones may need to be analysed and considered. These may include the proximity of hazards, boundaries, elevations or meteorological and geological conditions. The maximisation of protective counter measures (which may involve selecting a different site) may also help in reducing the exposure. Knowledge about

a) extent of a threat level at a location;

b) the availability of relevant counter-measures, and their effectiveness; and

c) the interactions and inter-dependencies of threats, assets and counter-measures

may be elicited from domain specialists, such as meteorologists, water authorities, architects etc.

The prototype KBDSS analyses the interactions and inter-dependencies among risk entities, to determine the risk exposure. The domain knowledge is represented in the Knowledge Base, and processed using production rules. This rule-based approach allows the heuristics of domain experts to be represented in such a way that a conclusion about a specific risk exposure can be reached.

Up to this point, the prototype KBDSS considers the data and knowledge input and reports identified risk exposures. In order to assess the potential cost of loss exposure, further "knowledge" is required, about, for example

a) the cost of replacing destroyed assets;

b) the average daily loss due to each asset's non-availability;

c) the time needed to restore services; and

d) the expected frequency of threat occurrence.

From this data, the potential loss exposure (cost) from a threat occurrence is assessed. By considering also the cost of annualised counter-measures, the system carries out a cost benefit analysis of the loss exposure and appropriate counter-measures, to determine which Risk Management actions can be justified. It then recommends the best protective measures for the site.

7.3 Concluding Remarks

Because continuity of service is vital to most IT installations, much emphasis was given to the informed selection of counter-measures to provide protection against the destruction of IT assets and/or the loss of these services. This selection included counter-measures against the destruction of assets and denial of service.

It was guided by the rigorous evaluation of exposure type and their likely impact, and by considering the complex interactions and inter-dependencies among hazard-exposure zone boundaries, threats, counter-measures and assets.

The prototype KBDSS developed as part of the study deals specifically with the risk of external flood, and incorporates a "three environment" approach and a hazard exposure zoning method which were developed to handle that risk. Other types of risk may be considered using the underlying methodology, but different factors would be involved. For example, the factor "elevation", which may reduce exposure to the flood risk, may be irrelevant to exposure to a fire which arises naturally. Also, different hazards may demand different counter-measures. For example, in the case of fire, fire walls are a good protection; for flood, automatic flooding systems are appropriate.

Because the risk considered in the prototype is that of flood, and since "water finds its own level" it was assumed that, if flood penetrated an exposure-zone, all assets located there would be affected, even though some items might later be re-used. Thus, the position of a boundary line to separating two exposure-zones with different degrees of risk exposure was dictated, in the case of flood, by elevation and flood-proofing measures. In considering other risks, a modified approach may be needed. For the fire risk, the position of a boundary line between two exposure-zones will be dictated by the presence of fire walls, fire doors, extinguishers, sprinkler systems etc. It was therefore clear that, as between different risks , the interactions and inter-dependencies among exposures-zones, threats, counter-measures and assets were not exactly the same.

7.4 Value of research

Since computer disasters do not occur frequently, data on which to base computer disaster prevention is limited. This study develops a prototype knowledge based decision support system which shows what information is needed, and how that information is evaluated, to guide decisions regarding investments in flood protection measures. This decision support system is expected to improve the quality of these decisions by considering the interactions of risk entities and allowing counter-measures to be rigorously evaluated.

The new value in this research is in the previously unavailable levels of service protection advice which can be given to IT risk managers. What is new, and offers significant benefit for the IT community, is the much wider scope which can be given to IT Risk Management; and the opportunity for practical disaster prevention policies.

The principal benefits arising from this study are the more assured retention of assets, including

expensive, sophisticated equipment;

valuable software;

data (where the cost of re-creation can be high);

staff with scarce specialist skills; and vitally,

the service provided by the computer installation

This study has also addressed the question, "Who will benefit from this research?" It is expected that the following will benefit by adopting the Disaster Prevention methodology developed.

a) IT Risk Managers at mainframes or other large installations, for whom the methodology can be a powerful Decision Support System.

b) Insurance underwriters and other risk assessors.

c) Managers of IT installations with high value services, where the availability of service is critical for revenue, e.g. banking, stock exchanges.

d) Managers of IT installations where any service interruption is likely to be significant for operational reasons e.g. major telecommunications, military or other emergency services.

e) IT Risk Managers who are responsible for designing or planning risk prevention measures at new computer centres.

f) Any manager who is responsible for risk control measures at existing IT installations.

7.5 Possible Areas for Further Related Research

The following areas are suggested as deserving of further study, as a result of observations during this research.

i) The testing of the validity of the results of this research, and of the methodologies developed during this study, in the areas of:-

other types of naturally arising threat, e.g. windstorms, fire, volcanic eruptions;

all types of logical threat, such as unauthorised access to premises, systems and data; and

the various forms of man-made threats, such as industrial espionage, terrorist activity, civil disturbance.

ii) The consideration of alternative methods for the development of Computer disaster Prevention Decision Support System, such as object-oriented programming and neural networks.

iii) The development of two generic relational databases, one to act as a "default" register of all the assets likely to be found in an IT installation; and the second as a "default" check-list of all the types of counter-measures which may be appropriate for consideration. In both cases, "default sets of interactions and inter-dependencies should be included. The objective here would be to further aid the work to the IT manager responsible for implementing a Computer Disaster Prevention policy.

iv) To consider the applicability of KBS technology as the basis for assisting the management and control of other types of high-risk/low-probability threats, such as major fraud, nuclear disaster.

References

AASGARD, D.O. et al (1979)

"An Evaluation of Data Processing Machine Room Loss and Selected Recovery Strategies"

University of Minnesota, Management Information Systems Research Centre.

AFRAMP (No Date)

"Air Force Risk Analysis Management Program"

AF Reg. 300-XX, Vol.I-III, U.S.

ALTY, J.L. and COMBS, M.J. (1984)

"Expert Systems - Concepts and Examples"

The National Computing Center.

BS 6266 (1982)

"British Standard Code of Practice for Fire Protection for Electronic Data Processing Installations"

British Standards Institution.

BS 7083 (1989)

"British Standard Recommendations for the Accommodation and Operating Environment of Computer Equipment"

British Standards Institution.

BIELAWSKI, L. and LEWAND, R. (1988)

"Expert Systems Development - Building PC-Based Applications"

QED Information Sciences Inc., Wellesley, Massachusetts

BONYUN, D. and JONES, G. (1988)

"An Expert System Approach to the Modelling of Risks in Dynamic Environments"

Proceedings of Computer Security Risk Management Model Builder Workshop, May 24-26, 1988, Denver, Colorado, pp.203-224.

BOYER, Terrence J. (1982)

"Contingency Planning: An Opportunity for DP Management"

Computer Security Journal, Winter, pp.41-49.

BROWNE, Peter S. (1979)

"Security: Checklist for Computer Centre Self-Audits"

AFIPS Press.

CCTA (1987)

"A Description of the background to CRAMM"

Central Computer Telecommunication Agency, Proceedings of the 10th U.S. National Computer Security Conference, Baltimore, M.D., September, 1987.

CCTA (1989)

"Contingency Planning"

Central Computer Telecommunication Agency,U.K. Government Publication.

CCTA (1991)

"An Overview of CRAMM"

Central Computer Telecommunication Agency, The Government Centre for Information Systems,H.M. Treasury, I.T. Security and Privacy Branch, London, U.K.

CAMPBELL, R.P. and SANDS, G.A. (1979)

"A Modular Approach to Computer Security Risk Management"

Dept. ofthe Army, Washington, D.C., U.S.A. AFIPS Conference Proceedings, Vol.48., National Computer Conference, New York,June, 1979, pp.293-303.

CARROLL, J.M. (1984)

"Managing Risk: A Computer-Aided Strategy"

Stoneham, M.A.; Butterworth Publisher.

CHRISTENSEN, Steven R. and SCHKADE, Lawrence L. (1987)

"Financial and Functional Impacts of Computer Outages on Businesses

"The University of Texas at Arlington Center for Research on Information Systems.

CLARK, R. (1989)

"Risk Management - A New Approach"

BIS Applied Systems, London,U.K., Elsevier Science Publishers BV (North-Holland), IFIP.

CUADRADO, C.Y. and CUADRADO, J.L. (1985)

"Prolog Goes to Work"

Byte, August, 1985.

DOA (1977)

"ADP Security Handbook"

(US DA DIPS Manual Supplement)

U.S.Department of Agriculture, August 15, 1977.

DTI (1990)

Department of Trade and Industry

"Expert System Opportunities - Guidelines for the Introduction of Expert Systems Technology"

LONDON : HMSO.

De BACKER, Dr. C. (1981)

"Analysis of the Different Costs of Computer Security"

Convention Informateque Conference: Paris, France, 1981, Vol..2, pp.467-472.

DEPARTMENT OF THE ARMY, U.S. (1972)

"Flood-Proofing Regulations"

Washington, D.C., Office of the Chief of Engineers, June 1972, 79P.

ELBRA, R.A. (1992)

"Computer Security Handbook"

NCC Blackwell Limited, Oxford, England.

FIPS PUB. 31 (1974)

"Guidelines for Automatic Data Processing, Physical Security and Risk Management"

Federal Information Processing Standards Publication, U.S. Department of Commerce, National Bureau of Standards.

FIPS PUB. 65 (1979)

"Guidelines for Automatic Data Processing Risk Analysis"

Federal Information Processing Standards Publication, U.S. Department of Commerce, National Bureau of Standards.

FAITHFULL, M. and WATT, S. (1991)

"The Survivor's Guide to I.T. Centre Design"

Elsevier Science Publishers Ltd

FORDYCE, S. (1982)

"Computer Security: A Current Assessment"

Computers and Security, January, 1982, pp.9-16.

FORSITH, R. (1984)

"Expert Systems: Principles and Cases Studies"

Chapman and Hall.

FROST, R.A. (1986)

"Introduction to Knowledge Base Systems"

William Collins Sons & Company Ltd.

GILBERT, Irene E. (1989)

"Guide for Selecting Automated Risk Analysis Tools"

NIST Special Publication, 500-174, U.S. Department of Commerce, National Institute of Standards and Technology.

GOLDBLUM, Edward (1987)

"Computer Disasters and Contingency Planning"

A Research Study Sponsored by the Amdahl Corporation and Carried Out by Butler Cox & Partners Limited, 1987.

HAACK, Marr T. (1984)

"Insuring the Data Processing Risk"

Bests Review, January, 1984, pp.44, 46, 48, 50, 100.

HARMON, P. and KING, D. (1985)

"Artificial Intelligence in Business Expert Systems"

John Wiley & Sons Inc.

HARMON, P.; MAUS, R. and MORRISSEY, W. (1988)

"Expert Systems Tools and Applications"

John Wiley & Sons Inc.

HART, A. (1986)

"Knowledge Acquisition for Expert Systems"

Kogan Page Ltd., New Technology Modular Series.

HAYES-ROTH, F.; WATERMAN, D.A. and LENAT, D. (1983)

"Building Expert Systems"

Addison-Wesley.

HOFFMAN, L.J. and NEITZEL, L.A. (1980)

"Inexact Analysis of Risk"

Proceedings of the 1980 I.E.E.E.International Conference on Cybe-netics and Society, October 1980.

HOFFMAN, L.J.; MICHELMAN, E.H. and CLEMENTS, D. (1983)

"Securate-Security Evaluation and Analysis Using Fuzzy Metrics"

AFIPS Conference Proceedings, 1983, Vol.47, pp.531-540.

HU, S.D. (1990)

"Expert Systems for Software Engineers and Managers"

Chapman and Hall, Ltd.

IBM (1980)

"Security Assessment Questionnaire"

International Business Machines,GX20-2381-0, 1980.

IBM (1985)

"Security Assessment Questionnaire"

International Business Machines, GX20-2381-1,1985

IST/RAMP (1979)

"RAMP - What It Is ..... How To Use It ..... What It Does"

International Security Technology Inc., 1979

JACKSON, P. (1986)

"Introduction to Expert Systems"

Addison-Wesley.

JACKSON, P. (1990)

"Introduction to Expert Systems"

Second Edition, Addison-Wesley.

JOHNSON, L. and KERAVNOU, E.T. (1985)

"Expert Systems Technology: A Guide"

Abacus Press, Information Technology & Systems Series

KATZKE, S. (1985)

"Summary of Key Issues Presented as Panel Session - Outline of a Conceptual Model for Risk Management"

Federal Information System Risk Analysis Workshop, Montgomery, Alabama, January, 1985.

KATZKE, S. (1988)

"A Government Perspective on Risk Management of Automated Information Systems"

Proceedings of Computer Security Risk Management Model Builder Workshop, May 24-26, 1988, Denver, Colorado, pp.3-20.

KELLER, R. (1987)

"Expert Systems Technology: Development and Application"

Yourdon Press, Engelwood Cliffs, NJ.

KRAUSS, L.I. (1980)

"Contingency Planning - How to Survive a Disaster"

Management Review, June 1988, pp.78-83.

KRIZ, J. (1987)

"Knowledge Based Expert Systems"

Ellis Horwood, Books in Information Technology

KUNREUTHER, Howard (1976)

"Limited Knowledge and Insurance Protection"

Public Policy, 1976, Vol.24,pp.227-261.

LEWIS, N. (1988)

"Using Binary Schemas to Model Risk Analysis"

Proceedings of Computer Security Risk Management Model Builder Workshop, May 24-26, 1988, Denver, Colorado, pp.35-48.

LUGER, G.F. and STUBBLEFIELD, W.A. (1989)

"Artificial Intelligence and the Design of Expert Systems"

The Benjamin/Cummings Publishing Company, Inc

MARCELLA, Albert J. (1985)

"Disaster Recovery Planning - The Next Step"

Corporate Accounting, Spring, 1985, pp.60-66.

MAYERFIELD, H. (1988)

"Definition and Identification of Assets as the Basis for Risk Management"

Proceedings of Computer Security Risk Management Model Builder Workshop, May 24-26, 1988, Denver, Colorado.pp.21-34.

MICHIE, D. (1979)

"Expert Systems in the Micro-electronic Age"

Edinburgh University Press, Edinburgh

MICHIE, D. (1982)

"Introductory Readings in Expert Systems"

Gordon and Breach, Science Publishers,Inc.

MOSES, Robin H. and GLOVER, Ian (1988)

"Introduction to CRAMM"

Proceedings of Computer Security Risk Management Model Builder Workshop, May 24-26, 1988, Denver, Colorado.

MOSLEH, A. (1988)

"A Matrix/Bayesian Approach to Risk Management of Information Systems"

Proceedings of Computer Security Risk Management Model Builder Workshop, May 24-26, 1988, Denver, Colorado, pp.103-116.

NBS SPECIAL PUBLICATION, 500-133 (1985)

"Technology Assessment: Methods for Measuring the Level of Computer Security"

U.S. Department of Commerce, National Bureau of Standards.

NATIONAL AUDIT OFFICE and CCTA (1987)

A U.K. Government Report on I.T. Installation Risks.

NIELSEN, Norman R. and RUDER, Brian (1980)

"Computer System Integrity Vulnerability"

Information Privacy, Vol.2, No.1, January, 1980.

OTTWELL, Ken and ALDRIDGE, Bruce (1989)

"The Role of Vulnerability in Risk Management"

Computer Security Journal, Volume VI, Number 1.

PALMER, I.C. and POTTER, G.A. (1989)

"Computer Security Risk Management"

Jessica Kingsley Publishers in Association with the Control Risks Group, London.

PARKER, D.B. (1981)

"Managers Guide to Computer Security"

Reston, VA : Reston Publishing, 1981

PATTERSON-HINE, F.A. and KOEN, B.V. (1987)

"Comment on: An Algorithm for Exact Fault-Tree Probabilities Without Cut Sets"

I.E.E.E. Transactions on Reliability, December, 1987, Vol. R-36, No. 5.

PINDER, B. and HOVER (1990)

"Notes from Contingency Planning Seminar"

Datashield, U.K., 1990.

PRICE WATERHOUSE (1990)

"Disaster Recovery and Business Continuity Planning: How to Approach Business Survival"

A Seminar for Senior Management, November 7th, 1990, Newcastle upon Tyne, U.K.

ROMAN, David (1986)

"Beyond Risk: Keep DP Risks in Hand"

Computer Decisions, June 30, 1986, pp.58-61.

SDC (1979)

"Navy Risk Assessment Methodology"

System Development Corporation, U.S.,TM-WD-7999/001/03, July 1979.

SALTMARSH, Timothy J. and BROWNE, Peter S. (1983)

"DATA Processing - Risk Assessment"

Advances in Computer Security Management, Vol.2, Edited by H. M. Wofsey, 1983, John Wiley & Sons Ltd.

SCHMIDT, E. (1988)

"Conceptual Model of the Risk Management Process"

Proceedings of Computer Security Risk Management Model Builder Workshop, May 24-26, 1988, Denver, Colorado, pp.89-102.

SCHWIEGER, Bradley J. (1986)

"Status of Disaster Recovery Plans for Total Operations"

The Internal Auditor, October 1986, pp.28-30.

SELL, P. (1985)

"Expert Systems - A Practical Introduction"

Macmillan.

SHAIN, Michael (1989)

"Information Security for Managers"

U.K., Macmillan Publishers Ltd.,1989.

SIMONS, G.L. (1985)

"Expert Systems and Micros"

NCC Publications, Manchester.

SMITH, M.R. (1989)

"Commonsense Computer Security"

McGraw-Hill Book Company.

SMITH, S.T. (1986)

"LAVA : A Conceptual Framework for Automated Risk Assessment"

Los Almos Labs. Document LA-UR-86-2282, Los Almos National Laboratory, 1986.

SMITH, S.T. (1988)

"LAVA : An Expert System Framework for Risk Analysis"

Proceedings of Computer Security Risk Management Model Builder Workshop, May 24-26, 1988, Denver, Colorado, pp.179-202.

SWANK, B. William (1981)

"Disaster Recovery Plan"

Journal of System Management, October 1981,pp.16-20.

TAMERIN, J.S. and ROSNIK, H.L.P. (1971)

"Risk Taking by Individuals, Option-Case Study: Cigarette Smoking"

Perspectives on Benefit Risk Decision Making, Report of a Colloquium Conducted by the Committee on Public Engineering Policy, Washington: National Academy of Engineering.

TIGO, Jon William (1989)

"Disaster Recovery Planning"

Yourdon Press, Prentice Hall Building, Englewood Cliffs, N.J., 07632.

TROY, Eugene F. (1988)

"Introduction and Statement of Purpose"

Proceedings of Computer Security Risk Management Model Builder Workshop, May 24-26, 1988, Denver, Colorado, pp.1-2.

TURBAN, Efraim (1990)

"Decision Support and Expert Systems: Management Support Systems"

Collier Macmillan, 1990.

VAN der GAAG, L.C. (1986)

"Prolog: An Expert System Building Tool"

Report CS-R8616, Centre of Mathematics and Computer Science, Amsterdam, April,1986.

WALKER, T.C. and MILLER, R.K. (1986)

"Expert Systems 1986: An Assessment of Technology and Applications"

SEAI Technical Publications, Madison, GA.

WATERMAN, D.A. (1986)

"A Guide to Expert Systems"

Addison-Wesley.

WEISS, S.M. and KULIKOWSKI, C.A. (1983)

"A Practical Guide to Designing Expert Systems"

Chapman and Hall Ltd

WERNSTCIN, N.D. (1980)

"Unrealistic Optimism About Future Life Events"

Journal of Personality and Social Psychology, 1980, Vol.39,pp.806-820.

Appendix A
System Code

Actions

```
action start
do put(12)
and new_slot( delays, site, nothing )
and the possible_controls of the site become
  {barrier and elevation and 'A' and 'B' and 'C' and flood_system
   and refuge and power and access and computer and outside_elev }
and !
and the site`s controls become {computer and refuge}
and ask zone
and check_zone
and forward_chain( fcfs, misfire, fail, once, envirs )
and forward_chain( fcfs, misfire, loss_set, once, loss_rules )
and prove(forward_chain( fcfs, misfire,  fail, once, zone ) )
and isa_slot( losses, site, L )
and write( 'The losses at the site would be ')
and write( L ) and nl and !
and get_losses
and forward_chain( fcfs, misfire,  fail, once, cben_rules )
and get_insurance_details
and show_results and ! .

action fc(G)
do forward_chain( fcfs, misfire, loss_set, once, G ) .
action fc2(G)
```

do forward_chain(fcfs, misfire, fail, once, G) .

Frames

frame site

default losses is nil

and default delays is {0}

and default total_cost is 0 .

frame building

default flood_system is nil .

frame uncon_envir

default controls are nil.

frame semicon_envir .

Groups

group loss_rules

zone1_1, zone1_2, zone1_2a, zone1_3, zone1_4, zone1_5, zone1_6,

zone2_1, zone2_2, zone3_1, zone4_1.

group zone1

z1power_1, z1power_2, z1access_1, z1access_2,

z1refuge_1, z1refuge_2, z1refuge_3, z1refuge_4, z1refuge_5, z1refuge_6,

z1computer_1, z1computer_2, z1computer_3, z1outside_elev .

group old_cb_rules

cb_outside_1, cb_power_1, cb_access_1, cb_outside_2, cb_power_2, cb_access_2,
cb_computer_1, cb_computer_1a, cb_computer_2, cb_computer_2a, cb_computer_3,
cb_building_1, cb_building_2, cb_building_3,
cb_personnel_1, cb_personnel_2 .

group sug_rules z2_1, z2_2, z3, typeA, typeB, barrier, elevation .

group envirs envir_1, envir_1a, envir_2, envir_4, envir_5 .

group zone2
z2power_1, z2outside_1, z2access_1 .

group zone3 z3power_1 .

group cben_rules cb1, cb2, cb3.

Questions

question zone
'In which zone is the site';
choose one of zone1, zone2, zone3, zone4 ;
because '1-below RFD 2-as 1 but elevated 3-above the RFD in parts 4- all above the RFD' .

question uncontrolled_measures
'Uncontrolled environment measures in force';
choose some of barrier, elevation .

question barrier_question

'Is there an effective barrier in place ?';

choose one of yes, no .

question semi_controlled_measures

'What is the type of building';

choose one of 'A', 'B', 'C' ;

because 'A - completely water-proof, B - partially water-proof, C - not water-proof '.

question automatic_flood_system

'Is there an AFS - Automatic Flooding System';

choose one of yes, no

because ' An AFS will flood the building to counter ballance the external flood

pressures' .

question site_access

'Is there site access above the Regulatory Flood Datum';

choose one of yes, no .

question reserve_power

'Is there reserve power above the Regulatory Flood Datum';

choose one of yes, no .

question site_refuge

'Is there personnel refuge above the Regulatory Flood Datum';

choose one of yes, no .

question computer_location

'Is the computer room above the Regulatory Flood Datum';

choose one of yes, no .

```
question outside_elevation
'Is the outside elevation above the Regulatory Flood Datum';
choose one of yes, no .

question daily_loss
'What is the average daily loss to the business';
input X such that number(X).

question time_down
'How long will it take to recover the full business service';
input X such that number(X).

question flood_frequency
'What is the frequency of flood occurance' ;
input X such that number(X) ;
because 'e.g. 0.1 means once in ten years' .

question building_loss
'How much will it cost to replace the BUILDING';
input X such that number(X).

question outside_loss
'How much will it cost to replace all of the vulnerable OUTSIDE ASSETS';
input X such that number(X).

question computer_loss
'How much will it cost to replace the COMPUTER ROOM';
input X such that number(X).
```

question personnel_loss

'How much will it cost to replace any PERSONNEL that may be lost in the flood (not insurances)' ;

input X such that number(X);

because 'Include recruitment, training and relocation but not life insurance costs' .

question power_cost

'What will be the cost of locating reserve POWER above the Regulatory Flood Datum';

input X such that number(X).

question access_cost

'What will be the ANNUAL cost of installing ACCESS above the Regulatory Flood Datum';

input X such that number(X).

question elevation_cost

'What will be the ANNUAL cost of ELEVATING the building above the Regulatory Flood Datum';

input X such that number(X).

question outside_elevation_cost

'What will be the ANNUAL cost of ELEVATING the outside assets above the Regulatory Flood Datum';

input X such that number(X).

question barrier_cost

'What will be the ANNUAL cost of building a BARRIER to prevent flooding';

input X such that number(X).

question refuge_cost
'What will be the ANNUAL cost of building a REFUGE to save personell';
input X such that number(X).

question comp_rfd_cost
'What will be the ANNUAL cost of raising the COMPUTER ROOM above the Regulatory Flood Datum';
input X such that number(X).

question 'type A cost'
'What will be the ANNUAL cost of improving the building to type A - completely water-proof';
input X such that number(X).

question 'type B cost'
'What will be the ANNUAL cost of improving the building to type B - partially water-proof ';
input X such that number(X).

question flood_system_cost
'What will be the ANNUAL cost of an AUTOMATIC FLOODING SYSTEM';
input X such that number(X).

question power_down
'What would be the down time due to loss of POWER';
input X such that number(X).

```
question access_down
'What would be the down time due to loss of ACCESS';
input X such that number(X).

question building_down
'What would be the down time due to the loss of the BUILDING';
input X such that number(X).

question outside_down
'What would be the down time due to the loss of the OUTSIDE ASSETS';
input X such that number(X).

question personnel_down
'What would be the down time due to the loss of PERSONNEL';
input X such that number(X).

question computer_down
'What would be the down time due to the loss of the COMPUTER ROOM';
input X such that number(X).

question property_ins
'What is the annual premium of insurance for the dammage ?' ;
input X such that number(X).

question service_ins
'What is the annual premium of insurance for loss of service ?' ;
input X such that number(X).

question recovery_measures
```

'What is the annual cost of recovery measures ?' ;

input X such that number(X).

Relations

relation check_for_controls

if the uncon_envir has a barrier

or the uncon_envir has a elevation .

relation the Frame has a Slot

if the Slot of the Frame is effective .

relation there are no controls

if the controls of the uncon_envir are nothing

and the type of the building is not 'A'

and the type of the building is not 'B' .

relation there are no controls

if the controls of the uncon_envir are nothing

and the type of the building is 'B'

and the flood_system of the building is nil .

relation get_losses % in case of only outside losses

if the site`s losses is {outside}

and get_costs(outside)

and maximum(site`s delays, Max)

and write('The maximum DOWN TIME is ') and write(Max)

and write('The REPLACEMENT COST is ') and write(site`s total_cost) and nl

and ask flood_frequency

```
and ARC is site`s total_cost * flood_frequency
and write('The ANNUAL REPLACEMENT COST is ') and write(ARC) and nl
and the annual_loss_exp of the site becomes ARC
and write('The TOTAL ALE is ') and write( annual_loss_exp of the site ) and nl
and offer_controls .

relation get_losses      % the usual case
if the site`s losses is List
and List is not nothing
and for every on(Item, List)
   do get_costs(Item)
   end for
and ask daily_loss
and maximum( site`s delays, Max )
and write('The maximum DOWN TIME is ') and write( Max )
and Down_loss is daily_loss * Max
and write( ' : the MAXIMUM DOWN TIME loss is ') and write( Down_loss ) and nl
and write( 'The REPLACEMENT COST is ') and write(site`s total_cost) and nl
and ask flood_frequency
and ADTL is Down_loss * flood_frequency
and ARC is site`s total_cost * flood_frequency
and write('The ANNUAL DOWN TIME loss is ') and write(ADTL) and nl
and write('The ANNUAL REPLACEMENT COST is ') and write(ARC) and nl
and the annual_loss_exp of the site becomes ADTL+ARC
and write('The TOTAL ALE is ') and write( annual_loss_exp of the site ) and nl
and offer_controls .

relation get_losses
if write('There will not be losses in this case.') and nl
```

```
and !
and fail .

relation the site is in zone1 or zone2
if zone = zone1 or zone = zone2
and ! .

relation check_type
if the site`s controls do not include barrier
and the site`s controls do not include elevation
and building`s type becomes the answer to semi_controlled_measures
and the answer to semi_controlled_measures is 'B'
and ask automatic_flood_system
and check_flood_system .
relation check_type .

relation check_flood_system
if automatic_flood_system = yes
and building`s flood_system becomes effective
and include the flood_system in the site`s controls.
relation check_flood_system .

relation check_zone
if the site`s location becomes zone
and the answer to zone is zone2
and the elevation of the semicon_envir becomes effective .

relation check_zone .
```

```
relation loss_set
if lookup( losses, site, L)
and L is not nil.

relation offer_controls
if suggestions become the site`s possible_controls
and the site`s controls are Controls    % these existing controls are set by rules
and for every on(Item, Controls )
   do remove Item from suggestions
   end for
and forward_chain( fcfs, misfire, fail, once, sug_rules)
and get_control_costs .

relation get_insurance_details
if the site`s losses are { outside }
and service_ins becomes 0
and recovery_measures becomes 0
and ask property_ins
and other_insurance  .

relation get_insurance_details
if the site`s losses are { access }
and property_ins becomes 0
and ask service_ins
and other_insurance  .

relation get_insurance_details
if the site`s losses are { power }
and property_ins becomes 0
```

```
and ask service_ins
and other_insurance .

relation get_insurance_details
if the site`s losses are { access and power }
and property_ins becomes 0
and ask service_ins
and other_insurance .

relation get_insurance_details
if the site`s losses are { power and access }
and property_ins becomes 0
and ask service_ins
and other_insurance .

relation get_insurance_details
if  ask service_ins
and ask property_ins
and other_insurance .

relation other_insurance
if  recovery_measures > -1
and site`s comb_prop_serv becomes property_ins + service_ins
and write( 'Comb property and service insurances = ') and write( site`s
comb_prop_serv )
and site`s comb_prop_rec becomes property_ins + recovery_measures
and nl and write( 'Comb property insurance and recovery costs = ')
and write( site`s comb_prop_rec ) and nl
```

and write('The total annual control costs are : ') and write(site`s
annual_control_costs)

Rules

rule zone1_1
if the location of the site is zone1
and there are no controls
then site`s losses become {building and computer_room and outside and power and access} .

rule zone1_2
if the location of the site is zone1
and the uncon_envir has a barrier
then the site`s losses become {power} .

rule zone1_2a
if the location of the site is zone1
and the uncon_envir has a elevation
then the site`s losses become {outside and power and access} .

rule zone1_3
if the location of the site is zone1
and the type of the building is 'A'
then the site`s losses become {outside and power and access} .

rule zone1_4
if the location of the site is zone1
and the type of the building is 'B'

and the building has a flood_system

then the site`s losses become {outside and power and access} .

rule zone1_5

if the location of the site is zone1

and the type of the building is 'B'

and the building does not have a flood_system

then the site`s losses become {building and outside and power and access} .

rule zone1_6

if the location of the site is zone1

and the type of the building is 'C'

then the site`s losses become {building and outside and power and access} .

rule zone2_1

if the location of the site is zone2

and the uncon_envir has a barrier

then the site`s losses become {power} .

rule zone2_2

if the location of the site is zone2

and the uncon_envir does not have a barrier

then the site`s losses become {outside and power and access} .

rule zone3_1

if the location of the site is zone3

then the site`s losses become {power} .

rule zone4_1

if the location of the site is zone4

then the site`s losses become nothing .

rule z1access_1

if the location of the site is zone1

and the type of building is not 'B'

and the type of building is not 'C'

and the site`s losses include access

and the answer to site_access is yes

then remove access from the site`s losses

and include access in the site`s controls .

rule z1access_2

if the location of the site is zone1

and the type of building is 'B'

and the building`s flood_system is effective

and the site`s losses include access

and the answer to site_access is yes

then remove access from the site`s losses

and include access in the site`s controls .

rule z1power_1

if the location of the site is zone1

and the type of building is not 'B'

and the type of building is not 'C'

and the site`s losses include power

and the answer to reserve_power is yes

then remove power from the site`s losses

and include the power in the site`s controls .

rule z1power_2

if the location of the site is zone1

and the type of building is 'B'

and the building`s flood_system is effective

and the site`s losses include power

and the answer to reserve_power is yes

then remove power from the site`s losses

and include the power in the site`s controls .

rule z1refuge_1

if the site`s location is zone1

and the type of building is 'B'

and the building`s flood_system is not effective

and the answer to site_refuge is no

and the answer to uncontrolled_measures is nothing

then include personnel in the site`s losses

and remove refuge from the site`s controls.

rule z1refuge_2

if the site`s location is zone1

and the type of building is 'C'

and the answer to site_refuge is no

and the answer to uncontrolled_measures is nothing

then include personnel in the site`s losses

and remove refuge from the site`s controls .

rule z1refuge_3

if the site`s location is zone1

and the type of building is not 'B'

and the type of building is not 'C'

and the answer to site_refuge is no

and the answer to uncontrolled_measures includes elevation

then include personnel in the site`s losses

and remove refuge from the site`s controls .

rule z1refuge_4

if the location of the site is zone1

and the type of building is 'B'

and the building`s flood_system is effective

and the answer to site_refuge is no

and the site`s controls include refuge

then include personnel in the site`s losses

and remove refuge from the site`s controls .

rule z1refuge_5

if the location of the site is zone1

and the site`s controls include barrier

and the answer to site_refuge is no

and the site`s controls include refuge

then include personnel in the site`s losses

and remove refuge from the site`s controls .

rule z1refuge_6

if the location of the site is zone1

and the type of building is 'A'

and the answer to site_refuge is no

and the site`s controls include refuge

then include personnel in the site`s losses

and remove refuge from the site`s controls .

rule z1computer_1

if the location of the site is zone1

and the type of building is not 'B'

and the type of building is not 'C'

and the site`s location is zone1

and the site`s controls do not include elevation

and the answer to computer_location is no

then include computer_room in the site`s losses

and remove computer from the site`s controls .

rule z1computer_2

if the site`s location is zone1

and the type of building is 'B'

and the building`s flood_system is effective

and the site`s controls do not include elevation

and the answer to computer_location is no

then include computer_room in the site`s losses

and remove computer from the site`s controls .

rule z1computer_3

if the location of the site is zone1

and the type of building is 'B'

and the building`s flood_system is not effective

and the site`s controls do not include elevation

% and the answer to computer_location is no

then include computer_room in the site`s losses

and remove computer from the site`s controls .

```
rule z1outside_elev
if the location of the site is zone1
and the site`s controls do not include barrier
and the answer to outside_elevation is yes
then include outside_elev in the site`s controls
and remove outside from the site`s losses .

rule z2_1
if the location of the site is zone2
and suggestions are SUGS
then take_out( {elevation,'A','B','C',flood_system }, SUGS, NEW_SUGS )
and suggestions become NEW_SUGS .

rule z2_2
if the location of the site is zone2
and the losses of the site do not include {access and power and outside}
and suggestions are SUGS
then take_out( {barrier}, SUGS, NEW_SUGS )
and suggestions become NEW_SUGS .

rule z3
if the location of the site is zone3
then the suggestions become {power} .

rule typeA
if the site`s controls include 'A'
and suggestions are SUGS
then take_out( {barrier, elevation, 'B', 'C', flood_system }, SUGS, NEW_SUGS )
```

and suggestions become NEW_SUGS .

rule typeB

if the site`s controls include 'B'

and suggestions are SUGS

then take_out({barrier, elevation, 'A', 'C'}, SUGS, NEW_SUGS)

and suggestions become NEW_SUGS .

rule existing

if the building of the site is existing

and suggestions are SUGS

then take_out({elevation}, SUGS, NEW_SUGS)

and suggestions become NEW_SUGS .

rule barrier

if the site`s controls include barrier

and suggestions are SUGS

then take_out({elevation, 'A', 'B', 'C', flood_system, access, outside_elev }, SUGS,

NEW_SUGS)

and suggestions become NEW_SUGS .

rule elevation

if the site`s controls include elevation

and suggestions are SUGS

then take_out({barrier, 'A', 'B', 'C', flood_system}, SUGS, NEW_SUGS)

and suggestions become NEW_SUGS .

rule envir_1

if the site`s location is zone1

and the answer to uncontrolled_measures is nothing

then the uncon_envir`s controls becomes the answer to uncontrolled_measures

and include the answer to semi_controlled_measures in the site`s controls

and check_type.

rule envir_1a

if the site`s location is zone1

and the answer to uncontrolled_measures is not nothing

then the uncon_envir`s controls becomes the answer to uncontrolled_measures

and include the answer to uncontrolled_measures in the site`s controls

and check_type .

rule envir_2

if the site`s location is zone2

and the answer to barrier_question is yes

then include barrier in the site`s controls

and the uncon_envir`s controls becomes the barrier .

rule envir_4

if the site`s controls includes barrier

then the barrier of the uncon_envir becomes effective .

rule envir_5

if the site`s controls includes elevation

then the elevation of the uncon_envir becomes effective .

rule z2power_1

if the site`s losses include power

and the answer to reserve_power is yes

then remove power from the site`s losses
and include the power in the site`s controls .

rule z2outside_1
if the site`s controls does not include barrier
and the answer to outside_elevation is yes
then include outside_elev in the site`s controls
and remove outside from the site`s losses .

rule z2access_1
if the site`s controls does not include barrier
and the answer to site_access is yes
then include access in the site`s controls
and remove access from the site`s losses .

rule z3power_1
if the site`s losses include power
and the answer to reserve_power is yes
then remove power from the site`s losses
and include the power in the site`s controls .

rule cb1
if the site`s location is zone1
and the site`s losses include {computer_room and outside and building and access and power and personnel }
then the case becomes a worst_case_scenario .

rule cb2
if the case is a worst_case_scenario

then wcso1z1 becomes barrier_cost + power_cost + refuge_cost + comp_rfd_cost
and the site`s wcso1z1 becomes {barrier and power and refuge and computer }
and wcso2z1 becomes elevation_cost + power_cost + access_cost + refuge_cost +
outside_elevation_cost
and the site`s wcso2z1 becomes {elevation and power and access and refuge and
outside }
and wcso3z1 becomes 'type A cost' + power_cost + access_cost + refuge_cost +
comp_rfd_cost + outside_elevation_cost
and the site`s wcso3z1 becomes {'type A' and power and access and refuge and
computer and outside}
and wcso4z1 becomes 'type B cost' + flood_system_cost + power_cost + access_cost
+ refuge_cost + comp_rfd_cost + outside_elevation_cost
and the site`s wcso4z1 becomes {'type B' and 'flood system' and power and access and
refuge and computer and outside} .

rule cb3
if the site`s location is zone2
and the site`s losses include { outside and power and access }
then new_slot(case, global, wcsz2)
and wcs1z2 becomes barrier_cost + power_cost
and the site`s wcso1z2 becomes { barrier and power }
and wcs2z2 becomes outside_elevation_cost + power_cost + access_cost
and the site`s wcso2z2 becomes { outside and power and access } .

Templates

template has_a the ^ has a ^ ; the ^ does not have a ^ .

template no_conts there are no controls .

template one_or_two	the site is in zone1 or zone2 .

Appendix B

Examples of Input / Output Screens

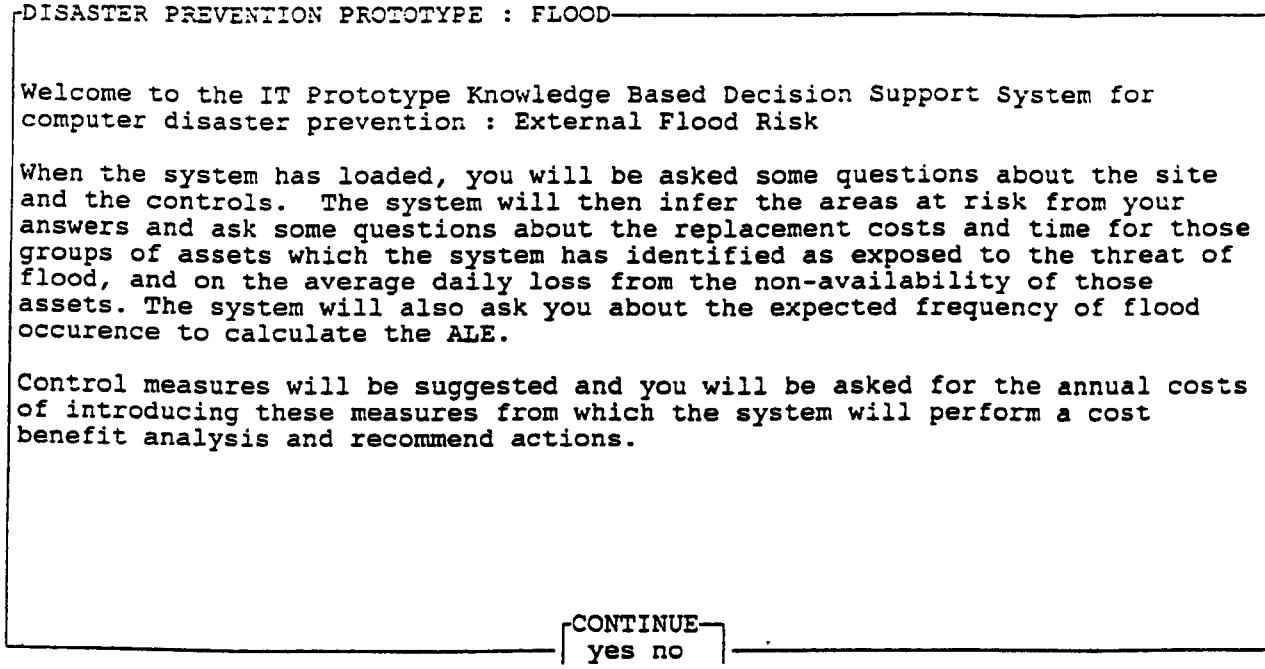
This Appendix provides examples of how the actual input and output screens may appear to the user when running the system for a consultation about the required protective measures for a site exposed to the risk of flood from external sources. It is organised into three parts as follows.

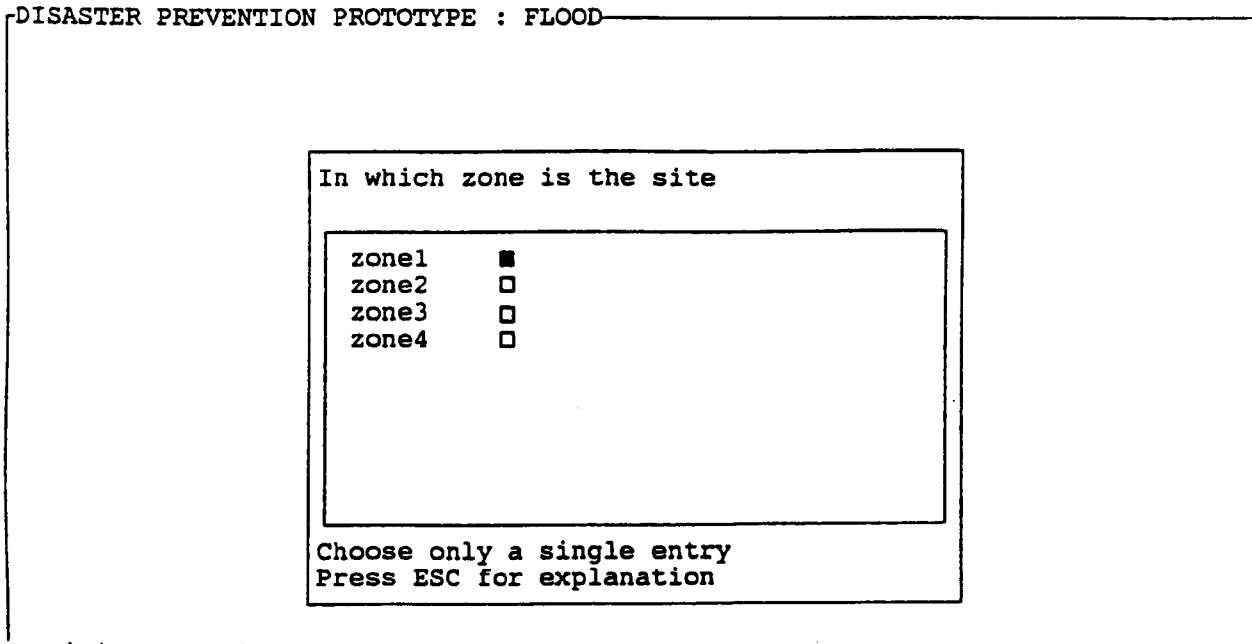
A) the first part deals with Input screens that capture the data which are required for analysing the risk exposure to which the installation is exposed and as a result, an output screen will report on the site's overall vulnerability.

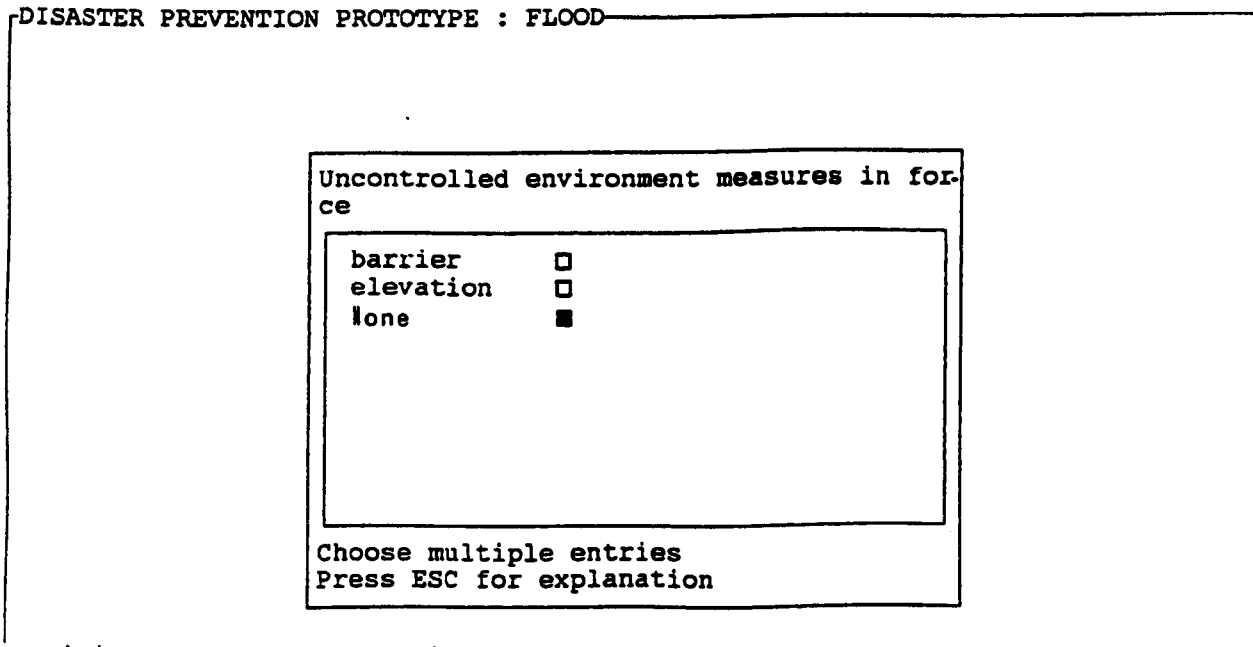
B) the second part deals with further input screens required for assessing the potential loss exposure and two output screens will the report on this loss. The first output screen shows the single loss exposure and the second output screen shows the installation's overall annual loss exposure after considering the data input about the expected frequency of flood occurrence.

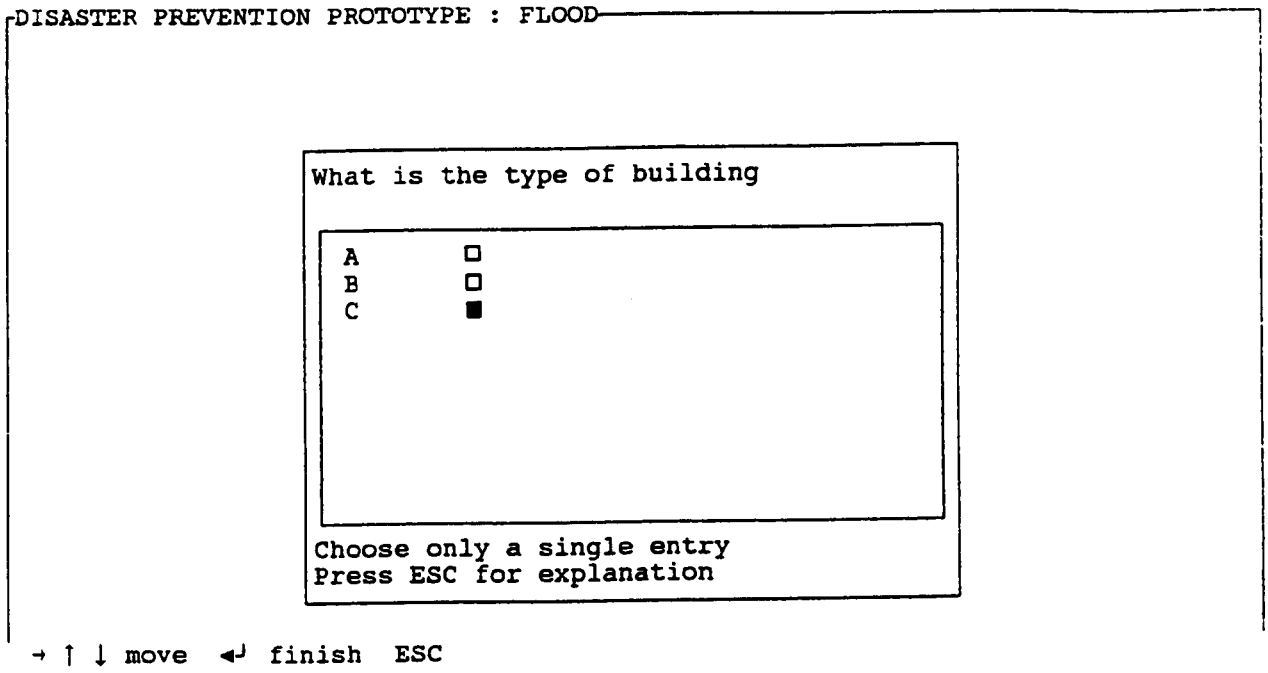
C) the third part deals with Input screens to consider the annualised costs of relevant counter-measures (including insurance and disaster recovery measures costs) and as a result, an output screen will report on the most suitable counter-measures.

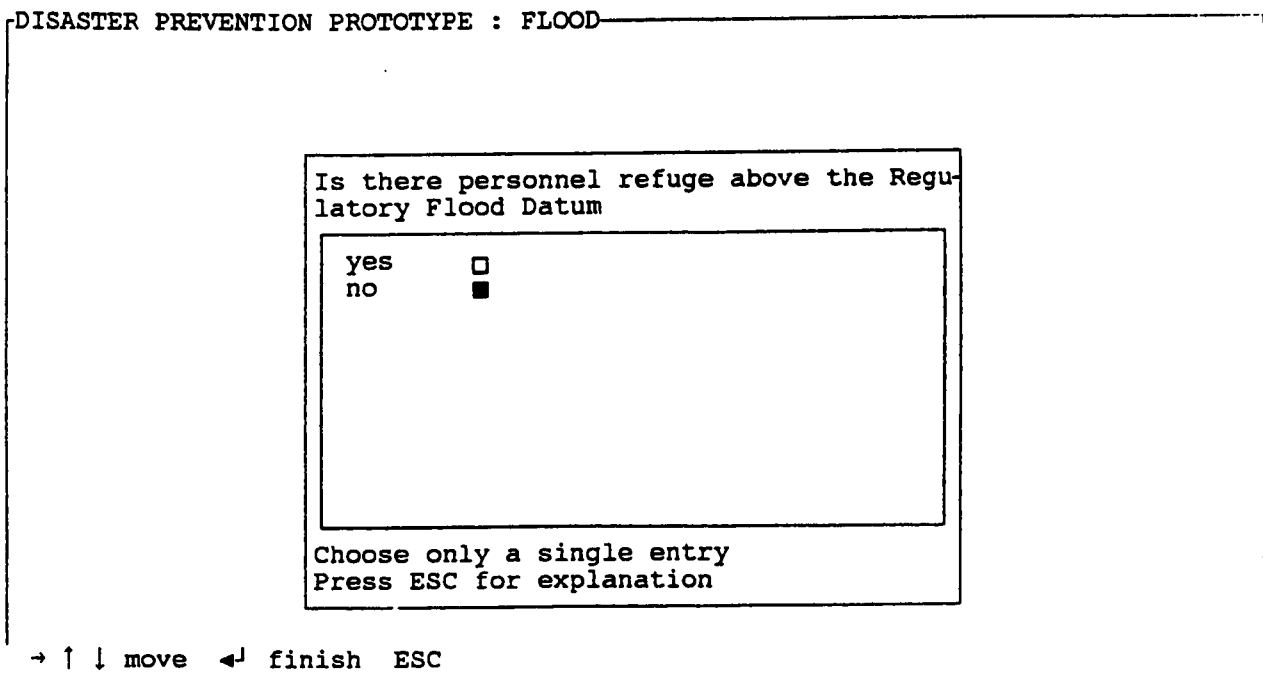
In this particular consultation, it is assumed that the installation is located in zone 1 and all relevant disaster prevention measures should be considered are not in place. It is a worst case scenario for a site located in zone 1 and the prototype system will also allow several consultations to be made as required, taking account of data on the location of the site in a specified zone, and on the presence or absence of relevant counter-measures.

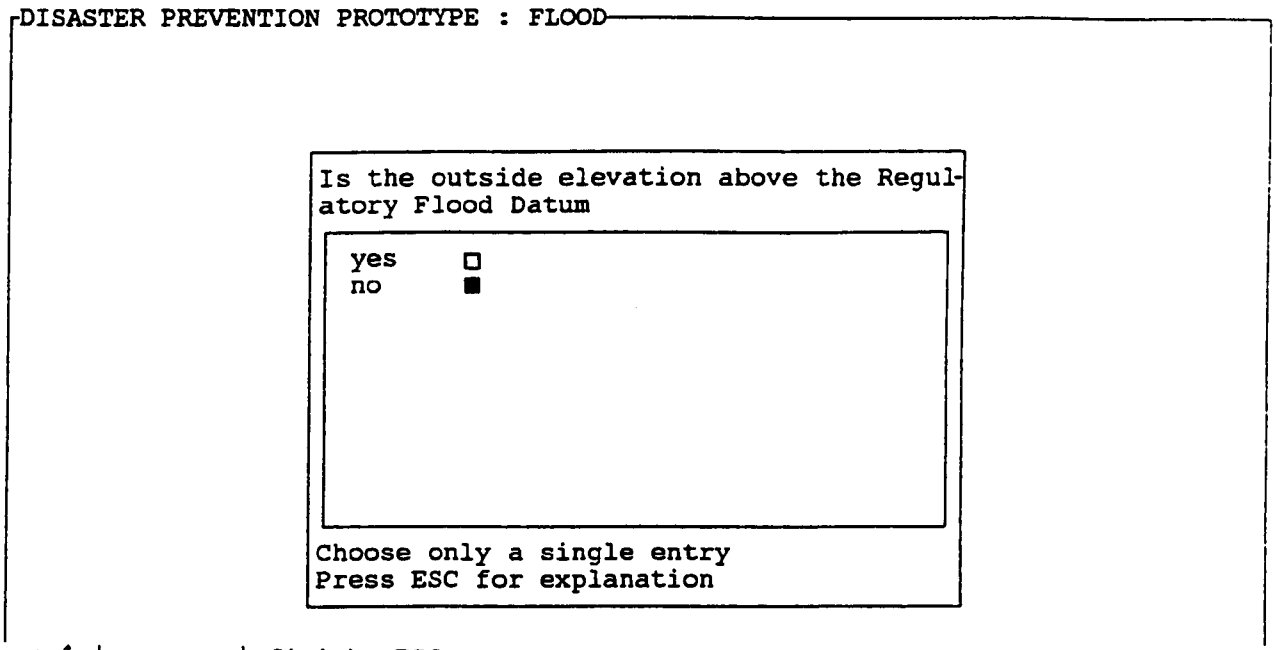

DISASTER PREVENTION PROTOTYPE : FLOOD
Welcome to the IT Prototype Knowledge Based Decision Support System for computer disaster prevention : External Flood Risk
When the system has loaded, you will be asked some questions about the site and the controls. The system will then infer the areas at risk from your answers and ask some questions about the replacement costs and time for those groups of assets which the system has identified as exposed to the threat of flood, and on the average daily loss from the non-availability of those assets. The system will also ask you about the expected frequency of flood occurence to calculate the ALE.
Control measures will be suggested and you will be asked for the annual costs of introducing these measures from which the system will perform a cost benefit analysis and recommend actions.
CONTINUE
yes no

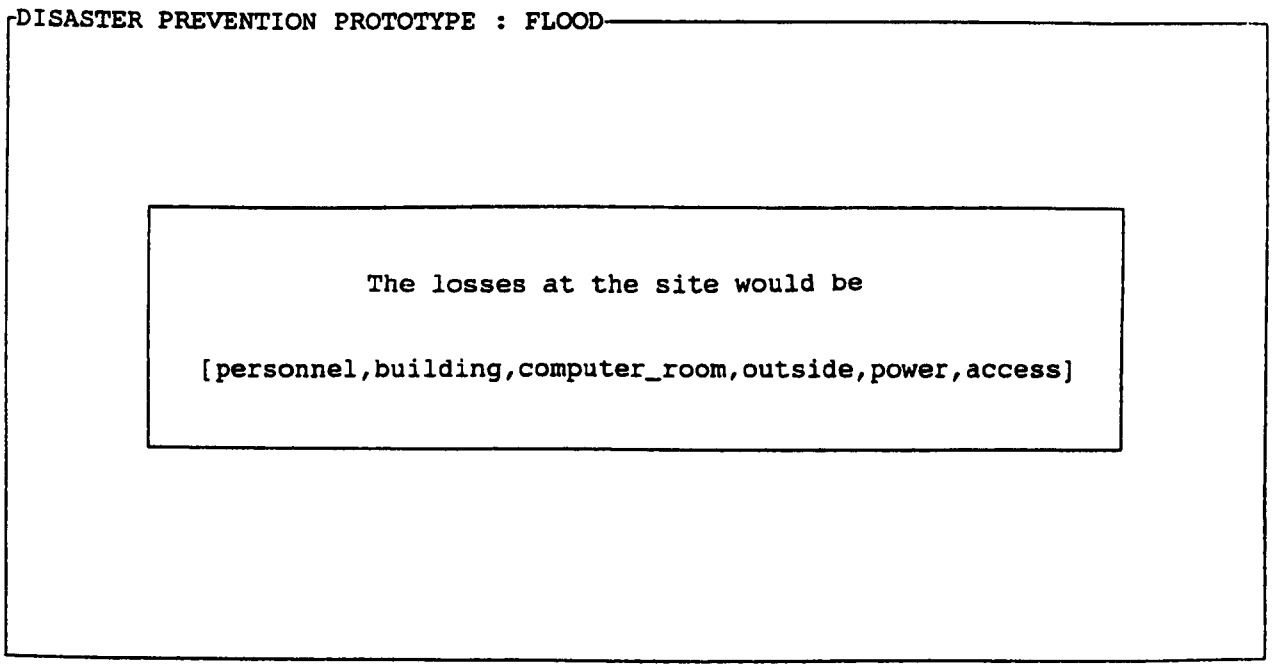

DISASTER PREVENTION PROTOTYPE : FLOOD
In which zone is the site
zone1 ■
zone2 □
zone3 □
zone4 □
Choose only a single entry
Press ESC for explanation
→ ↑ ↓ move ↵ finish ESC


DISASTER PREVENTION PROTOTYPE : FLOOD
Uncontrolled environment measures in force
barrier □
elevation □
none ■
Choose multiple entries
Press ESC for explanation
→ ↑ ↓ move + - on/off ↵ finish ESC


DISASTER PREVENTION PROTOTYPE : FLOOD
What is the type of building
A
B
C
Choose only a single entry
Press ESC for explanation
→ ↑ ↓ move ↵ finish ESC


DISASTER PREVENTION PROTOTYPE : FLOOD
Is there personnel refuge above the Regulatory Flood Datum
yes
no
Choose only a single entry
Press ESC for explanation
→ ↑ ↓ move ↵ finish ESC


DISASTER PREVENTION PROTOTYPE : FLOOD
Is the outside elevation above the Regulatory Flood Datum
yes
no
Choose only a single entry
Press ESC for explanation
→ ↑ ↓ move ↵ finish ESC


DISASTER PREVENTION PROTOTYPE : FLOOD
The losses at the site would be
[personnel,building,computer_room,outside,power,access]

DISASTER PREVENTION PROTOTYPE : FLOOD

How much will it cost to replace any PERSONNEL that may be lost in the flood (not insurances)

5000000

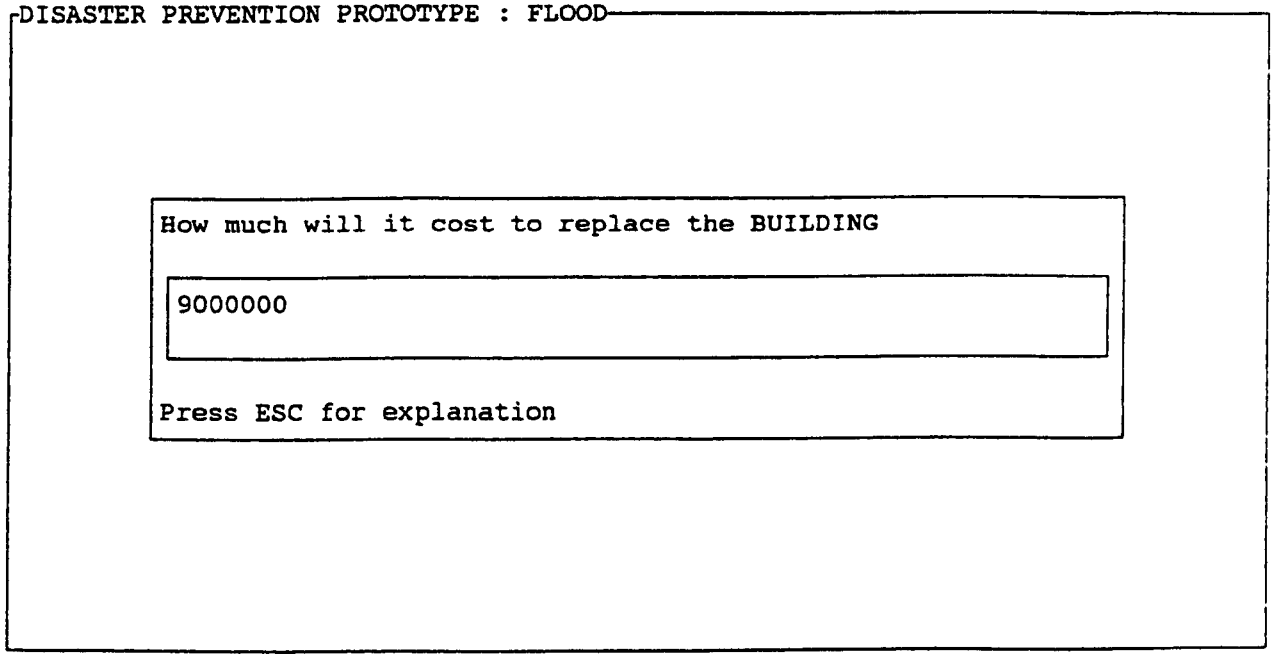
Press ESC for explanation

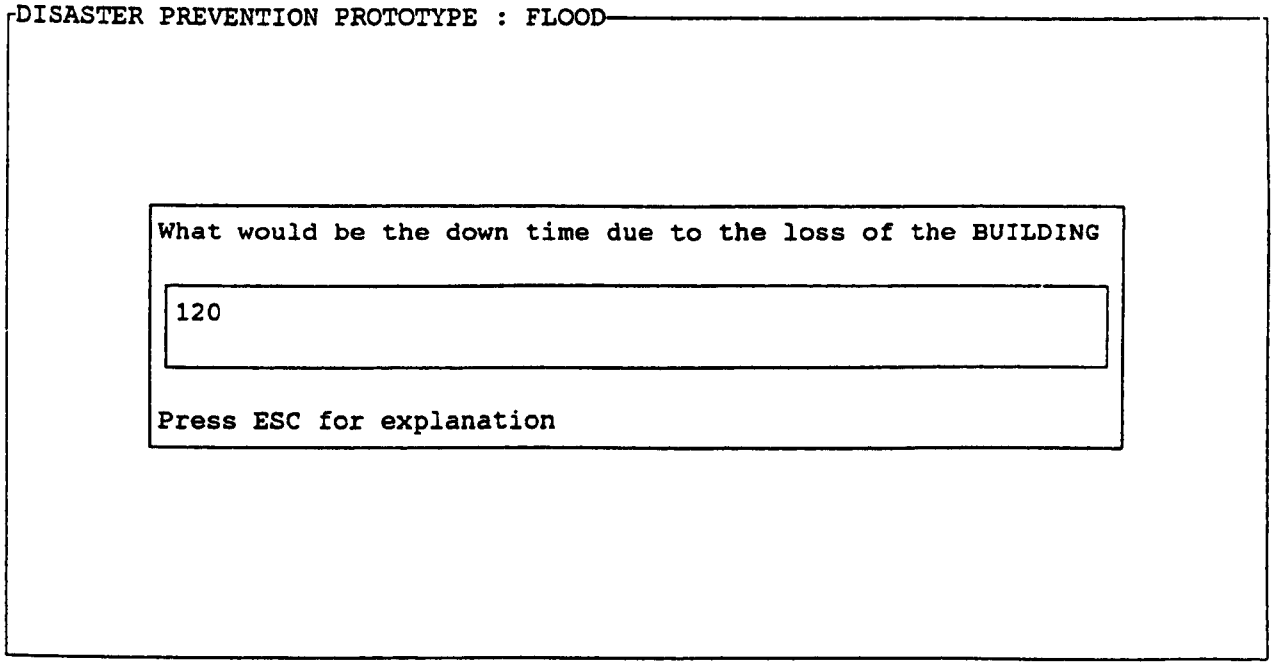
DISASTER PREVENTION PROTOTYPE : FLOOD

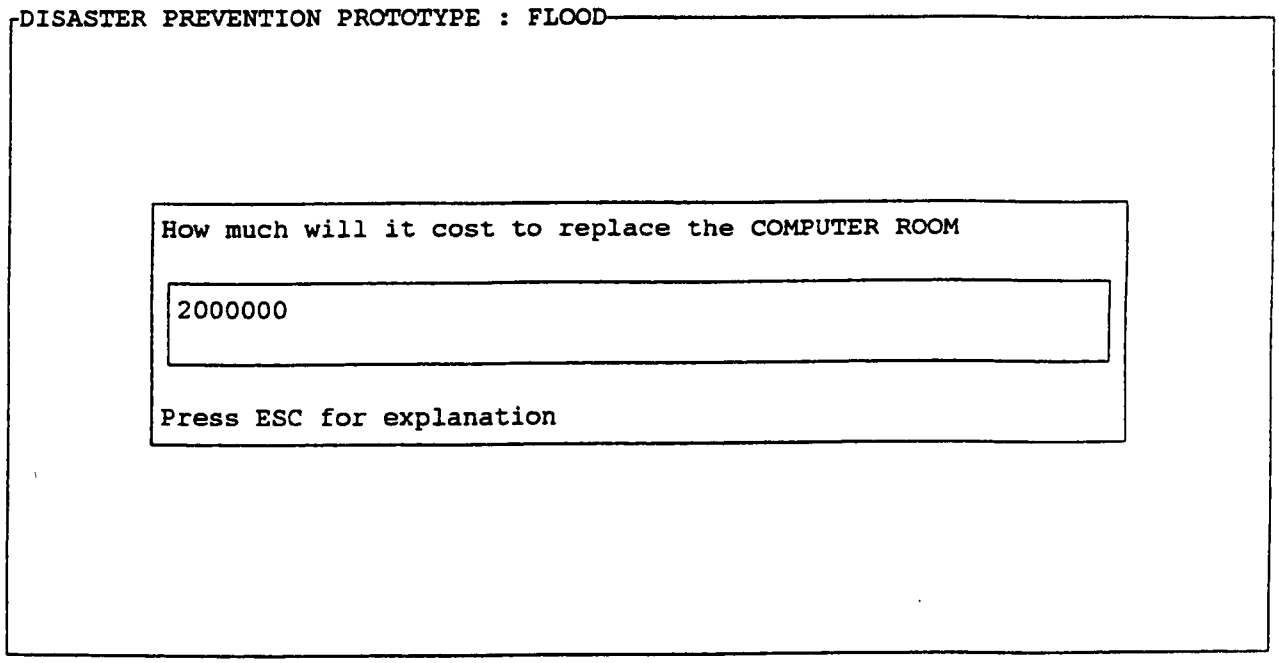
What would be the down time due to the loss of PERSONNEL

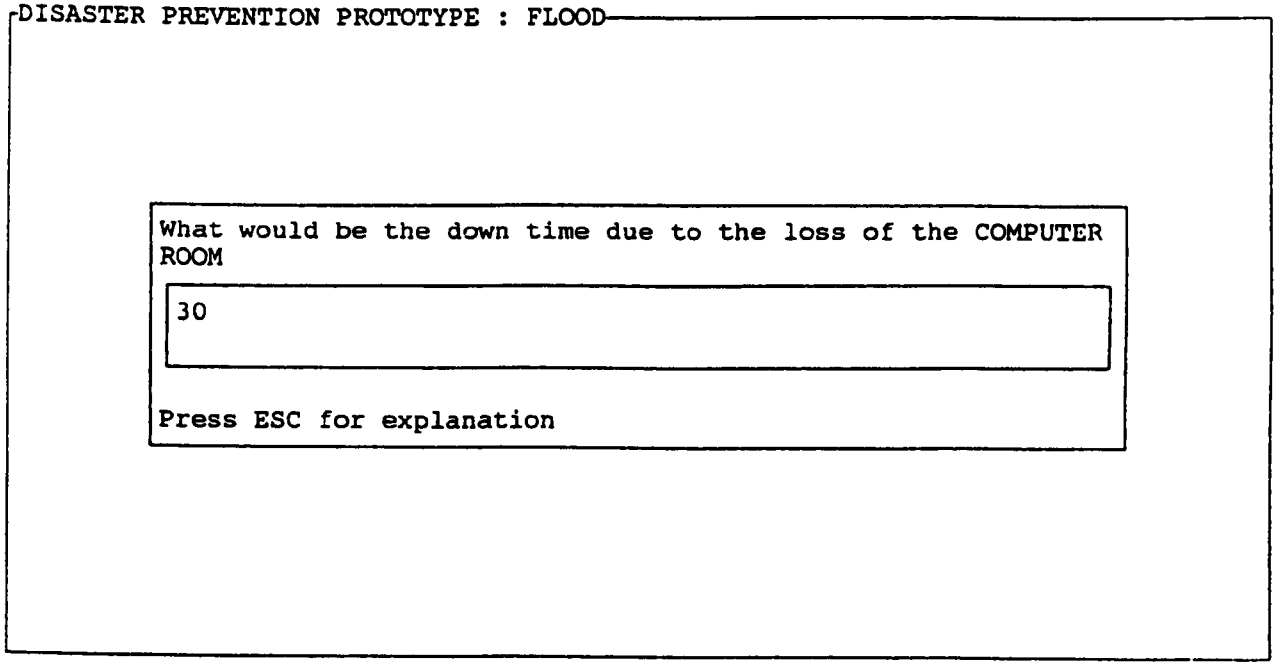
20

Press ESC for explanation


DISASTER PREVENTION PROTOTYPE : FLOOD
How much will it cost to replace the BUILDING
9000000
Press ESC for explanation


DISASTER PREVENTION PROTOTYPE : FLOOD
What would be the down time due to the loss of the BUILDING
120
Press ESC for explanation


DISASTER PREVENTION PROTOTYPE : FLOOD
How much will it cost to replace the COMPUTER ROOM
2000000
Press ESC for explanation


DISASTER PREVENTION PROTOTYPE : FLOOD
What would be the down time due to the loss of the COMPUTER ROOM
30
Press ESC for explanation

DISASTER PREVENTION PROTOTYPE : FLOOD

How much will it cost to replace all of the vulnerable OUTSIDE ASSETS

500000

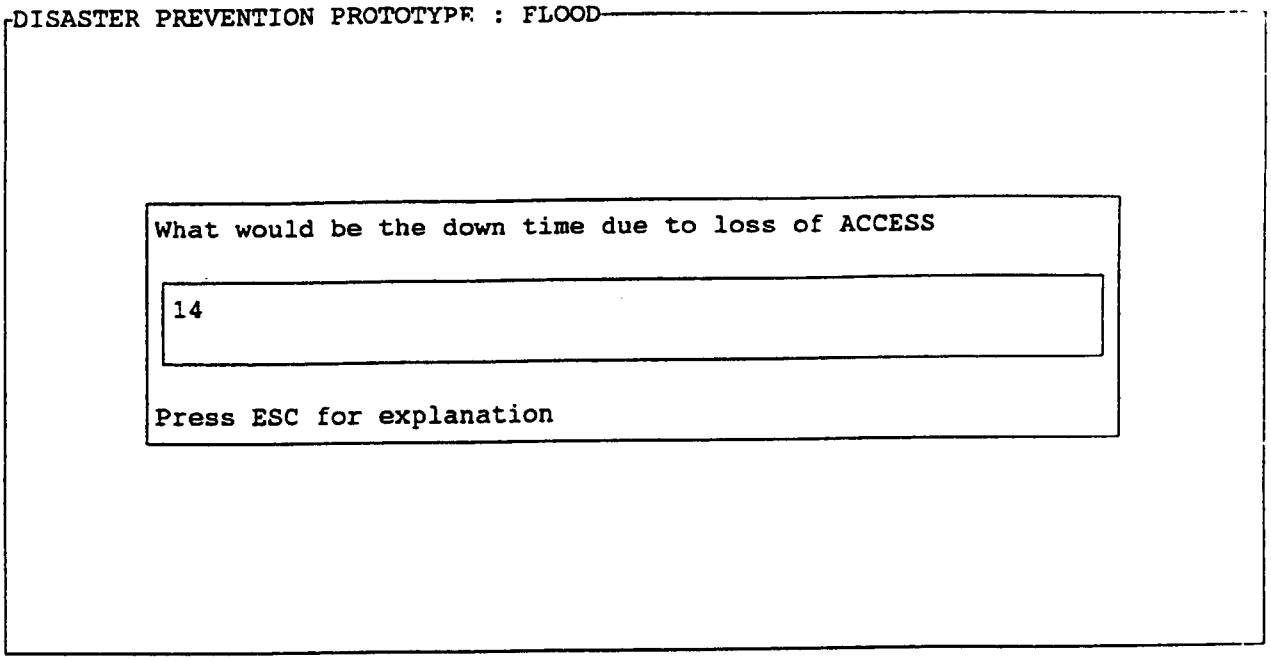
Press ESC for explanation

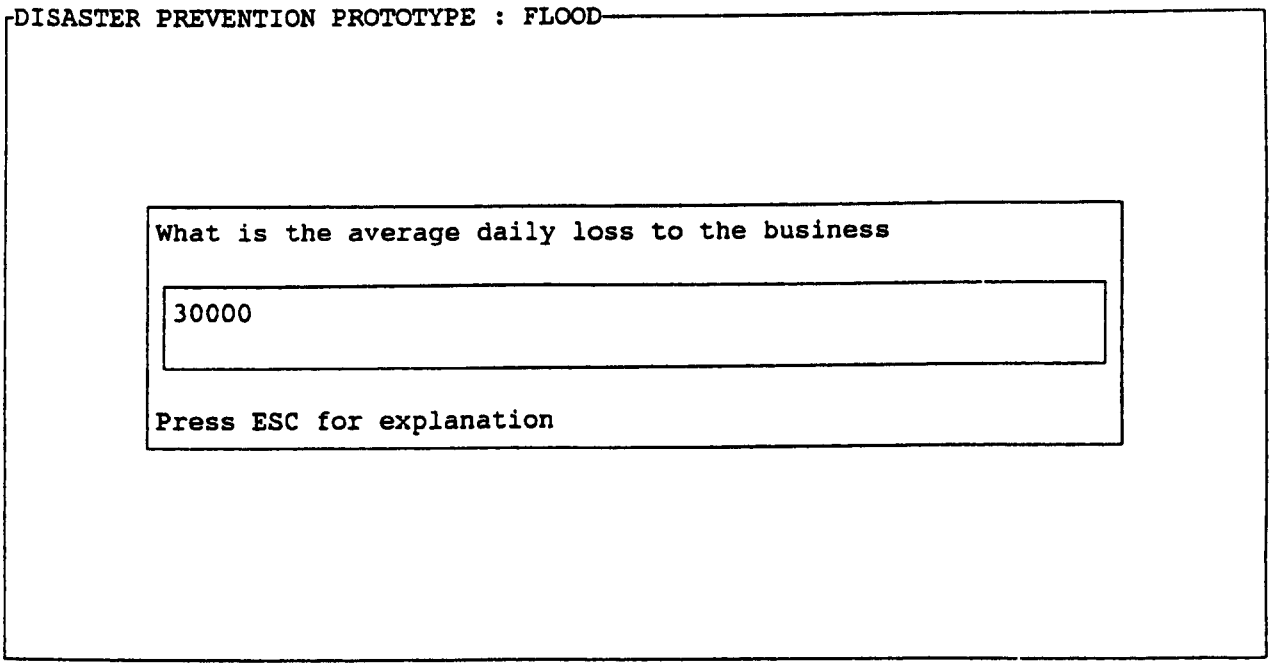
DISASTER PREVENTION PROTOTYPE : FLOOD

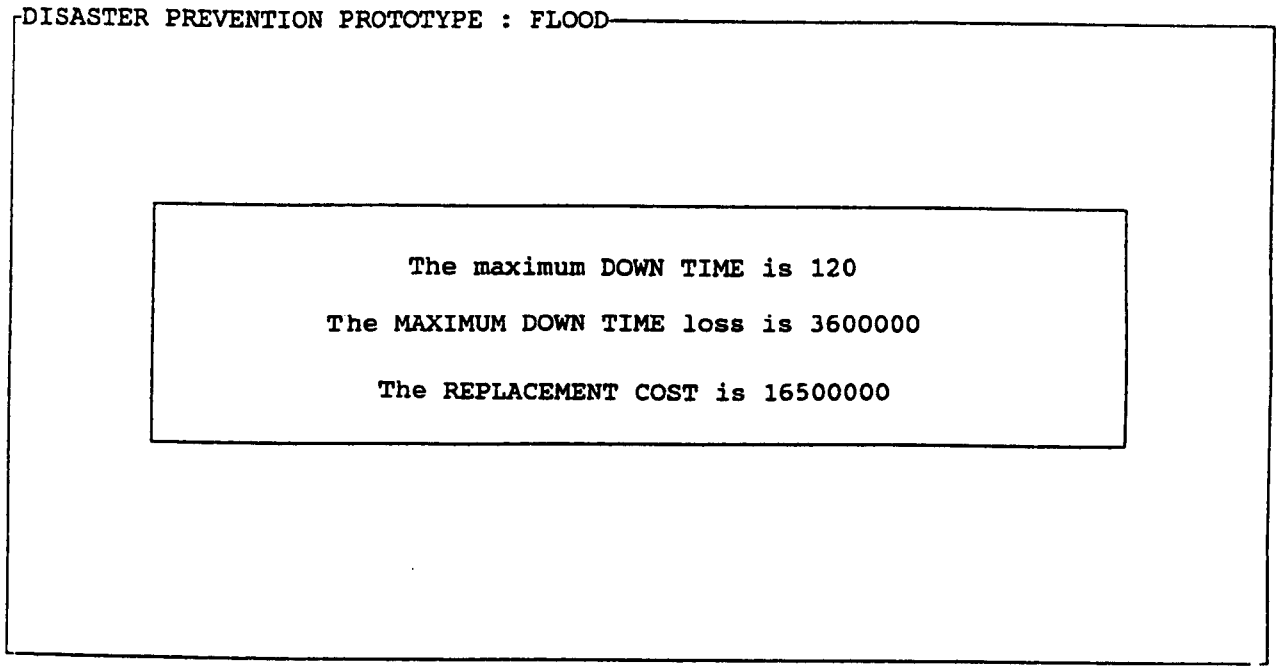
What would be the down time due to loss of POWER

5

Press ESC for explanation


DISASTER PREVENTION PROTOTYPE : FLOOD
What would be the down time due to loss of ACCESS
14
Press ESC for explanation


DISASTER PREVENTION PROTOTYPE : FLOOD
What is the average daily loss to the business
30000
Press ESC for explanation


DISASTER PREVENTION PROTOTYPE : FLOOD
The maximum DOWN TIME is 120
The MAXIMUM DOWN TIME loss is 3600000
The REPLACEMENT COST is 16500000

DISASTER PREVENTION PROTOTYPE : FLOOD

What is the frequency of flood occurance

0.02

Press ESC for explanation

DISASTER PREVENTION PROTOTYPE : FLOOD

The ANNUAL DOWN TIME loss is 72000

The ANNUAL REPLACEMENT COST is 330000

The TOTAL ALE is 402000

DISASTER PREVENTION PROTOTYPE : FLOOD

What will be the ANNUAL cost of building a BARRIER to prevent flooding

50000

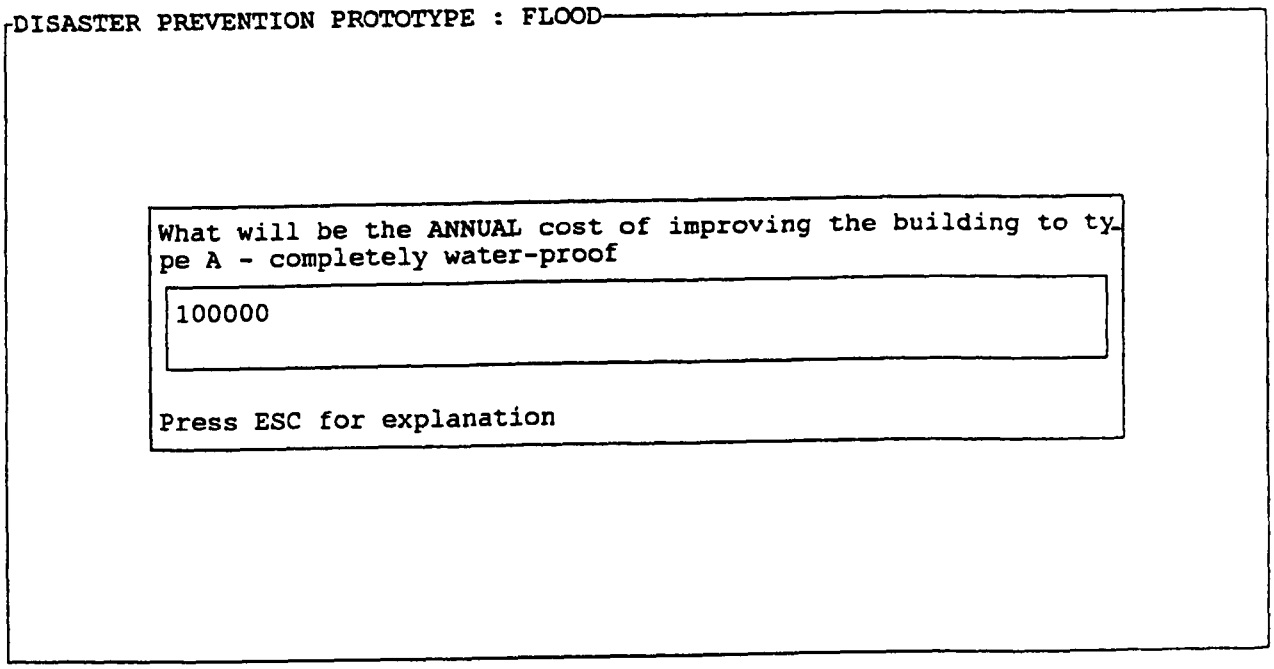
Press ESC for explanation

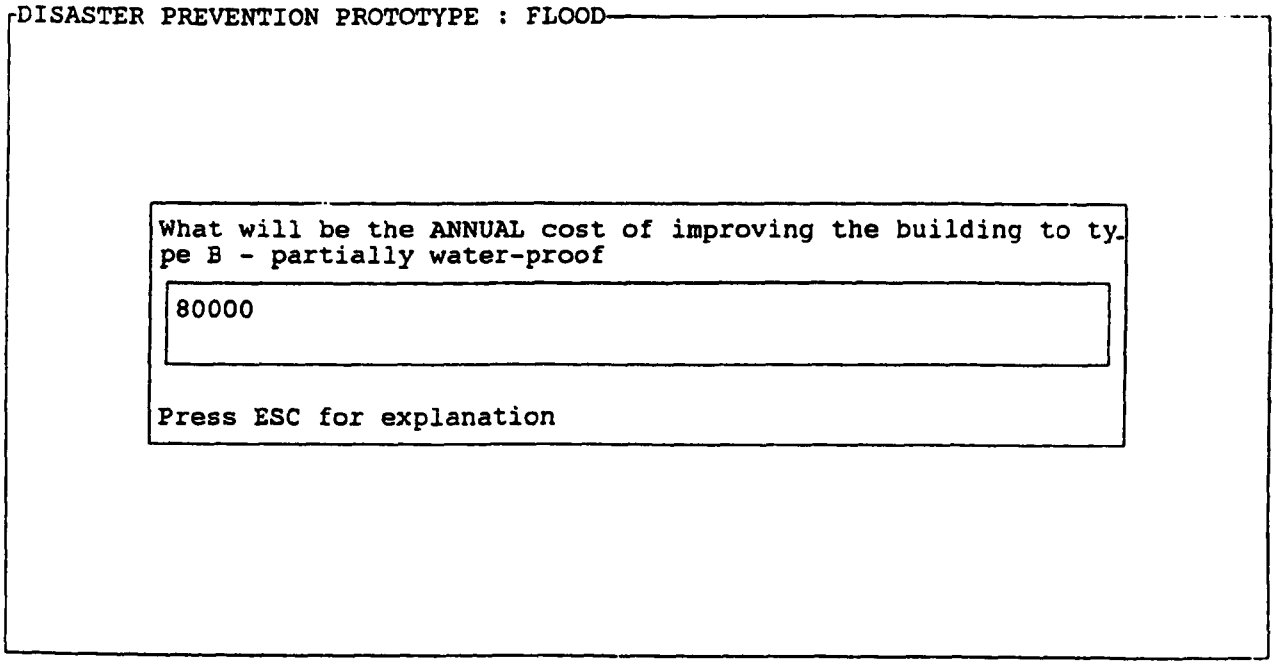
DISASTER PREVENTION PROTOTYPE : FLOOD

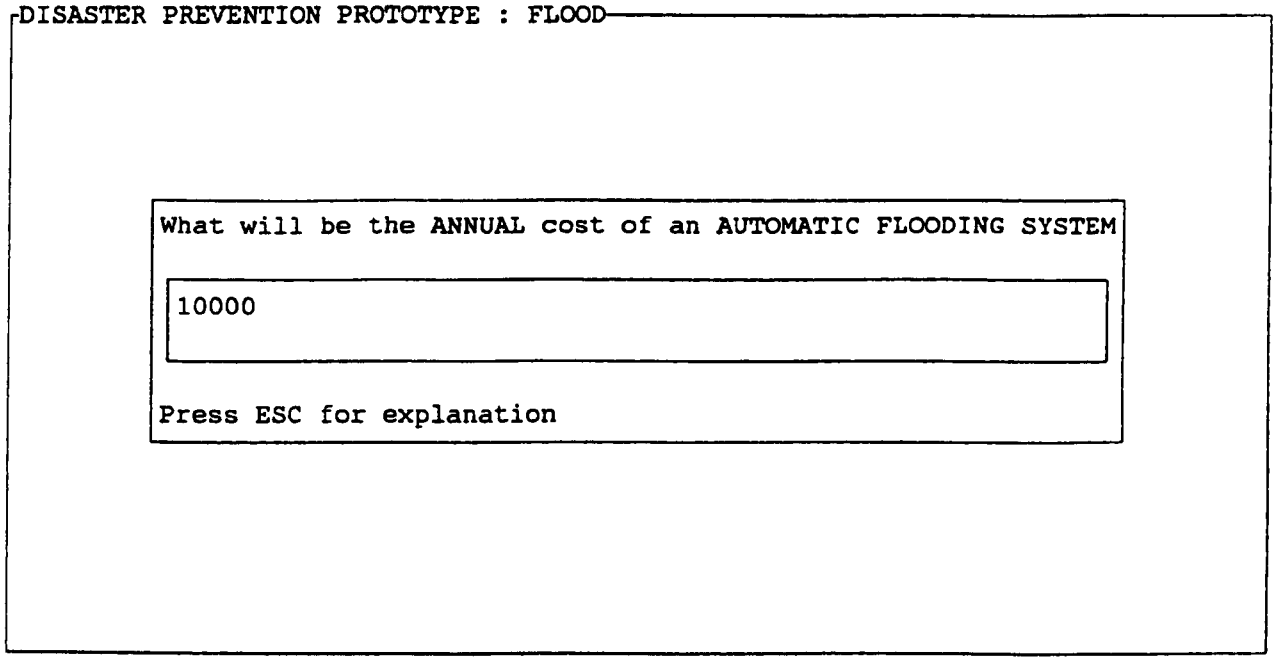
What will be the ANNUAL cost of ELEVATING the building above the Regulatory Flood Datum

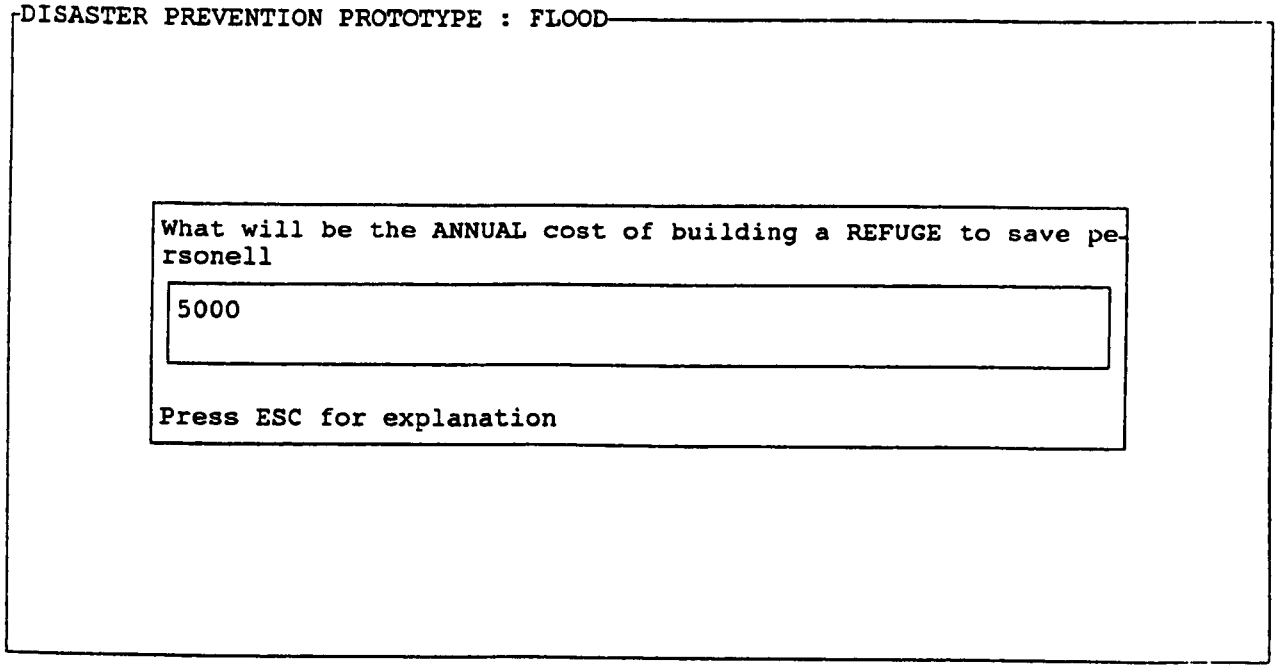
70000

Press ESC for explanation


DISASTER PREVENTION PROTOTYPE : FLOOD
What will be the ANNUAL cost of improving the building to ty_
pe A - completely water-proof
100000
Press ESC for explanation


DISASTER PREVENTION PROTOTYPE : FLOOD
What will be the ANNUAL cost of improving the building to ty_
pe B - partially water-proof
80000
Press ESC for explanation


DISASTER PREVENTION PROTOTYPE : FLOOD
What will be the ANNUAL cost of an AUTOMATIC FLOODING SYSTEM
10000
Press ESC for explanation


DISASTER PREVENTION PROTOTYPE : FLOOD
What will be the ANNUAL cost of building a REFUGE to save personell
5000
Press ESC for explanation

DISASTER PREVENTION PROTOTYPE : FLOOD

What will be the cost of locating reserve POWER above the Regulatory Flood Datum

2000

Press ESC for explanation

DISASTER PREVENTION PROTOTYPE : FLOOD

What will be the ANNUAL cost of installing ACCESS above the Regulatory Flood Datum

2500

Press ESC for explanation

DISASTER PREVENTION PROTOTYPE : FLOOD

What will be the ANNUAL cost of ELEVATING the outside assets above the Regulatory Flood Datum

3000

Press ESC for explanation

DISASTER PREVENTION PROTOTYPE : FLOOD

What will be the ANNUAL cost of raising the COMPUTER ROOM above the Regulatory Flood Datum

1000

Press ESC for explanation

DISASTER PREVENTION PROTOTYPE : FLOOD

What is the annual premium of insurance for loss of service ?

1000000

Press ESC for explanation

DISASTER PREVENTION PROTOTYPE : FLOOD

What is the annual premium of insurance for the dammage ?

150000

Press ESC for explanation

DISASTER PREVENTION PROTOTYPE : FLOOD

What is the annual cost of recovery measures ?

250000

Press ESC for explanation

DISASTER PREVENTION PROTOTYPE : FLOOD

Comb property and service insurances = 1150000

Comb property insurance and recovery costs = 400000

The total annual control costs are : 322500

The recomendation is to implement WCS option 1, cost = 58000

This includes : [barrier,power,refuge,computer]

CONTINUE
yes no

Appendix C

LPA *flex*™ toolkit

What is flex?

flex is an expressive and productive hybrid expert system toolkit mixing:

- frame-based programming
- data-driven programming
- rule based programming

These are fully integrated within a logic programming environment and supplemented with the additional functionality of a declarative AI language to provide a powerful and versatile expert systems development toolkit.

Who is flex for?

flex is aimed at three distinct groups of expert systems developers:

- application programmers who want a shell with a Natural Language
- AI programmers who want an open language-based environment
- Prolog programmers who need extra functionality and structures

On what machines is flex available?

flex programs are 100% portable across a wide range of hardware platforms, from vanilla micros (including 8088 PCs with 640K and 1M MacPlus) and workstations through to departmental machines and mainframes.

What structures does flex use?

The key structures in flex include:

- slot-based frames and instances to store data and facts
- rules, rulesets and relations to encode knowledge and expertise
- functions and procedures for defining imperative processes
- questions and groups for generating end user dialogues

What language does flex use?

flex has its own quasi-English Knowledge Specification Language (KSL) for defining rules, frames and procedures. This language is very expressive and both easy-to-program and easy-to-read. KSL enables developers to write simple and concise statements about the expert's world which can then be understood and maintained by non-programmers.

What about other programming languages?

flex has gateways to most other languages. You can import procedures implemented in those languages using the interfaces to C, Pascal and Prolog.

What about interfaces to databases?

flex can access conventional data through various database interfaces. These are machine dependant. On the PC, for example, there are interfaces into DBaseIII and UNIX interfaces to Oracle , Sybase.

What about the development environment?

This is machine dependant but includes high-level access to windows, scrolling menus, dialogues and graphics with special frame-browsing facilities, source level debugging and an intelligent syntax analyser.

What about performance?

In flex, rules and frames are incrementally compiled, giving the behaviour of an interpreted system with the speed of execution of a compiled one.

Is flex an open or closed system?

flex is an open or soft system. flex allows developers to augment the default behaviour with user-defined routines, either in flex or Prolog. This allows developers to extend or customise flex in a high-level and declarative manner.

What are frames and slots?

Frames are analogous to records in databases and consist of any number of attributes or slots (similar to fields). Frames provide a convenient way of storing related pieces of knowledge and/or data. Frames are far more powerful than conventional records due to their ability to be organised into a frame hierarchy and then have information inherited from frame to frame through the hierarchy. Inheritance avoids unnecessary duplication of data, simplifies code and provides a more readable and maintainable system.

What sort of inheritance does flex support?

flex has very powerful inheritance facilities. It supports single, multiple, specialised and negative inheritance. Multiple inheritance allows frames to inherit multiple values for an attribute, specialised inheritance refers to the ability to override normal inheritance and inherit from any arbitrary slot in any arbitrary frame and negative inheritance to suppress normal inheritance. With flex you can inherit different attributes from different frames to represent those 'exceptional cases' that seem to proliferate in the real world. There are several; variants for the search algorithm to be used in inheritance, i.e. depth-first vs breadth-first, variable bound to level search, and others.

What is data-driven programming?

In flex you can attach data-driven procedures (demons, constraints and watchdogs) to slots, or frames (launches). These procedures will then be automatically executed whenever that slot's value is updated or accessed, or a new sub-frame is created. This technique, often referred to as procedural attachment, enables applications to automatically react to a change in state in a manner similar to OOPS. Demons can be attached to a specific slot of a specific frame or to a specific slot in any frame.

What is rule-based programming?

Knowledge and expertise can often be expressed as 'if-then' rules, where the 'if' part contains the preconditions and the 'then' part the action or conclusion.

How are rules linked?

Rules are then linked or chained together by an inference or 'knowledge' engine which matches the conditions of one rule to the conclusions of another. This engine can either work forwards or backwards. Forward-chaining (data-driven reasoning) is where given an initial database of facts, you match facts with the conditions of a rule and then add that rule's conclusion to the database. Backward chaining (goal-driven reasoning) is where you try to prove the conclusion of a rule by showing its sub-conditions can be satisfied.

How are rules chosen?

When all the conditions of a rule are satisfied, the rule is said to be triggered and ready to fire. Where there are many rules triggered it is often necessary to use a conflict-resolution scoring scheme (crss) to decide which rule should be fired (or else just use the first one). In flex, you can attach static or dynamic 'weights' to each rule with a score clause, and then use the built-in crss mechanism with/without a threshold value, or define your own crss scheme.

How are rules maintained?

The basic flex cycle is to select a rule from the agenda which can be triggered, fire this rule and then re-order the agenda, ready for the next cycle. Agenda manipulation is very versatile and there are various built-in algorithms, or you can define your own with rules to reason about rule agendas, i.e. meta-rules.

How are rules organised?

Large rule bases can be grouped into rule-sets and dynamically loaded into or removed from the agenda. This helps debugging and maintenance and can be used to implement efficient control strategies.

What about debugging facilities?

Prolog has its own interactive source-level symbolic debugger and this is extended in flex to the tracing of slot-updates and rule firing.

What is KSL?

The KSL syntax for rules, frames and procedures is rich and English-like making the knowledge base virtually self-documenting. The KSL enables you to express knowledge and facts in a natural and simple way. Furthermore, KSL is extendible through synonyms and templates. Mathematical, boolean and conditional expressions and functions are available along with set abstractions. Being based on a programming language enables KSL to support both logical and global variables. By supporting variables in rules, flex avoids unnecessary rule duplication and needs fewer rules than most expert systems.

What is Prolog?

Prolog is a symbolic programming language based on logic. Developed in the 70's in Europe, it was adopted by the Japanese as the core language for their 5th Generation Research Program. Prolog is a type free programming language with dynamic data structures and automatic garbage collection. This frees the programmer from the implementation of data.

Prolog has its own built-in inference engine and Prolog procedures (called relations) can be read as backward-chaining rules. This has lead to Prolog's wide usage in the realm of knowledge-based and expert systems.

Prolog supports recursion and has a comprehensive set of list processing routines. Prolog has a fast pattern matcher to help solve symbolic problems with automatic backtracking for searching large solution spaces.

What about user interaction?

Expert systems typically require a lot of user input. flex contains its own complete question and answer system based on menu-driven dialogue screens. Rules and questions can have explanations attached as because clauses which are then used whenever the user asks why a question is being asked or why a particular rule was chosen. There are also file browsing routines so that explanations can be extracted from large pre-structured text files.

Appendix D

Glossary of KBS Terms

Artificial Intelligence (AI) — The discipline devoted to make computers perform tasks which would require intelligence if done by a human being.

Backward chaining — an inference technique used in expert systems sometimes known as goal-driven reasoning, is a technique of selecting a goal which one wishes to establish and trying to prove or disprove it by working back through a succession of sub-goals towards original data.

Database — a collection of independent or inter-related data values stored together.

Declarative — a form which states the properties of a result rather than the way in which it will be obtained, i.e. what, not how.

Domain knowledge — expertise about a particular field.

Entity relationship — a way of modelling real world objects and their interactions.

Execute — to carry out an instruction or program.

Expert — a specialist with skill and knowledge.

Expert system (ES) — programs allowing computers to access and apply human expertise.

Expert system shell — a set of programs capable of forming an expert system when loaded with the relevant knowledge.

Fact — an assertion of truth about an object. A fact is always terminated by a period. A prolog program consists of facts and rules.

Forward chaining — an inference technique used in expert systems sometimes referred to as data driven reasoning, starts with a set of known data to which relevant rules in the knowledge base are applied. The result is a new set of data to which the knowledge base is applied again and so on.

Frames — a method of representing knowledge as a data object in which all attributes of the frame are treated as a unit but can inherit properties from associated higher-level frames.

Heuristics — methods of problem solving through successive trial and error attempts at a solution rather than by predetermined algorithm. Simple rules of thumb which are used to derive a conclusion from sometimes incomplete or uncertain evidence. They have no guarantee of success but sometimes prove useful.

Inference — the formal process of reasoning which searches a knowledge base to draw conclusions based on the evidence of the case being considered.

Interface — the component by which the user interacts with the system.

Knowledge — Expertise, practical skill and learning.

Knowledge acquisition — the process of identifying, collecting and refining knowledge. This may require interviews with experts, research in a library or personal

experience. The person undertaking the knowledge acquisition must convert the acquired knowledge into a form that can be used by a computer program.

Knowledge base — a collection of facts, inferences and procedures, corresponding to the types of information needed for problem solution.

Knowledge-based system — a computer system designed to emulate human expertise in some specific domain. Typically it should possess a knowledge base of facts, rules and heuristics about that domain. This computer system should also have a capability for engaging in an interactive consultation with its user as a human expert might. In addition it should be able to show the user how a solution was reached.

Knowledge engineer — the person who specialises in assessing problems, acquiring knowledge and building knowledge-based systems. This implies training in cognitive science, computer science and artificial intelligence. It also suggests experience in the development in one or more knowledge-based systems.

Knowledge representation — the method used to encode and store facts and relationships between them in a knowledge base. Production rules, frames and semantic networks are all ways to represent knowledge.

LISP (LISt Processing) — a language which processes programs and data in the form of chained lists.

Production rule — a modular knowledge structure representing a single chunk of knowledge, in If-Then or Antecedant-Consequent form. Common in expert systems.

Prolog — stands for programming in logic, a logic oriented artificial intelligence language developed in France and popular in Europe and Japan.

Prototype — an initial version of a system that is used as a base for constructing the future system.

Rule — a way of representing knowledge. In the expert system's set of rules, each rule relates a set of Conditions to a set of associated Actions. When the Conditions are found to be true, the appropriate actions are initiated.

Rule-based system — rules in If-Then or Antecedant-Consequent form are a simple and common representation of knowledge. A system which holds knowledge in this form is called a rule-based system or production system.