# EVIDENTIARY TREATMENT OF COMPUTER-PRODUCED MATERIAL: A RELIABILITY BASED EVALUATION

Cameron Spenceley

A thesis submitted for the degree of
Doctor of Philosophy

University of Sydney
2003

# Abstract

The law of evidence imposes conditions upon how and when material may be used in legal fact finding, which is a process by which facts are ascertained for the purposes of facilitating the application of substantive law. The conditions which are applied by the law of evidence to the use of material of a given kind can be said to give rise to a regime of evidentiary treatment for that material. Computer-produced material is a kind of material that may be subject to different regimes of evidentiary treatment. Different approaches to the evidentiary treatment of computer-produced material have been advocated in a variety of contexts. Computer-produced material is an important source of potential evidence because computers are used to store, process and produce information on a widespread basis.

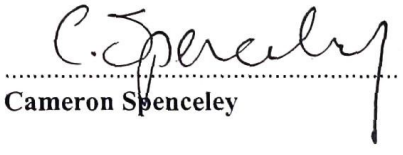
The purpose of this study is to evaluate the principal approaches to the evidentiary treatment of computer-produced material in common law jurisdictions. This evaluation is premised upon the contention that evidentiary treatment should promote the goals of legal fact finding. A primary goal of legal fact finding is the identification of truth by rational means. The accuracy of the information that may be contained in computer-produced material is therefore an issue which any approach to evidentiary treatment must address. More importantly, the issue must be addressed in a manner that is congruent with the objective of rational truth identification.

Computer-produced material is properly viewed as the artefact of a process of information transformation. The accuracy of information contained in that material depends upon the information that is provided to a computer as input and upon how that information is dealt with. Any concern about the accuracy of the output translates to a concern about the accuracy of input *and* a concern about whether the computer has operated correctly in dealing with that input. In this context 'correct' operation means operation that conforms to the parameters of some task that the computer is expected to perform. Whilst considerable attention is given in the literature to the question of accuracy of input, there has been little serious scrutiny of the question of correct operation.

Approaches to evidentiary treatment are evaluated by reference to the extent to which they address the issue of correct operation in a manner that exhibits a rational foundation. Relevant knowledge and principles from the fields of computer science and reliability engineering are applied in this evaluation. It is concluded that each of the approaches considered suffers from significant shortcomings. An alternative approach to evidentiary treatment that seeks to remedy these shortcomings is presented.

# Declaration of Originality

To the best of my knowledge, this thesis contains no copy, paraphrase or summation of the published or unpublished work of any other person, except where duly acknowledged in the text. This thesis contains no material that has been previously presented for a degree at the University of Sydney or at any other university.



**Cameron Spenceley**

# Acknowledgments

I would like to acknowledge the support and assistance of the following people in connection with my candidature for this thesis. I am most grateful in particular to my supervisor, Professor Les McCrimmon and to the Director of Research at the Sydney Law School, Professor Terry Carney. Without their generous advice, guidance, support and counsel, the thesis would not have come to fruition.

I am also very grateful to those outside the Sydney Law School who took time from their own busy schedules to discuss aspects of my research and to provide me with their advice and suggestions. From the University of Sydney: Professor Peter Eades, head of the School of Information Technologies and Professor Malcolm Quine of the School of Mathematics and Statistics. From the University of New South Wales: Dr Ray Eaton of the School of Electrical Engineering and Telecommunications.

I would also like to thank the staff of the Law, Fisher, Engineering, Badham and Mathematics libraries at the University of Sydney, and the staff of the University of New South Wales libraries, for the time which they devoted to my research needs. Their help was invaluable to me.

# CONTENTS

# DETAILED TABLE OF CONTENTS

# TABLE OF CASES

# TABLE OF STATUTES

## AUSTRALIA

### Commonwealth



[‡] Unless a contrary indication is given in the thesis, references to sections in the *Evidence Act* 1995 (Cth) are also references to the equivalent sections in the *Evidence Act* 1995 (NSW) and the *Evidence Act* 2001 (Tas). When a reference is made to a provision in either of the latter two Acts, the reference is to the provision in that Act only.

**New South Wales**



**Queensland**

## South Australia



## Tasmania



## Victoria



## UNITED KINGDOM

## UNITED STATES OF AMERICA



## ISRAEL



## SOUTH AFRICA

# 1. Introduction

## 1.1 Background

### 1.1.1 Law, legal fact finding and 'evidentiary treatment'

This research deals with an aspect of the activity that can be called 'legal fact finding'. Legal fact finding is the process by which facts are ascertained for the purposes of facilitating the application of law. Legal fact finding is an important process because a basic function of law is the attribution to particular facts of significance within a normative social schema.

The application and enforcement of law depends not only upon the content of the law but also upon the ascertainment of the facts to which it is to apply. Particular laws may aim to secure specific social, cultural or economic objectives. They may do so very well or very poorly. How and when such laws apply is determined by the existence of facts that correspond with the subject matter that they purport to govern. In this sense, facts are crucial to the enterprise of law.[1]

---

[1] For a related discussion that emphasises the failure of traditional legal scholarship to embrace the importance of facts and factual analyses see: Twining W L "Taking Facts Seriously" in Twining W L, *Rethinking Evidence: Exploratory Essays* (Oxford: Blackwell, 1990) 12-31.

Legal fact finding is most often and most visibly carried on in common law jurisdictions within the context of adversarial litigation. Adversarial litigation is conducted in a non-cooperative environment in the sense that

> [t]he common law courtroom trial is a forum in which arguments of the disputing parties are pitted against each other. ... The defining characteristic of adjudication in common law systems is its adversarial nature, reflected in the practice and culture of litigation.[2]

Further, it involves parties who will necessarily have different interests as to its outcome. At the same time, the parties play an active role in determining the course that the litigation will take and the scope of the factual dispute upon which it is focused.

> [P]rocedural action is controlled by the parties and the adjudicator remains essentially passive. In the fact-finding domain, this implies that the litigants and their counsel decide what facts shall be subject to proof.[3]

These matters suggest that the process of litigation cannot depend upon the parties reaching agreement as to the manner in which it is to be conducted. Adversarial litigation is a mechanism by which disputes may be determined, but it gives rise to the recursive consideration that the operation of the mechanism may itself be the subject of disputes. Such a mechanism must therefore be capable of determining these *procedural* disputes. This capacity is delivered through 'procedural law', which includes the law of evidence. As Damaska argues, procedural law is ancillary to the substantive laws around which principal disputes that give rise to litigation are concentrated. He suggests that "procedure is basically a handmaiden of the substantive law."[4]

---

2 Australian Law Reform Commission, *Issues Paper 20: Review of the Adversarial System of Litigation* (Canberra: Australian Government Publishing Service, 1997) paragraph 2.9.

3 Damaska M R, *Evidence Law Adrift* (New Haven: Yale University Press, 1997) 74.

4 Damaska M R, *The Faces of Justice and State Authority* (New Haven: Yale University Press, 1986) 148.

Procedural law dictates what parties to litigation must, may, and may not, do. In doing this, it provides a basis for the determination of disputes about how a given instance of litigation is to proceed. The 'law of evidence' is part of the procedural law. Although procedural law is readily distinguished from substantive law for its ancillary character, determining what part of the procedural law belongs merely to 'procedure' and what belongs to 'evidence' is a difficult matter.[5] It is clear that regulation in either area can have an influence upon legal fact finding. For example, the law relating to the availability and use of subpoenas might properly be regarded as part of the law of 'procedure' rather than the law of evidence.[6] Even so, subpoenas are an important tool for enabling wide-ranging information gathering and for ensuring that information can be made available to the fact finder. Further, as the Australian experience[7] has demonstrated, important principles such as privilege can be common to both pre-trial processes and what takes place in court.

It may in any event be that efforts to erect a boundary that distinguishes evidence law from other parts of the procedural law will not be fruitful in all contexts.[8] For this thesis, the precise classification of particular aspects of procedural law is not as important as the examination of how those aspects influence legal fact finding.

---

5 For details of one attempt that led to an unexpected result see Australian Law Reform Commission, Report 26, *Evidence (Interim)* (Canberra: Australian Government Publishing Service, 1985) Volume 1, paragraphs 27-47. The commission had recommended the enactment of a new Commonwealth Evidence Act that provided for a test for legal professional privilege that departed from the prevailing common law test in Australia, which was the 'sole purpose' test established by the High Court of Australia in *Grant v Downs* (1976) 135 CLR 674. The commission recommended the use of 'dominant purpose' test for privilege. This was the test for privilege that was enacted in the *Evidence Act* 1995 (Cth). Because of the way in which the commission had attempted to divide the law of evidence and the law of procedure, the new test was intended to apply only to material that was to be used as evidence in court. This left the application of the test for other purposes to be determined by the common law. In *Esso Australia Resources Limited v Commissioner of Taxation* (2000) 201 CLR 49, a majority of the High Court of Australia were of the view that the resulting distinction between the test for the availability of the privilege in respect of material that was to be used for pre-trial processes (such as that obtained through subpoenas and discovery) and the test for the availability of the privilege in respect of material that was to be used as evidence at trial created an unacceptable anomaly (at 73 per Gleeson CJ, Gaudron and Gummow JJ and at 105 per Callinan J). The Court overruled the earlier decision in *Grant v Downs* and so altered the common law relating to the privilege for all purposes in Australia. This change applied even in the majority of Australian jurisdictions which had not adopted the Evidence Act reforms.

6 For one such characterisation see: Australian Law Reform Commission, Note 5 at paragraphs 41 and 46.

7 Note 5.

8 For a further discussion of this point see Jolowicz J A, *On Civil Procedure* (Cambridge: Cambridge University Press, 2000) 60.

Within this setting, it is apparent that the law of evidence deals with a single, yet vitally important aspect of what takes place in court, namely the regulation of the kind of material that can be used by the parties to influence the course of legal fact finding.

The law of evidence is largely, but not exclusively, characterised by a number of restrictions upon the material that the parties may offer to the fact finder to persuade it to accept the factual contentions that each wishes to propound. It has been said that these restrictions operate as a limitation upon the 'freedom of proof'[9] that the parties would otherwise enjoy. Damaska argues that such restrictions are a direct response to the non-cooperative nature of the conduct of litigation. In his view, they are "first and foremost the child of the adversary system."[10]

1.1.1.1 The goals of legal fact finding

The foregoing considerations indicate a need for a law of evidence that imposes restrictions upon the means of proof that parties may use, but they do not indicate what the ultimate purpose of that law should be. The law of evidence is not concerned merely to provide a basis for the determination of disputes about the use of material. It is also the mechanism by which the goals of legal fact finding may be pursued. Principal[11] amongst these goals is the identification of truth.

> Truth finding must be a central purpose, whatever the tribunal. Unless we are to assume that the substantive law is perverse or irrelevant to the public welfare, then its enforcement is properly the aim of litigation: and the substantive law can best be enforced if litigation results in accurate determinations of facts made material by the applicable rule of law. Unless reasonably accurate fact-finding is assumed, there does not appear to be any sound basis for our judicial system.[12]

---

9 The term is used by Twining: Twining W L, "What is the Law of Evidence" in Twining, Note 1, 178-218 at 178. Twining suggests that "[f]ree proof" means an absence of formal rules that interfere with free enquiry and natural or commonsense reasoning (at 194).

10 Damaska, Note 3 at 2.

11 As is discussed in section 1.2.3, there are other goals such as economy, expedition and a broad notion of 'fairness'. The latter accounts for the rules of evidence, such as rules relating to privilege, that exclude material from use in fact finding on grounds that are not related to its probative value.

12 Weinstein J B, "Some Difficulties in Devising Rules for Determining Truth in Judicial Trials" (1966) 66(2) *Columbia Law Review* 223-246 at 243.

Wigmore argued in this context that "all of the fundamental rules have some reason underneath. They are not arbitrary. Their aim is to get at the truth by calm and careful reasoning."[13] That truth identification is and should be centrally important in the law of evidence is not a proposition that has been the subject of serious question.

It is not just historical scholarship that supports the proposition that truth identification is an important goal of legal fact finding. The importance of this goal is reflected in contemporary debates about what material should or should not be available to fact finders. Those debates invariably proceed on the basis of the premise that truth identification is the desired outcome of litigation. As it happens, this premise is barely mentioned because it is not the point in issue. What is in dispute is not whether fact finding should seek the truth, but what materials and methods will best aid its search.

Debates about the forensic application of deoxyribonucleic acid (DNA) examination technology comprise a contemporary illustration of these concerns. DNA is a complex molecule that is found in all human cells. It consists of four basic building blocks, or 'nucleotides', arranged in a double stranded linear sequence.[14] The order in which the nucleotides appear can and does vary between individuals. The DNA

---

[13] Wigmore J H, *A Students' Textbook of the Law of Evidence* (Brooklyn: Foundation Press, 1935) 5. See also Fox R W, "Expediency and Truth Finding in the Modern Law of Evidence" in Campbell E and Waller L (eds) *Well and Truly Tried* (Sydney: Law Book Company, 1982) 140-176 at 141; Stone J, Wells W A N (ed) *Evidence: Its History and Policies* (North Ryde, New South Wales: Butterworths, 1991) 59: "The ultimate purpose of the law of evidence today is to ensure that the facts found, to which the court is to apply the rules of substantive law, are more likely to be true than false."; Damaska Note, 3 at 76; Twining W L, "The Rationalist Tradition of Evidence Scholarship" in Twining, Note 1, 32-91 at 78.

[14] Each nucleotide consists of one of four chemical bases - adenine, cytosine, guanine or thymine and a sugar-phosphate bond. It is customary to distinguish the four nucleotides by the initial letters 'A', 'C', 'G' or 'T': Brown T A, *Genetics: A Molecular Approach* (London: Chapman & Hall, 3rd ed, 1998) 24-26. The nucleotides form pairings (always A-T and C-G) at each matching point along the double stranded sequence. These are commonly called 'base pairs'. The length of a DNA molecule is usually expressed as a number of base pairs: Brown at 36. See also Nagylaki T, *Introduction to Theoretical Population Genetics* (New York: Springer-Verlag, 1992) 1.

molecule is typically long enough[15] to permit a very large number of unique sequences. It is said that DNA complement of one person is entirely unique.[16]

The value of testing that can reveal even portions of the DNA sequence of an individual[17] is said to arise from the possibility of establishing matches between DNA material taken from a known person, such as a defendant in a criminal case, and DNA material left by an unknown individual that is found at some place of interest, such as the scene of a crime. It is said that 'matches'[18] in pattern[19] for several distinct portions[20] of a DNA molecule indicate an extremely remote possibility that the samples for which the matches are obtained do not come from the same individual.[21] In legal proceedings in which the identification of an unknown individual who is associated with some time and place is in issue, the results of forensic DNA testing are said to be a powerful indicator of truth that should be made available to decision makers.[22]

Despite these apparent virtues, forensic DNA testing has had, and continues to have, its critics. Concerns about the accuracy of testing procedures, and about the validity of the statistical assumptions that are presented with test results to explain their

---

15 The human chromosomes are, for example, between 55 and 250 million base pairs long and the total number length for all chromosomes is about 3 billion base pairs: Brown, Note 14 at 278. Even though 99.5% of all human genetic material is the same: Inman K and Rudin N, *An Introduction to Forensic DNA Analysis* (New York: CRC Press, 1997) 29, the remaining portion (0.5%) exhibits more than enough variability to be useful for forensic purposes that require discrimination between individuals.

16 Except in the case of identical twins: Krawczak M and Scmidtke J, *DNA Fingerprinting* (Oxford: BIOS Scientific Publishers, 1994) at 61.

17 Which is the current limit of the technology: Inman and Rudin, Note 15 at 87 and also at 180, showing 28 of the portions for which forensic testing procedures exist.

18 In this context, exact 'matches' are not possible because of the limitations of the measurement techniques. Two samples will be said to match when they are sufficiently similar according to some accepted criterion. For an example, see Krawczak and Scmidtke, Note 16 at 70.

19 The particular pattern at a given location is called an 'allele': Brown, Note 14 at 427.

20 A point in the DNA molecule at which an allele occurs is called a 'locus': Brown, Note 14 at 442. The definition is a little misleading because a locus is not a single discrete point, but rather it is the space or range within the molecule that the allele occupies.

21 For example Krawczak and Scmidtke, Note 16 at 63; Inman and Rudin, Note 15 at 91-94.

22 See for example: Inman and Rudin, Note 15 at 1, 152 and 157-158; Krawczak and Scmidtke, Note 16 at 61 and National Academy of Sciences, *The Evaluation of Forensic DNA Evidence*. (Washington: National Academy Press, 1996) at 2.

significance, have contributed to intense debates – dubbed the 'DNA wars'.[23] What these debates reflected was a shared concern for aiding truth-finding efforts in litigation. The disputes were not about *what* the goal was, but how best to achieve it.

> [S]cientists on both sides of the DNA debate sometimes wrote with passion because as scientists they are ordinarily passionate about getting things right, and never more so than when core values like lives and justice are at stake.[24]

An important adjunct to the goal of truth identification is the dictate that this goal should be pursued by *rational* means. As Twining's historical account demonstrates, this latter contention is fundamental to contemporary thinking about the role of evidence law.[25] Twining argues in particular that

> [d]espite the strains and disagreements there is a truly remarkable homogeneity about the basic assumptions of almost all specialist writings on evidence... Almost without exception Anglo-American writers about evidence share very similar assumptions, either explicitly or implicitly, about the nature and ends of adjudication...[26]

1.1.1.2 A basis for evaluating the law of evidence

The articulation of goals of legal fact finding serves an important purpose in the context of this thesis. It provides a basis for the evaluation of the law of evidence. It may, for instance, be asked whether a particular rule of evidence promotes or hinders the realization of the primary goal of rational truth identification. The same question may be also be asked of the way in which the law of evidence deals with a particular kind of material.[27] It is this latter question with which this thesis deals. What the

---

23 Krawczak and Schmidtke, Note 16 at 73-77 citing Lewontin R C and Hartl D L, "Population Genetics in Forensic DNA Typing" (1991) 254 *Science* 1745-1750 (one of the more prominent pieces in which the challenges to statistical assumptions were made). For a comprehensive review that highlights the lingering concerns see: Lempert R, "After the DNA Wars: A Mopping Up Operation" (1997) 31 *Israel Law Review* 536-572.

24 Lempert, Note 23 at 538.

25 Twining, Note 13 at 71-72.

26 Twining, Note 13 at 71.

27 The term 'material' is used in this thesis to describe something that a party might seek to use in litigation as a means of proof, whether or not it is or would be admissible under existing or proposed

thesis is concerned to examine and to evaluate is the 'evidentiary treatment' of particular material.

The term 'evidentiary treatment' refers to how, and under what conditions, the law of evidence material permits material to be used in legal fact finding. Material may be excluded from use in all circumstances, or only in some. It may be admitted under certain conditions by force of a specific rule of evidence, despite the existence of rules that would otherwise exclude it. In this context, rules of evidence are analysed collectively, rather than individually. It can be said that certain rules of evidence will give rise to an 'expression' of a particular approach to the evidentiary treatment of material of a particular kind. This thesis examines the effect of the approaches to evidentiary treatment, and of their particular expressions, rather than the effect of individual rules of evidence.

### 1.1.2 Computer-produced material

#### 1.1.2.1 Computers

This thesis deals with the evidentiary treatment of a particular kind of material, namely that which has been produced by a computer. The term 'computer' refers to the kind of device that is encompassed by a standard dictionary definition. By one such definition, a computer is

> an electronic device, usually digital, for storing and processing data (usually in binary form), according to instructions given to it in a variable program.[28]

---

rules of evidence. The term 'evidence' is reserved for material that has been admitted under the rules of evidence that apply in a given proceeding.

[28] Moore B (ed), *The Australian Oxford Dictionary* (Melbourne: Oxford University Press, 1999) 275. Strictly speaking, a computer need not use electronic components but because this is the universal form in which modern computers exist, this thesis is limited to consideration of this kind of computer. For further details see generally Lucas H C, *Introduction to Computers and Information Systems* (New York: Macmillan: 1986) 51-57. As to the early implementations of the computer in purely mechanical forms see Lucas at 46-50; Boyce J C, *Digital Logic and Switching Circuits: Operation and Analysis* (Englewood Cliffs, New Jersey: Prentice-Hall, 1975) 2-3; Kurzwell R, "When Will HAL Understand What We Are Saying? Computer Speech Recognition and Understanding" in Stork D G (ed), *HAL's Legacy: 2001's Computer as Dream and Reality* (Cambridge, Massachusetts: MIT Press, 1997) 131-169 at 146-147.

A computer comprises at least two distinct elements: the tangible device that executes a program of instructions, which is the 'hardware' and the program of instructions itself, which is the 'software'.[29] These are, in a strict sense, the only elements that must be engaged in order to facilitate the production of material by a computer. A scenario for the operation of a computer that is restricted to these two elements entails, however, a significant limitation. A single program of instructions can do no more than cause the production of the same specific material that its design contemplates. This means that in the absence of some variational influence, the same program will cause the same material to be produced every time that it is executed.

This limitation is removed when the program is prepared in a way that enables it to accept the provision of variable input information when it is executed and to respond to that input information in a distinct way. If a program can respond differentially to variable input information, then its usefulness will far exceed that of a simple program that possesses no such capacity. As common experience indicates, virtually all software operates in this way. In most cases, the capacity to accept input information and to respond to it differentially is essential to the intended function of the program. Programs that provide word processing, electronic mail and information storage and retrieval functionalities are familiar examples of such software.

The operation of a computer therefore almost invariably involves three fundamental elements: hardware, software and input information.[30] The roles of these elements can be stated as follows. Hardware executes the instructions that comprise the software. These instructions facilitate the acceptance of some input information and invoke in the hardware a particular response that will often be different for different

---

[29] For a discussion of the roles of the two elements see for example Lucas H C, *Introduction to Computers and Information Systems* (New York: Macmillan, 1986) 79 and Lee G, *From Hardware to Software* (London: Macmillan, 1982) 210.

[30] At least one definition of the term 'software' contemplates that this term may include input information as well. It refers to "computer programs, procedures and possibly associated documentation and data pertaining to the operation of a computer system": Institute of Electrical and Electronics Engineers, *The IEEE Standard Dictionary of Electrical and Electronic Terms* (New York: Institute of Electrical and Electronics Engineers, 6th ed, 1996) 1006.

inputs. As a result, a given program of instructions will, when combined with distinct input information, cause the hardware to operate in a particular and distinct way.

1.1.2.2 Computer-produced material

The specific interest of this thesis is with what is referred to here as 'computer-produced material'. This term defines material that has been produced by a computer by the means indicated in section 1.1.2.1 and which contains information that is perceivable by a human fact finder.[31] It includes tangible material, such as printed pages and intangible material, such as the images that appear on a display device. The term excludes material that contains information that cannot be perceived by a human fact finder, such as information that is represented only by electronic or magnetic quantities.[32] The rationale for this limitation is that such information has no potential for use unless and until it is converted to some perceivable form. It is unproductive to consider any characteristics of given information, such as its accuracy, until that information is in a form in which those characteristics can matter.

Computer-produced material is a source of information. More than this, computers are sufficiently ubiquitous that the information that they store, process and produce has a potential to be relevant to a wide range of instances in which litigation may arise. The importance of computers is a matter that requires little emphasis, but which is saliently expressed in the following passages.

> Computerized information systems are at the heart of all modern organizations. Such systems are used as a means of both obtaining competitive advantage and re-engineering the business processes of the organization itself. They have thus become a key component in the success of

---

31 The term computer-produced material is largely synonymous with the term 'computer output'. Although the terms are generally interchangeable, the latter term is used where the subject of discussion is the information that a computer may produce in a general sense, rather than a specifically manifested item of material. The former term is used where the context requires greater emphasis upon a specific item of material (albeit that any such item will also contain information).

32 This does not necessarily exclude information that is expressed as images, symbols, or even sounds.

> virtually every business and as the range of computerized applications grows, so too does their importance.[33]
>
> We can look at virtually any industry—automotive, avionics, oil, telecommunications, banking, semiconductors, pharmaceuticals—all these industries are highly dependant upon computers for their basic functioning.[34]

This potential for the use of computer-produced material as an aid to legal fact finding gives rise to a need to consider the conditions under which such use ought to be permitted. This equates to a need to consider what regime(s) of evidentiary treatment might be appropriate for computer-produced material.

### 1.1.3 The evidentiary treatment of computer-produced material

#### 1.1.3.1 The role of evidentiary treatment

A regime of evidentiary treatment will be manifested whenever a particular kind of material is offered as evidence. This will be the case even where no consideration has been given to the possibility of special treatment for the material in question. A 'default' regime in which a particular kind of material is simply dealt with according to existing rules of evidence is nonetheless a regime of evidentiary treatment. It may be that for some material, such a default regime is entirely appropriate. Alternatively, the law of evidence may provide for a more 'active' approach to evidentiary treatment that may respond to what are regarded as unique characteristics of the material in question.

In all cases, the evaluative framework that was identified in section 1.1.1 remains apposite. Any approach to evidentiary treatment can be evaluated by references to the extent to which it facilitates realisation of the goal of truth identification by rational means. Evidentiary treatment governs the "process of adducing evidence and passing

---

[33] Flowers S, *Software Failure, Management Failure: Amazing Stories and Cautionary Tales* (Chichester: Wiley, 1996) 2.

upon probative value"[35] it must therefore "be based ultimately upon the canons of ordinary reasoning."[36] A reasonable initial expectation in this regard is that an approach to evidentiary treatment will respond to the likelihood that the information that is contained in material of interest will be accurate. This implies that an approach to evidentiary treatment should entail an attempt rationally to distinguish information that is likely to be accurate from information that is not likely to be accurate.

It is necessary to refer in this context to what is likely, as opposed to what is certain, because the accuracy of particular information is not a matter that is ever resolved by a legal fact finding process in any absolute sense. Sources of information contribute to the conclusions that are reached in a particular fact finding exercise, but they are not scrutinised against any supervening indicator of truth. Legal fact finding is directed to the resolution of particular issues that are framed by antecedent procedural processes.[37] It is not concerned to certify directly the accuracy of any given item of information of which it may make use.

To restrict the fact finder to the use of only information that can be demonstrated absolutely to be accurate would not be feasible because it would give rise to an insurmountable circularity. The truth about the issues to which the information related would have to be ascertained to an absolute certainty before that same information could be used to make findings about those same issues. In formulating and evaluating approaches to evidentiary treatment, it is therefore necessary to consider that legal fact finding has to make use of inferences, not merely those matters which can be the subject of conclusions that carry an absolute certainty.[38]

In some cases computer-produced material will reveal obvious signs of inaccuracy on its face. These signs may be as patent as text that is incomprehensible, or figures that

---

[34] Lyu M R "Introduction" in Lyu M R (ed) *Handbook of Software Reliability Engineering* (Los Alamitos: IEEE Computer Society Press: 1996) 1-25 at 3.

[35] Wigmore J H, *Evidence in Trials at Common Law* (Tillers revision) (Boston: Little Brown and Company, 1983) §30.

[36] Note 35.

[37] In civil litigation in common law jurisdictions, the focus of these processes is almost invariably the written pleadings that are submitted to the court by each party.

[38] See generally Twining, Note 13 at 74.

indicate values that fall outside any reasonable limits for the subjects concerned.[39] In these cases of obvious inaccuracy, the material is plainly worthless to a party that would otherwise be disposed to offer it to a fact finder. Of far greater significance are the cases in which the material shows nothing on its face that indicates inaccuracy. It is these cases that present the most significant challenge for any approach to the evidentiary treatment of computer-produced material.

What must be considered in such cases are the factors that affect the accuracy of information that is contained in computer-produced material, as opposed to those which are merely *indicative* of inaccuracy. This consideration is fundamental to any attempt to distinguish material that is likely to be accurate from material that is not likely to be accurate. The factors of interest in this context are those which can influence the content of the material that a computer produces, and these are indicated by an analysis of the manner in which a computer operates.

1.1.3.2 Factors that influence accuracy

A description of the manner in which a computer operates was given in section 1.1.2.1. It is, however, more useful in this introductory discussion to consider the operation of a computer at a higher level of abstraction. At this level it can be said that the operation of a computer realises a process of *information transformation.* Input information that is supplied to the computer is transformed to specific output according to a scheme of processing that is described by the architecture of the relevant software and hardware. The most important feature of this process is that, for particular input, the output that is expected is defined by the process that the hardware and software have been designed to implement.

---

[39] An example would be a reading produced by hospital monitoring equipment that purports to record the temperature of a patient as, say, 500 degrees Celsius.

This analysis is consistent with, for instance, the view that software is "essentially an instrument for transforming a discrete set of inputs into a discrete set of outputs."[40] The process involves the transformation or 'mapping' of particular input information that is selected from some input domain to corresponding outputs in some output domain. A depiction of an information transformation process appears in figure 1-1.

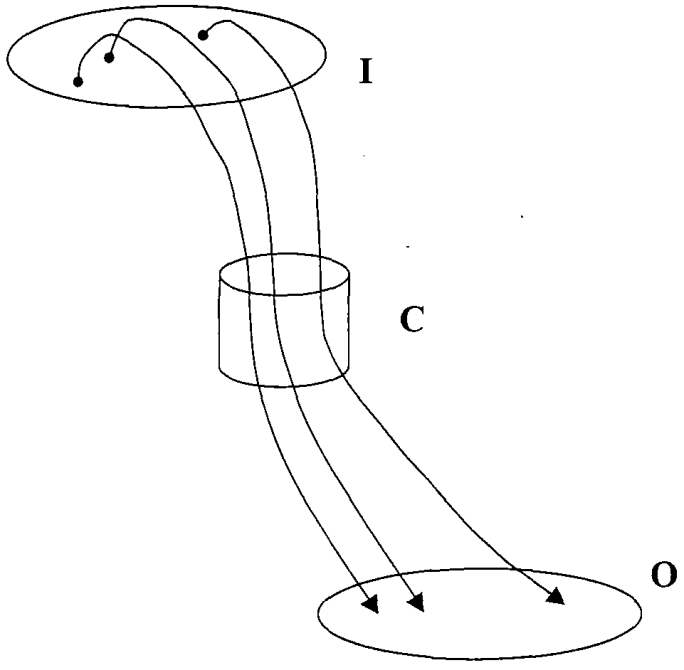



**Figure 1-1: An information transformation process maps specific inputs to specific outputs**[41]

In figure 1-1, the computer C is a combination of some software and hardware. Discrete inputs from the input space I result in (or map to) discrete outputs in the output space O. Each arrow represents a distinct instance of operation of the software

---

[40] Goel A L, "Software Reliability Models: Assumptions, Limitations, and Applicability" (1985) 11 *IEEE Transactions on Software Engineering* 1411-1423, 1411. Strictly speaking it is the combined operation of the hardware and software that realises the process. Goel's (engineering) analysis of what takes place is remarkably similar to the analysis of the United States Court of Appeals for the Federal Circuit in *In re Alappat* 33 F 3d 1526 (Fed Cir, 1994). The *Alappat* court characterised a computerised process that it was considering in connection with a patent claim as one that functioned "to transform one set of data to another through what may be viewed as a series of mathematical calculations" (at 1570).

[41] Diagram adapted from Boehm B W, *Software Engineering Economics* (Englewood Cliffs, New Jersey: Prentice-Hall, 1981) 373, Figure 24-1.

with different input information. Output can—and should—therefore be seen as the *artefact* of the process of information transformation that the computer realises. Its characteristics are those which the process imparts. While the choice of particular input information will also influence the form of the corresponding output, the way in which the computer deals with or 'transforms' that input is at least as influential in this regard.[42]

This account identifies two factors that affect the accuracy of information that is contained in computer-produced material. The first is the accuracy of the input information. For example, a computerised library catalogue system might be used to produce a list of library holdings by the author 'Jones' that were published after 1990. The accuracy of any output from the system is conditioned upon the correct stipulation of these two search parameters (and also upon the accuracy of the underlying catalogue data that are to be queried).

The second factor is the operation of the computer in the sense that what it does with the input information conforms to the task that it is expected to perform. In the example just given, the computer must search all of the recorded library holdings with the parameters that were supplied and it must identify all of the matching records. If the correct input information were supplied but the computer performed, for some reason, a search for library holdings by the author 'Jones' that were published before (rather than after) 1990, then the output would not be accurate despite the accuracy of the input.

---

[42] An information transformation conceptualisation of this kind is used in Boehm, Note 41 at 372-373 and Littlewood B, "Modelling Growth in Software Reliability" in Rook P (ed), *Software Reliability Handbook* (London: Elsevier, 1990) 137-153 at 141. Figure 1-1 involves two important simplifications that require explanation. First, there is the possibility that different inputs may lead to the same output under certain conditions. Second, the depiction of inputs as single points within a closed region also covers the cases in which the input information comes from different sources. A particular program may use many individual pieces of information from different sources as input. These may include previously stored information, or information that is obtained from another computer via some communications infrastructure, such as the Internet. What is provided by a user on a given occasion may only be a small part of the 'input'. This will be the case where, for example, a user provides search or query parameters that are to govern the manner in which some repository of stored information (such as a database) is examined.

1.1.3.3 Attitudes to the factors that influence accuracy

To facilitate realisation of the goal of rational truth identification, an approach to the evidentiary treatment of computer-produced material must address each of the two discrete factors that were mentioned in section 1.1.3.2, namely accuracy of input and correctness of operation. It must consider and respond appropriately to the potential for each of these factors to give rise to inaccuracies in the output. The need to address both accuracy of input and correctness of operation has not, however, been recognised in the existing approaches to the evidentiary treatment of computer-produced material. What is evident instead is a distinct focus upon the accuracy of input information as the sole or overwhelmingly dominant influence over the accuracy of output.

This focus appears in the comparatively early (1973) review of this subject by the New South Wales Law Reform Commission. In its *Report on Evidence (Business Records)*,[43] the Commission argued that

> [t]here is no doubt that business records can be made and kept by a computer to a degree of accuracy which cannot, as a practical matter, be attained by a corresponding clerical system. Commonly they are so made, and kept. But errors in such records do occur. The cause of error is rarely a malfunctioning of the computer equipment.[44]

And that

> Generally speaking it may be said that with a well-prepared program the possibility of undetected error being caused by technical defects in the processing of information is very remote indeed. Apart from this, the reliability of the components now used in the electronic circuitry of computers and modern techniques of regular preventive maintenance make technical error a rare occurrence.[45]

---

43 New South Wales Law Reform Commission, *Report on Evidence (Business Records)* (Sydney: NSW Government Printer, 1973).

44 New South Wales Law Reform Commission, Note 43 at paragraph 39.

45 New South Wales Law Reform Commission, Note 43 at Appendix D, paragraph 50. The Law Reform Commission of Western Australia also considered the subject of the admissibility of computer output and again adopted a similar view. It suggested rather more succinctly, but no less significantly, that "a computer itself does not normally make mistakes": Law Reform Commission of Western Australia, *Report on the Admissibility in Evidence of Computer Records and Other Documentary Statements* (Perth, The Commission, 1978) 57.

In formulating recommendations for the evidentiary treatment of computer-produced material under New South Wales law, the Commission was plainly concerned only with the issue of the accuracy of input information. After asserting that "[s]tatements in business records produced by computers should be reliable if the information from which the statement was derived is reliable",[46] the Commission suggested that

> conditions should be imposed to safeguard the reliability of the source material ... [but] it [was] essential to relieve businesses using computers from the burden of proving strictly in legal proceedings the various steds [sic – steps] involved in the keeping of their records."[47]

The Australian Law Reform Commission subsequently embraced the attitudes of the New South Wales Law Reform Commission on this topic. In its 1981 *Hearsay Evidence Proposal* the Australian Law Reform Commission argued that

> [w]hile it is true that errors, accidental and deliberate, occur and can occur at every stage of the process of record keeping by computers the fact is, however, that they are the exception rather than the rule. They tend to occur at the stage when the information is fed into the system and there are techniques available which can be, and are, employed at each stage of the record keeping process to eliminate error.[48]

This passage was copied almost verbatim into the Commission's 1985 *Evidence (Interim) Report*.[49] It and other portions of the 1981 proposal were "fully endorsed" by the New Zealand Law Reform Commission in a report on the same subject.[50] The United Kingdom Law Commission reached a similar conclusion. In making its recent (1997) recommendations for repeal of section 69 of the *Police and Criminal Evidence*

---

46 New South Wales Law Reform Commission, Note 43 at Appendix D, paragraph 73. The Commission's use of the term 'reliable' is imprecise, but plainly refers to accuracy rather than functional dependability.

47 New South Wales Law Reform Commission, Note 43 at Appendix D, paragraph 74.

48 Australian Law Reform Commission *Evidence Reference, Research Paper No. 3, Hearsay Evidence Proposal* (Sydney: Australian Law Reform Commission, 1981) 127-128.

49 Australian Law Reform Commission, Note 5 at volume 1, paragraph 705.

50 New Zealand Evidence Law Reform Committee, *Report on Business Records and Computer Output*, (Wellington: The Committee, 1987) paragraph 127.

*Act* 1984 (UK), the Law Commission[51] relied upon the views of Colin Tapper, who had earlier argued that

> most computer error is either immediately detectable or results from error in the data entered into the machine. So widely has this been accepted that it has become institutionalized into the acronym "GIGO," or "garbage in, garbage out.[52]

These views exhibit a remarkable uniformity about the limited importance of possible problems with the operation of the computer. They are, however, views that have been subjected to virtually no serious scrutiny. The principal concern of this thesis is to review the way in which the principal approaches to the evidentiary treatment of computer-produced material deal with correctness of operation as a potential influence upon the accuracy of computer output. Do views of the kind cited here possess an adequate foundation, having regard to their potential to shape approaches to the evidentiary treatment of computer-produced material? In particular, is their adoption consistent with the aim of truth identification by rational means?

The possibility that is considered in this thesis is that attitudes of the kind identified here do *not* provide a suitable premise for an approach to the evidentiary treatment of computer-produced material. The parameters of the arguments that are presented in this thesis are described in the following section.

## 1.2 Organisation of the thesis

### 1.2.1 Argument

Any approach to the evidentiary treatment of computer-produced material must address the possibility that computer output will contain inaccuracies that are the

---

[51] Law Commission, *Evidence in Criminal Proceedings Hearsay and Related Topics* (London: H.M.S.O., 1997) at paragraph 13.7.

[52] Tapper C, "Discovery in Modern Times: A Voyage around the Common Law World" (1991) 67 *Chicago-Kent Law Review* 217-271, 248.

result of the *operation* of the computer. Since a principal goal of legal fact finding is the identification of truth by rational means, the manner in which this possibility is addressed must itself have a rational foundation. The principal existing approaches to the evidentiary treatment of computer-produced material fail to do this.

This argument is directed to the way in which the existing approaches ascertain and address the empirical characteristics of the process by which a computer produces output. The single most important of those characteristics is the *reliability* of that process.[53] It is important because it has the potential to be an indicator of the manner in which a computer has operated. What the thesis seeks to establish is that where unsupported assumptions about characteristics such as reliability are made, and are used in place of demonstrable propositions, the resulting outcome will not meet the requirements of rationality that attach to legal fact finding.

### 1.2.2 Methodology

The thesis establishes the arguments referred to in section 1.2.1 by reviewing and presenting principles and knowledge from the fields to which those arguments relate. It then uses that material as a basis for the evaluation of the principal approaches to the evidentiary treatment of computer-produced material.

The foci of enquiry are two distinct processes: legal fact finding, which is a social process, and the operation of computer hardware and software, which is a physical process. Physical and social processes are necessarily heterogeneous and an important foundation of the enquiry that is undertaken in this thesis is the proposition that there is in fact a nexus between the two processes. This nexus is the concept of

---

[53] This thesis uses a definition of reliability that is taken from an engineering context. It is "the ability of an item to perform a stated function under stated conditions for a stated period of time": Institute of Electrical and Electronics Engineers, Note 30 at 904. In the sense considered in this thesis, it is appropriate to refer to the reliability of devices or processes, because they have a *functional* role. It is less appropriate, and to a degree confusing to refer to the 'reliability' of information. Information is more precisely characterised as accurate or inaccurate, or in terms of the likelihood that it is accurate or inaccurate. It should be noted, however, that the distinction drawn here is generally not observed in the relevant legal literature.

reliability that was introduced in section 1.2.1. Reliability is a concept that is relevant in both the 'legal' environment (in which approaches to evidentiary treatment are formulated and applied) and the 'engineering' or 'technological' environment in which computers are designed, produced and tested.

In the latter environment, the application of a concept of reliability to computers has stimulated considerable interest (and associated research and publication efforts) because it is a determinant of important product characteristics such as quality and suitability.[54] At the same time, the concept relates directly to the notion of correctness of operation that was considered in section 1.1.3.2. It is a means by which the likelihood of correct operation might be possibly quantified or qualified. For this reason, it is relevant in the legal environment as well.

The premise for the methodology that is used in this thesis is that a concept (the reliability of computers) that is commonly relevant to both environments ought to be assessed and applied consistently. Since the legal environment is governed by the consideration that approaches to evidentiary treatment should facilitate rational truth identification, it ought to exhibit consistency with the engineering environment in the way in which it deals with the concept of the reliability of computers.

The operation of computers and the characteristics of their output are matters that have an empirical nature. As such, these matters might have been investigated empirically. An empirical methodology would, however, have been inadequate because it could not have produced findings about reliability that were applicable at a suitably general level. What the thesis argument deals with is the way in which knowledge and beliefs about a physical process are applied in a general context. As is discussed in chapter five, the predominant existing approaches to the evidentiary treatment of computer-produced material purport to have an application that is not limited to particular technologies, or to specific circumstances or patterns of use of technology.[55] An attempt to investigate the reliability of computers by means of a

---

[54] See for example the discussion in Lyu, Note 34 at 3-5.

[55] As to the undesirability of framing laws against such a specific backdrop, see *Dow Jones & Company Inc. v Gutnick* (2003) 194 ALR 433 at 465 per Kirby J. His Honour said that "[g]enerally

necessarily limited study could not produce findings suitable for the evaluation of regimes of such general application.

### 1.2.3 Scope

This thesis examines approaches to the evidentiary treatment of computer-produced material within the confines of the argument and methodology that were outlined in sections 1.2.1 and 1.2.2. A number of issues that are related to this subject are not considered. They include the following matters.

*Accuracy of input*

The thesis does not consider the adequacy of the manner in which the possibility of inaccurate input is addressed by existing approaches to evidentiary treatment. It is clear that this issue is independent of the issues of proper operation and reliability. Even if input were to be dealt with in a manner that is optimal having regard to the goal of facilitating rational truth identification, this would not compensate for inadequacies in the way in which the issue of correctness of operation may be addressed.

A further consideration is that, unlike the question with which this thesis deals, the question of accuracy of input has already received considerable attention in the prior literature. Further, as is noted briefly in chapter six in connection with a proposal for a new approach to evidentiary treatment, there is at least one existing approach to evidentiary treatment that deals adequately with this issue.

---

speaking, it is undesirable to express a rule of the common law in terms of a particular technology. Doing so presents problems where that technology is itself overtaken by fresh development."

*Ambiguous specification or imperfect understanding of intended function*

The thesis does not consider the potential of impact of problems that may arise because the fact finder does not properly comprehend the task or function that a particular combination of hardware and software is designed to perform. This is in large part a question of the consequences of the difference between a computer 'not doing the right job' and a computer 'not doing the job right'. It is possible to envisage cases in which software, for example, exhibits perfect reliability in the sense that it performs a specified function without failure. Even in such cases, it may be that the relevant function is not known or understood (and therefore not intended to be utilised) by a person who uses the software or indeed by a fact finder to whom the relevant output is provided.

Although this is an important problem, its significance is discrete from the problem considered in this thesis. Like the question of accuracy of input information, it is properly viewed as cumulative to the problems that are considered in this thesis.

*Compromise of reliability by deliberate action*

A further 'cumulative' problem arises out of the possibility that the operation of a computer may be compromised as a result of matters that fall outside conventional reliability analysis. This possibility may arise in context of deliberate action to alter or to substitute program code or stored data so that the function that is actually performed when a program is executed differs from that which the user expects. Agents for such alteration or substitution in this context include popularly documented 'computer viruses' and similar phenomena.

Consideration of the foregoing matters (input information, functional ambiguity and deliberate subversion) may well reveal further shortcomings in existing approaches to the evidentiary treatment of computer-produced material. These shortcomings would, however, be discrete and independent in the sense that their presence does not negate the criticisms that may be made in this thesis. At most, a consideration of such matters would yield further, separate bases of criticism of the existing approaches to evidentiary treatment.

*Computer-produced material as 'demonstrative evidence'*

This thesis considers the evidentiary treatment of material that has been produced in the 'real world' by hardware and software that has not been designed or used to meet the needs of legal fact finding. It does not consider the special case in which a computer is used to provide a graphical or similar demonstration of some event, phenomenon or transaction that may be in issue in a given case. Whilst such cases naturally engage considerations of reliability and accuracy, they also engender considerations that are unique to the fact that such 'demonstrative evidence' is used not as a principal record of events, but to reconstruct or re-tell a version of the facts in dispute. As such, these cases fall outside the scope of this thesis.

*Other goals of legal fact finding*

The thesis also omits a consideration of goals of legal fact finding other than truth identification. Such goals include matters such as economy, expedition[56] and 'fairness'. As to the first two of these goals, it has been said that

> [a] law suit is not an abstract scientific investigation to discover absolute truth. It is a very practical affair aimed at resolving disputes between parties within a reasonable time and at a reasonable cost.[57]

The relevant rationale is that establishing the facts to which the substantive law is to be applied cannot take unduly long, not least in criminal cases in which a defendant may be held in custody before and during the trial. It also should not be a prohibitively expensive undertaking, especially in those civil cases in which litigants must usually sponsor the process from their own resources.[58]

---

[56] As to the goals of economy and expedition see generally Australian Law Reform Commission, Note 5 at volume 1, paragraph 57; Woolf H K, *Access to Justice, Interim Report to the Lord Chancellor on the Civil Justice System in England and Wales* (London: HMSO, 1995) at chapter 1, paragraph 3.

[57] Law Reform Commission of Canada, *Report on Evidence* (Ottowa: Information Canada, 1975) 52.

[58] Woolf, ***Interim,*** Note 56 at chapter 3, paragraphs 13 and 18.

Outcomes of economy and expedition may in fact compete with truth identification because the dedication of abundant time and expenditure are what permit the kind of detailed and meticulous enquiry that promotes accuracy in the identification of truth. It will be more likely that superior results will flow from activities that are directed to the active identification and questioning of witnesses to an event and to the retrieval of contemporaneous records of it. The reality is, however, that these positive actions take more time and cost more money than doing nothing at all to investigate a matter. Reasonable economy and expedition are outcomes that are sought from adjudication but which can compete fiercely with the goal of truth identification.

Fairness is a less tangible, but nevertheless important, objective that can compete with truth identification. In their introduction to *Evidence and Proof*, Twining and Stein argue that

> [t]here are also, as has been powerfully argued, certain 'process values'– amongst them fairness, integrity and the parties' participation–which should be maintained throughout all fact-finding processes independently of their effect on the accuracy of outcomes. These values emanate from political morality and should, arguably, shape the standards of admitting and examining evidence to which individuals affected by official decisions are to be entitled.[59]

Wigmore offered a consistent view.

> The ... features of the English system [of evidence] are due chiefly to the use of the jury in all cases, criminal and civil (except chancery), but *secondarily to a certain English spirit of fair play or 'true sport,'* in legal procedure generally.[60] (Emphasis added.)

A report by the New Zealand Law Reform Commission in 1987 that dealt specifically with the question of computer-produced material in an evidentiary context referred to fairness and cost saving and efficiency as policy objectives that were relevant to the formulation of legislation in this area.[61] In the United States, Fed. R. Evid. 102

---

[59] Twining W L and Stein A, 'Evidence and Proof' in *The International Library of Essays in Law and Legal Theory* (Aldershot: Dartmouth, 1992) xxiv.

[60] Wigmore, Note13 at 4.

[61] New Zealand Evidence Law Reform Committee, Note 50 at paragraph 123.

identifies similar values. This rule, which has been said to set out the "foundational legal principles that bind the trial court in its application of the rules",[62] provides that

> [the Federal Rules of Evidence] shall be construed to secure fairness in administration, elimination of unjustifiable expense and delay, and promotion of growth and development of the law of evidence to the end that the truth may be ascertained and proceedings justly determined.

Fairness is, however, a general term. It can relate to many matters, both inside and outside the courtroom. Privacy is one such matter, so that in the common law jurisdictions arbitrary searches by instrumentalities of law enforcement personnel are regarded as undesirable. Such searches are prohibited to varying degrees in these jurisdictions, although the prohibition is strongest in the United States.[63] Although it would involve time and expense, the widespread and pervasive examination of premises, personal effects and even genetic information could reveal material that may assist in the ascertainment of truth in the proceedings that may be pending in a particular jurisdiction at any given time. Despite this, regard for reasonable preservation of personal privacy competes with and, to a degree, subjugates the objective of truth identification.

Determination of the dispute by a disinterested adjudicator is also an aspect of fairness. A witness to events that subsequently become the subject of litigation is not permitted to play an adjudicative role—either as judge or juror—in that case.[64] This restriction is not, however, necessary to preserve the capacity to find the truth. In the absence of pre-existing sympathy or antipathy for one of the litigants,[65] a party who merely witnessed the events in question may be thought to be well equipped to

---

[62] Leonard D P, "Power And Responsibility In Evidence Law" (1990) 63 *Southern California Law Review* 937-1013 at 957.

[63] Where it is constitutionally enshrined: *Constitution of the United States of America, Amendments,* Article IV.

[64] A notable exception is the power of a judge to act summarily to deal with contempt when committed in the 'face of the court', as to which see generally: Miller C J, *Contempt of Court* (Oxford: Oxford University Press, 3rd ed, 2000) 140-147.

[65] Which are matters that also require disqualification of the relevant adjudicator as an aspect of procedural fairness. For a recent discussion see *R v Bow Street Metropolitan Stipendiary Magistrate Ex Parte Pinochet Ugarte (No 2)* [1999] 1 All ER 577 (HL) at 590-591 per Lord Goff of Chieveley. The consequences can, in some cases, be more extensive. In Australia a judicial officer who exercises federal jurisdiction in a case in which they have a personal interest may commit a criminal offence: *Crimes Act* 1914 (Cth), s 34.

determine the truth. Even so, fairness is said to require that such a party not act in the role of fact finder.

Can consideration of these important matters legitimately be severed from the goal of truth identification? It is arguable on the one hand that the manner in which these other goals of legal fact finding are addressed may mitigate the criticisms of existing approaches to evidentiary treatment. This may be the case when those criticisms have focused only upon the question of the how the goal of truth identification is pursued. An approach to evidentiary treatment which exhibits shortcomings with respect to the goal of truth identification may nevertheless provide a regime for evidentiary treatment that is inexpensive to the proponent of evidence because it imposes no restrictions upon the admissibility of material. Is it at all important that such an approach is more economical? The answer, according to Damaska, is that it might be. He contends that

> [a]ll adjudication—no matter what its purpose—lies in the domain of social activity where truth values cannot be maximized because they are not the only ones that count.[66]

Despite this consideration, an investigation that considers the goal of truth identification alone is legitimate. This conclusion follows from the fact that even if this goal should not be maximised, it cannot be disregarded as significant in its own right. Even if, as Damaska appears to suggest, there is to be something akin to an attempt to optimise the outcome with respect to several goals, then it is still necessary to have information about the outcomes that are delivered for each of the individual goals that are of interest. There will still be a need to know something about the extent to which approaches to evidentiary treatment have efficacy with respect to *each* goal of legal fact finding. This need persists for two reasons. First, each of the goals are valuable in their own right, even if none is paramount. Second, if there is to be any 'balancing' of the goals, then something must be known about how an approach to evidentiary treatment affects the realisation of each goal on a discrete basis.

It may also be the case that the goal of truth identification has a special significance that means that it is not entirely exchangeable with the other goals that have been mentioned. Twining's articulation of the assumptions that underlie the 'rationalist model' of legal fact finding supports this view. He refers in particular to an assumption that

> [t]he pursuit of truth (i.e. seeking to maximize accuracy in fact-determination) is to be given a high, but not necessarily an overriding, priority in relation to other values.[67]

When the potential for legal fact finding to give rise to serious outcomes—such as the imposition of substantial penalties in criminal cases—is considered, this proposition has an intuitive appeal that is not easily displaced. It is on this basis that the scope of this thesis may properly be limited to consideration of the goal of truth identification alone.

### 1.2.4 References to the Australian 'Uniform' Evidence Acts

This thesis includes references to the law of evidence in Australia. In three Australian jurisdictions, a substantial portion of that law is contained in near uniform *Evidence Acts*. The jurisdictions and enactments are the Commonwealth of Australia, which has enacted the *Evidence Act* 1995 (Cth);[68] New South Wales, which has enacted the *Evidence Act* 1995 (NSW) and Tasmania, which has enacted the *Evidence Act* 2001 (Tas).

References in this thesis to provisions of the *Evidence Act* 1995 (Cth) are, unless an indication to the contrary is given, also references to the equivalent provisions of the New South Wales and Tasmanian statutes.

---

66 Damaska, Note 3 at 121.

67 Twining, Note 13 at 73.

68 In addition to its application to federal courts, certain provisions of the Commonwealth Act apply in state courts exercising state jurisdiction: see *Evidence Act* 1995 (Cth), s 4.

### 1.2.5 Chapter outline

This thesis is divided into seven chapters including this introduction. An outline of the content of chapters two to seven is set out below.

Chapter two examines the goal of rational truth identification within the context of legal fact finding. It considers what the goal entails and the means by which it can be pursued. The focus of this chapter is the *process* of legal fact finding, and the environment in which it is undertaken. The findings of this chapter provide a general picture of what approaches to the evidentiary treatment of any material should achieve, and how they might be expressed.

Chapters three and four direct attention to the subject matter that is of specific interest to the thesis: computers and computer-produced material. They review the design, operation and reliability of computers from a physical perspective. This review necessarily encompasses the 'engineering' principles, methods and philosophies that underlie this topic. Chapter three describes the principal characteristics of computer-produced material that are relevant to its use in legal fact finding. It gives an account of the nature of the operation of a computer, and of the means by which output is produced.

Chapter four explores the reliability of computers. It considers the methods that are available for assessing reliability and the kinds of conclusions that can be drawn about reliability when it is considered from a physical perspective. The principal function of this chapter is to develop a foundation for testing 'legal' views about the reliability of computers. This, combined with the findings of chapter two, comprises a framework for the evaluation of existing approaches to evidentiary treatment.

Chapter five reviews the principal approaches to the evidentiary treatment of computer-produced material in common law jurisdictions. The premise for this review is that approaches to the question of evidentiary treatment must deal with the issue of reliability in a rational manner. The chapter identifies the assumptions that underlie the existing approaches and then applies the findings of chapters three and four to test these assumptions. Particular emphasis is given to the extent to which the

approaches depend upon false or unsupported assumptions about the reliability of computers.

Chapter six presents an alternate approach to the evidentiary treatment of computer-produced material. The chapter reiterates the 'problem' that the use of computer-produced material as evidence presents and considers the means by which the severity of the problem can be reduced. The approach that is presented seeks to overcome the shortcomings of the existing approaches and to do this in a way that exhibits a foundation that is demonstrably rational.

Chapter seven presents an overview of the findings of the thesis and the conclusions that may be drawn from them. It also discusses the potential for their further application.

# 2. Legal fact finding and truth identification

## 2.1 Introduction

Chapter one referred to rational truth identification as a primary goal of legal fact finding. The purpose of this chapter is to describe more fully what this goal entails, how it might be pursued and the mechanisms that may be engaged to pursue it. In doing this, this chapter reviews relevant aspects of the process of legal fact finding and the environment in which it is undertaken. The findings of this chapter support the thesis by identifying a framework within which approaches to evidentiary treatment can be evaluated.

The chapter commences by defining what rational truth identification entails, having regard to the theoretical premises that underlie legal fact finding. Methodologies of fact finding that are congruent with the rational ideal are also considered. The chapter next describes the mechanisms that are available within the law of evidence to realize rational truth identification. In this context the concepts of the 'admissibility' and the 'weight' of evidence are distinguished. The roles of these concepts in the context of rational truth identification are considered. The chapter concludes by considering particular constraints that apply to the formulation of any approach to evidentiary treatment. These constraints derive from a variety of historical and 'cultural' considerations that exert influence upon the development of Anglo-American evidence law. This latter examination serves two purposes. It provides a more

complete framework for the evaluation of existing approaches to evidentiary treatment. It also provides guidance as to the scope that exists for the development of alternative approaches to evidentiary treatment.

## 2.2 What is rational truth identification?

### 2.2.1 Foundations

> The 'rational' system is one which uses reason, so far as is feasible, in the determination of disputed questions of law and fact.[1]

William Twining's account of a rationalist tradition in Anglo-American evidence law[2] provides a starting point for examining the ideal of 'rationality' as it applies to legal fact finding. This account does not, however, provide a complete picture of all that rational truth identification necessarily entails. At the outset, what is 'rational' can be understood by distinguishing it from what might be called 'irrational' fact finding.[3]

References to 'irrational' fact finding are usually associated with the early modes of trial that were the forerunners of the modern law of evidence. Twining contrasts "a 'rational' mode of determining issues of fact in contrast with older 'irrational' modes of proof ... such as battle, compurgation, or odeal [sic – ordeal]."[4] According to Wigmore, these older modes of trial were prevalent prior to the thirteenth century[5] and involved

> no room for our modern notion of persuasion of the tribunal by the credibility of the witnesses, for the tribunal merely verified the observance of the due formalities and did not conceive of these formalities as directly addressed to its own reasoning powers.[6]

---

1 Twining W L, "The Rationalist Tradition of Evidence Scholarship" in Twining W L, *Rethinking Evidence: Exploratory Essays* (Oxford: Blackwell, 1990) 32-91 at 33.
2 Note 1.
3 This term is used by, for example, Twining: Twining, Note 1 at 72.
4 Twining, Note 1 at 73-74.
5 Wigmore J H, *Evidence in Trials at Common Law* (Tillers revision) (Boston: Little Brown and Company, 1983) §8.
6 Wigmore, Note 5 at §8.

The formalities included 'ordeals' in which the accused[7] was subjected to a process that would normally result in injury. The proceedings against the person would be dismissed only if 'divine intervention' prevented the injury.[8] Also included in the category of 'irrational modes of trial' was the process of trial by combat. Under that process the accuser and accused would do battle, the assumption being that the victor would have had the benefit of divine intervention. In another of the 'irrational' processes—trial by oath—the accused could swear an oath of innocence as to the subject of the proceedings. The fact that the oath taker was observed not to suffer 'divine wrath' was taken to be evidence of the truth of the oath or, more relevantly, an "incontrovertible divine indication of [their] innocence."[9]

Much is made of the chacterisation of these methods of fact finding as being directed not to the *discovery* of truth, but instead to the divine revelation of truth.

> Judgment was the judgment of God, a revealed, not a discovered, truth: and the role of the human tribunal was merely to provide a setting for the revelation.[10]
>
> ..
>
> Courts and judicial procedure merely set the stage for this revelation: to usurp its prerogative by independent human inquiry would have been little short of sacrilegious.[11]

Two important distinctions between rational and irrational fact finding follow from this. First, irrational fact finding is distinguished by the fact that it makes no attempt to discover the truth about "particular past events in issue in a case",[12] as opposed to seeking the revelation of truth by means external to the fact finder. Second, irrational fact finding involves no attempt at enquiry or reasoning to determine what may have occurred.

---

7 The historical accounts make few distinctions between criminal and civil proceedings and the term 'accused' is used here for consistency with those accounts. It seems, however, that some modes of trial applied to civil proceedings as well. See in this context: Stone J, Wells W A (ed) *Evidence: Its History and Policies* (North Ryde, New South Wales: Butterworths, 1991) 7.

8 Stone and Wells, Note 7 at 6.

9 Stone and Wells, Note 7 at 6. See also Thayer J B, *A Preliminary Treatise on Evidence at the Common Law* (reprint: South Hackensack, New Jersey: Rothman, 1969) 24-25.

10 Stone and Wells, Note 7 at 2.

11 Stone and Wells, Note 7 at 4.

12 Twining, Note 1 at 73.

When the distinctions are drawn in this way, they suggest that any attempt at enquiry and reasoned decision making by the fact finder will be rational. What has to be considered, however, is that when the ancient religious formalities are put to one side, the outcome of a process such as ordeal is still a source of information. It may be believed, for instance, that the physical effect of the process upon an accused reflects the truth or falsity of factual contentions that are made in given proceedings. If this is the case, then the use of observations of those effects (or their absence) to reach a reasoned conclusion about the relevant issues will meet the conditions of rationality that are implied by the two distinctions that were referred to above. Yet even if such a process may be considered rational, more is required before the attempt will be congruent with the rational identification of *truth*. Rational enquiry clearly involves the use of sources of information, but not all sources of information are likely to be indicative of truth.

The real problem with such a scenario is that there is no rational foundation for the belief that the relevant source of information (outcome of the ordeal) is likely to be indicative of truth. In the example that was given, the belief may have been wholly arbitrary. If a process of fact finding uses sources of information of this kind then, despite the fact that reasoning occurs on one level in the sense that the decision follows the observation, there is no basis for supposing that the outcome of the process will be congruent with the truth.

Wigmore approaches the problem more directly, arguing that the essence of rational fact finding is tied directly to the quality of the sources of information that are used.

> The contrast, it may be noted, between employing rational and nonrational modes of proof is after all not between the use of scientific reasoning and the employment of superstitious ordeals; it is rather between employing the best standards we know and those that we realize are not the best.[13]

The rational identification of truth requires more than just the use of sources of information and the production of reasoned decisions that are in conformity with what might be thought to be indicated by those sources. What is also required is some discrimination in the choices that are made about the information that is to be used.

---

[13] Wigmore, Note 5 at §9.

The nature and extent of this discrimination will be variable and will depend upon the inherent characteristics of the source of information in question. Some sources of information may be considered to have no capacity whatsoever to indicate truth, and it is for this reason that they should not be used. The outcome of a physical ordeal of the kind employed in early legal fact finding falls plainly into this category.

What is therefore vital to rational truth identification is the formulation of regimes of evidentiary treatment that are themselves based upon a rational consideration of the characteristics of accuracy of a given source of information. In some cases the consideration that is called for will involve a detailed examination of complex processes that are associated with that source of information. As the subsequent chapters of this thesis establish, computer-produced material presents such a case.

These considerations anticipate the need for regulation of the process of legal fact finding. As was identified in chapter one, the law of evidence meets many aspects of this need.[14] The purpose of this thesis is, however, to evaluate specific aspects of the law of evidence against the goal of rational truth identification. It is not a concern of this thesis merely to attempt to explain the basis for the existence of certain rules. The rules of evidence are considered here as means by which the goal of rational truth identification might be realised.[15]

### 2.2.2 Methods

Defining what rational enquiry is, and how it is to be carried out, involves more than distinguishing it from the irrational methods of enquiry that were referred to in section 2.2.1. It also involves more than identifying the need to constrain in some way the sources of information that may be used in a fact finding exercise. Yet beyond a broad distinction between the historical 'irrational' methods of fact finding and the

---

[14] As was observed in chapter one, the adversary system, which is an overlay upon the goals of legal fact finding, also produces a need for regulation.

[15] Although the following section makes reference to the extent to which existing practices conform to particular methodologies of enquiry, rules of evidence are not relied upon as authority for the existence of particular goals. Conversely, no attempt is made to explain the cases in which

contemporary 'rational' methods, the discussion in the literature is largely unhelpful in this regard.

That discussion descends quickly into adage and aphorism. There are references to "everyday practical affairs", "common sense", "the common course of events"[16] and to "calm and careful reasoning".[17] The articulation of these general notions is problematic. It leaves open the question of what the notions really entail and whether a particular methodology does or does not embrace them. At most, it signals to a proponent of a particular methodology the general kinds of claims that should be made in respect of the things that can be achieved through the use of that methodology. This does not advance matters toward a shared understanding of the basic concepts that are involved, and it is such a shared understanding that is required. There is a need to know and to agree upon what matters like 'everyday practical affairs' and 'common sense' really are.

To some, this absence of definition has, however, not been so troubling. Despite his extensive comparative surveys of a number of aspects of legal fact finding, Damaska concluded that "fortunately, also, there [was] no need for [him] to enter the intellectual minefield that surrounds the question of what rational enquiry precisely entails."[18] It is not surprising then that Twining reports simply that the "orthodox tradition of Evidence Scholarship" is one in which the "rationality of the process is by and large assumed".[19] In the present context, however, it is necessary to go somewhat further in exploring what rational fact finding actually entails.

Twining makes two observations that are especially useful to any attempt to define further a methodology for rational fact finding. First, rational enquiry is founded

---

particular rules may appear to be at odds with those goals nor, in the context of this thesis, is any such attempt necessary.

16 Twining W L "Taking Facts Seriously" in Twining, Note 1, 12-31 at 21.

17 Wigmore J H, *A Students' Textbook of the Law of Evidence* (Brooklyn: Foundation Press, 1935) 5.

18 Damaska M R, *Evidence Law Adrift* (New Haven: Yale University Press, 1997) 95.

19 Twining, Note 16 at 26.

upon correspondence based empiricism;[20] second, rational enquiry employs inductive reasoning.[21]

*Correspondence based empiricism*

Correspondence based empiricism involves a method of enquiry that is based upon the 'correspondence theory' of knowledge. This theory relies upon the existence of a reality that can be tested otherwise than by mere assertions that may or may not be true. It measures the truth of a statement by how well it corresponds with the reality of the subject of the statement. For example, the assertion that 'the sky is blue' will represent the truth if the sky really is blue (as opposed to red, grey or black) when the statement is made.[22]

Subscription to this mode of enquiry places very heavy emphasis upon observation and empirical experience.[23] This also explains the value that attaches to material that is reasonably contemporaneous with an event that is of interest. Photographs of the scene of an accident that are taken shortly after the accident seem more helpful than ones taken many months later. Testimony that is recounted from memory—as all testimony is—seems more valuable when only a relatively short time has elapsed between the observation and the occasion on which the testimony is given. In these cases there is less prospect that intervening events will detract from the ability of the relevant material (photographs, memory)[24] faithfully to recount the events or transactions that are of interest.[25] Close temporal connection with an event in

---

20 Twining, Note 1 at 72 and Twining, Note 16 at 26. See also Twining W L, *Theories of Evidence: Bentham and Wigmore* (London: Weidenfeld and Nicolson, 1985) 52. Damaska also considers that legal fact finding is based upon a correspondence theory of truth: Damaska M R, "Truth in Adjudication" (1998) 49 *Hastings Law Journal* 289-308, 291.

21 Twining, Note 1 at 73, 74. See also Wigmore, Note 5 at §30.

22 This brief account of the theory says nothing of its varieties, or of the criticisms that have been made of it. As to the latter see for example: O'Connor D J, *Correspondence Theory of Truth* (London: Hutchinson, 1975). For Damaska's formulation see Damaska, Note 20 at fn 3: "This theory is predicated on the idea that truth consists of a relationship between words and the world."

23 Compare Twining, *Theories of Evidence: Bentham and Wigmore*, Note 20 at 52: "observation, experience and experiment" (attributing these words to Bentham's *Rationales of Judicial Evidence*).

24 For discussions of some aspects of the problems of memory from a psychological perspective see Trankell A, *Reliability of Evidence. Methods for Analyzing and Assessing Witness Statements.* (Stockholm: Beckman, 1972) and also Law Commission of New Zealand, *Evidence Miscellaneous Paper 13: Total Recall? The Reliability of Witness Testimony* (Wellington: Law Commission, 1999).

25 In *Papakosmas v R* (1999) 196 CLR 297, Gaudron and Kirby JJ considered the probative value of evidence close in time to an incident of interest. Their Honours said that: "[t]he nature and

question carries with it a measure of authority because of the influence of empiricism in rational enquiry.

Rational enquiry that embraces correspondence based empiricism will also have little use for speculation and conjecture as potential aids to legal fact finding. Consistently with this expectation, the law of evidence places considerable limitations upon the use of opinions as evidence. These limitations are referable to empirical ideals. In the United States, Fed R. Evid. 402 illustrates this consistency. It limits non-expert testimony to "opinions or inferences which are rationally based on the perception of the witness". In Australia, sections 76-79 of the *Evidence Act* 1995(Cth) reflect similar restrictions upon the use of opinions as evidence.

*Inductive Reasoning*

Induction is "the process of inferring a general law or principle from the observation of particular instances."[26] Induction is unlike deduction because it does not produce demonstrably correct propositions that flow from stated premises.[27] Induction produces inferences, not conclusions. Although those inferences will always be inconclusive, they can vary in strength. For example, an inductive inference is said to be stronger if it is arrived at after observation of a large number of independent events.[28]

The employment of inductive reasoning has an important consequence: decisions about facts in litigation do not have to be supported by demonstrably correct conclusions. Decisions can be made when the evidence supports only inferences. If this were not the case, the range of enquiries with which fact finding could deal would be limited and the incompleteness of a body of evidence in any given case would usually be fatal to decision making. Indeed, the endorsement of inductive reasoning

---

degree of the connection necessary before a statement is probative of the fact asserted in it will, of course, depend on the nature of that fact and, if it be different, the fact ultimately to be proved. Even so, the connection will *ordinarily be found in the close contemporaneity of the statement with the fact in issue*" (at 315) (Emphasis added.)

[26] Simpson J A and Weiner E S (eds), *The Oxford English Dictionary* (Oxford: Clarendon Press, 2nd ed, 1989) volume seven, page 890.

[27] Copi I M, *Introduction to Logic* (New York : Macmillan, 5th ed, 1978) 32.

suggests that rational legal fact finding is possible even when there are gaps in the evidence.[29] This is a necessary consequence of the need for legal fact finding to reach conclusions. Moore expresses that need in the following terms.

> Where the testimony of the several witnesses relates to a fact of importance, and it is evident that *something* occurred, the court should be as reluctant to abandon an attempt to ascertain the truth among stories presenting conflicting details, as it would be to pronounce a written contract or a statute void for uncertainty and thus, in effect, declare that the parties or legislators had no intention whatever. As the Supreme Court of Wisconsin said, if rights were to be lost as a matter of course because of differences in the perceptive faculties, habits of attention, or the memories of witnesses, in a very large proportion of cases involving wrongs to be redressed, the law would fail to furnish a remedy.[30]

### 2.2.3 Philosophical considerations

The account of 'rational' fact finding that was provided in the preceding sections of this chapter involves an important premise, namely that there is a 'truth' that is in fact susceptible to discovery. This premise involves assumptions about truth that are not universally accepted at a basic philosophical level. As a result, it is necessary to consider briefly the relevant objections to those assumptions.

*Philosophical 'scepticism'*

Damaska[31] and Twining[32] review the key philosophical objections that have been expressed in connection with the stated aim of truth identification in legal fact finding. Both describe the essence of the philosophical challenge as one that disputes that

---

[28] Cohen J L, *An Introduction to the Philosophy of Induction and Probability* (Oxford: Clarendon, 1989) 6.

[29] Compare Twining, Note 1 at 73: "[t]he establishment of the truth of alleged facts in adjudication is typically a matter of probabilities, falling short of absolute certainty."

[30] Moore C C, *A Treatise on Facts or the Weight and Value of Evidence* (Northport, New York: Edward Thompson, 1908) Volume 2, 784-785.

[31] See Damaska, Note 20. The discussion in this article mirrors the approach in *Evidence Law Adrift*: Damaska Note 18.

[32] Twining W L, "Some Scepticism about Some Scepticisms" in Twining, Note 1, 92-152.

truth, or knowledge, can exist at all.[33] These are the concerns of the philosophical sceptic.[34]

Damaska seeks to answer these concerns by pointing to their failure to "account for existing fact-finding arrangements".[35] He also points out that the task faced by the fact finder in the adjudicative setting is very different from that faced by the philosopher.

> Because other intellectual genres are driven by different purposes even as they cope with the same aspects of reality, their criteria and conceptual instruments may be ill-suited to adjudicative fact-finding.[36]

Twining's answer to philosophical scepticism is to point to the relative scarcity of proponents of viewpoints that challenge the goals of evidence law at this fundamental level.[37] Twining is also in broad agreement with Damaska about the existence of a fundamental difference between the work of the philosopher and the work of the adjudicative fact finder.

> [T]he law is not concerned with ultimate questions about philosophical concepts of truth and proof. In this view the law proceeds *as if* there is a real world, accessible to the human mind, and that the truth of statements can be tested by evidence.[38]

These responses to philosophical scepticism are made against a compelling backdrop: the sheer longevity of legal fact finding as a social enterprise. Even though it has evolved over time, legal fact finding has retained the support of the legal and political infrastructures from which the paradigms and the power for the administration of law derive.

---

33 See Twining, Note 32 at 99-100; Damaska, Note 20 at 289. This account is a simplification, both of the ideas that the philosophical theories present and of the way that the two authors expose them.

34 The term abounds in the philosophical literature. See for instance Ayer A J, *The Problem of Knowledge* (London: Penguin, 1956) 36-41.

35 Damaska, Note 20 at 289.

36 Damaska, Note 20 at 297.

37 Twining, Note 32 at 103.

38 Twining, Note 32 at 100.

More importantly, the evolution of fact finding has not altered the underlying premise that the truth about a given state of affairs can be a legitimate subject of knowledge. As was discussed in section 2.2.1, the 'irrational' modes of legal fact finding have given way to methods that are said to be 'rational', but a truth-seeking aspiration remains. It was, in reality, present in both kinds of legal fact finding. The old methods were a bad way of seeking a worthwhile objective. They have been replaced by better methods, at least from a contemporary viewpoint. They may again be replaced in the future. It will, however, only be when the objective of truth identification is abandoned as wholly misconceived or illusory that philosophical scepticism will present a problem that has to be addressed in the context of an evaluation of the kind with which this thesis is concerned.

This does not mean that legal fact finding is free to press on arbitrarily and without any self-consciousness until it becomes so egregious that support for it evaporates. What it must do is strive to do the best that it can to discover truth. This is the course that the rationalists have taken.[39]

*The 'social construction' of reality*

A related philosophical objection to the viability of truth seeking involves the complaint that many of the matters that might be the subject of 'the truth' are not intrinsic to nature. Rather, it is said, they are socially constructed. As was discussed in section 2.2.2, the methodology of legal fact finding draws upon a correspondence theory of knowledge and its premise that there is an empirical reality against which the truth of statements can be measured.

The objection that is made is that this premise does not work well when the subject matter of the enquiry is socially constructed. It is worse still that, according to some theorists, 'reality' itself is a social construction that is subject to all manner of

---

[39] Twining, Note 1 at 74-75. For an example of a contemporary adoption of the notion of rationality in this context see *Evidence Act* 1995 (Cth), s 55(1). That section provides that "[t]he evidence that is relevant in a proceeding is evidence that, if it were accepted, could *rationally* affect (directly or indirectly) the assessment of the probability of the existence of a fact in issue in the proceeding." (Emphasis added).

economic, political, class and gender biases and distortions.[40] The answer to this contention is that even in the areas of enquiry in which these biases and distortions may emerge most directly, 'reality' cannot be disregarded or treated as though it lacks validity just because it involves social constructions. What is observable can still matter. MacCrimmon illustrates this dilemma in the context of gender-based construction.

> How do we reconcile a belief that reality is a social construction with the goal of incorporating the empirical reality of women's experiences [of, inter alia, domestic violence] into legal decision making?[41]

Damaska argues that the correspondence theory can be applicable to some social constructs, not just to intrinsic elements of nature.[42] In the example given in section 2.2.2 ('the sky is blue'), blue is not an intrinsic element of nature; it is a socially constructed element.[43] Despite this, the concept of distinguishable colour can still be used to identify the properties of a natural element.[44] Although the correspondence theory can work with social constructs, legal fact finding is rather more limited to matters of 'fact' that are distinct from value judgments. This may be because "[t]he law of evidence has traditionally viewed fact-finding in legal decisions as a rational process that can be divorced from questions of value."[45]

Damaska takes a similar position.[46] Some social constructs involve only 'simple' questions such as, to use Damaska's examples, the presence of a chemical in the blood of a deceased, rather than more value-laden enquiries about whether a situation

---

[40] See Twining, Note 32 at 113 and, regarding gender issues, the account of feminist legal theory in Harris J W, *Legal Philosophies* (London: Butterworths, 2nd ed, 1997) 295-296.

[41] MacCrimmon M, "The Social Construction of Reality and the Rules of Evidence" in "A Forum on Lavellee v. R: Women and Self Defence" (1991) 25 *University of British Columbia Law Review* 23-68 at 36-50 at 45.

[42] Damaska, Note 20 at 291.

[43] Damaska uses the example of the days of the week. Despite being "social artifacts unknown to nature" they can operate as a mechanism to describe reality, at least in terms of when a particular event occurred: Damaska, Note 20 at 291.

[44] Assuming that the concept of the 'sky' can be reconciled with some physical element. Is the sky the region above the earth, or is it the collection of particles that actually scatter the portion of the sunlight that we call blue? The potential for recursive enquiries of this kind is considerable.

[45] MacCrimmon, Note 41 at 37.

[46] Damaska, Note 20 at 300.

was dangerous, or a picture sexually explicit.[47] For the more complex questions, Damaska urges a measure of deconstruction.

> Adjudicators are expected to accept a story as true when it is amply supported by items of evidence relating to the facts of the case. And what these small foot soldiers of verity are expected to achieve is to establish that a match exists between factual propositions woven into the fabric of a story and the way the world really is.[48]

Despite this, there are limitations that must be faced. Law purports to regulate so many of the varied events, behaviours and transactions of a society that it inevitably gives rise to 'hard' questions about values. Even if, as Damaska argues,[49] the philosophical questions raise only a "false cause for alarms", there is a warning that still remains to be heard. This is, however, a warning for substantive, rather than procedural, lawmakers. The problem of dealing with value judgments is not, in any event one that is fatal to legal fact finding. There is not so much a problem with the ability to find the truth at all, as there is a problem with the ability to find the truth about facts that are difficult to untangle from questions of value. The "small foot soldiers of verity"[50] can be expected to accomplish only so much.

### 2.2.4 Implications

A number of matters emerge from the foregoing discussion of the concept of rational truth identification. Rationality involves enquiry by the fact finder, which in turn involves recourse to sources of information. Preference (as opposed to ambivalence) for an outcome that reflects truth requires that some importance be attached to the characteristics of accuracy of the sources of information that are to be consulted. More particularly, consideration of these characteristics is an important concern in the formulation of any regime for the evidentiary treatment of given material.

---

47 Damaska, Note 20 at 300.
48 Damaska, Note 20 at 292.
49 Damaska, Note 20 at 290-291.
50 Damaska, Note 20 at 292.

The kind of 'truth' or accuracy with which legal fact finding is concerned is one that is tied to an empirical or, perhaps, a 'physical' reality. This is the context in which the characteristics of accuracy of a given source of information must be considered. Despite this, rational truth identification does not seek conclusive truth. This means that the sources of information that may be used are not restricted to those that are demonstrably infallible. Rather, the proper consideration is whether there is a likelihood that the source of information will reflect an underlying empirical truth.

## 2.3 Mechanisms for truth identification

Section 2.2 considered the question: what does rational truth identification entail? This section examines the mechanisms that are available to pursue rational truth identification in legal fact finding in a common law framework. Two principal mechanisms[51] are relevant in this context. These are the application of rules about the 'admissibility' of evidence, and the process of allocating 'weight' to the evidence that has been found to be admissible.

Rules about admissibility are applied to determine whether a fact finder may make use of particular information. By contrast, the process of allocation of 'weight' involves the fact finder evaluating, in a way that is largely unconstrained, the strength of the admissible information. The purpose of this latter process is to enable decisions about truth to be made on the basis of a body of (potentially) conflicting evidence.

What weight is, and how it is applied, are matters that are said to lack clarity.

> Perhaps 'weight' is one of the concepts which cannot easily be given a distinct conceptual identity and are best recognised as functions of other concepts. On this view, weight is primarily a function of relevance and credibility and our understanding of it is likely to be more intuitive than analytical; weight is something we are more likely to 'appreciate' than to understand.[52]

---

[51] Also important is the employment in common law jurisdictions of an 'adversarial' mode of trial and rules about the onus and standard of proof. These matters are considered below in section 2.4.

[52] Roberts G, *Evidence: Proof and Practice* (Sydney: Law Book Company, 1998) 77. Another view is that "[i]t is impossible to generalise safely and precisely about how a court should weigh facts,

It is also said that

> [u]nlike admissibility, the weight of evidence cannot be determined by arbitrary rules, since it depends mainly upon common sense, logic and experience. 'For weighing evidence there can be no canon. Each case presents its own peculiarities and in each common sense and shrewdness must be brought to bear upon the facts elicited.' (*R v Madhub Chunder* (1874) 21 W.R. Cr. 13 at 19 per Birch J)[53]

And that

> [i]f the law about how to approach proof of facts were comprehensive, all questions of fact, by which I mean all decisions about matters of evidence, including the weight to be given to evidence, would be governed by a rule. But, generally, matters of weight are not governed by bright lines. For the most part, they are left to the good sense of the tribunal.[54]

Imwinkelried argues that this absence of comprehension of the concept of weight is explicable by reference to shortcomings in the training of lawyers.

> The heavy emphasis on threshold admissibility questions in the reported cases certainly contributed to the neglect of weight issues. Moreover, most attorneys receive little or no law school training in evaluating the weight of evidence, including scientific evidence. Most evidence courses and texts concentrate on the admissibility of evidence, its legal sufficiency to sustain various burdens, and substitutes for evidence. The courses and texts completely overlook weight analysis.[55]

So it is that rules about admissibility are more explicit, and more precisely defined, than are rules about the allocation of weight. It follows that any particular regime of evidentiary treatment has necessarily to be expressed, in large part, in terms of rules about admissibility.

---

or about the cogency or usefulness of difference types of evidence...": Wells W A, *Evidence and Advocacy* (Sydney: Butterworths, 1988) 215.

[53] Howard M N(ed), *Phipson on Evidence* (London: Sweet & Maxwell, 15th ed, 2000) at paragraph 6-16.

[54] Kerans R P, *Standards of Review Employed by Appellate Courts* (Edmonton, Alberta: Juriliber, 1994) 76.

[55] Imwinkelried E J, "A New Era in the Evolution of Scientific Evidence – A Primer on Evaluating the Weight of Scientific Evidence" (1981) 23 *William and Mary Law Review* 261-290, 272-273.

This implies that the scrutiny of any such regime requires some appreciation of the structure, characteristics and limitations of the rules about admissibility. What is needed in this context is an understanding of their capacity to facilitate goals of legal fact finding, such as the goal of rational truth identification. An examination of these rules is undertaken from this perspective in section 2.3.1.

### 2.3.1 Admissibility of evidence

#### 2.3.1.1 Overview

As Wigmore observes

> [a]dmissibility signifies that the particular fact is relevant, and something *more—that it has also satisfied all the auxiliary tests and extrinsic policies.* Yet it does not signify that the particular fact has demonstrated or proved the proposition to be proved, but merely that it is received by the tribunal for the purposes of being weighed with other evidence.[56]

Wigmore's characterisation of the concept of admissibility is significant in the present context. It suggests not only that the concept may be used as a mechanism for realisation of the goal of rational truth identification, but that it has other purposes as well. Rules of admissibility are directed to a number of matters, of which truth identification is only one. Examples of the other matters include a number of 'policy' concerns, such as the confidentiality of the lawyer–client relationship, and fairness to the accused in a criminal case. These concerns are addressed by rules that exclude the admissibility of certain lawyer client communications[57] and which exempt an accused from the obligation to give evidence that may incriminate them.[58]

Rules about admissibility determine whether or not material can be used as evidence in particular proceedings and this suggests two respects in which they may promote rational truth identification. They can be used to exclude material that is likely not to

---

[56] Wigmore, Note 5 at §12.

[57] For example: *Evidence Act* 1995 (Cth), ss 118-119.

[58] For example, *Constitution of the United States of America*, Amendments, Article V: "No person ... shall be compelled in any criminal case to be a witness against himself"; *Evidence Act* 1995 (Cth), s 128.

aid the search for truth and to include material that is likely to aid the search for truth. Used in this way, the rules will be directed to determining the capacity that given material may have to aid in "the accurate determination of facts" in the sense described by Weinstein.[59]

This suggests that the development of a regime of evidentiary treatment for particular material will involve a consideration of the structure and content of the existing rules about admissibility and of the necessity (and possibility) of changing that structure or content in some way. The following section describes the principal aspects of the rules. Although the possibility for local variations exists, the aspects that are considered here can be observed in a number of common law jurisdictions. Those aspects are: a principal rule of relevance, a predominant set of rules that exclude material (accompanied by exceptions to those rules) and a less significant set of rules that operate to include evidence or to mandate presumptions about facts.

*A principal rule of relevance*

Relevance manifests itself as the existence of a sufficiently close connection between given material (evidence) and a particular factual subject matter (an issue). For example, s 55(1) of the *Evidence Act* 1995 (Cth) provides that:

> [t]he evidence that is relevant in a proceeding is evidence that, if it were accepted, could rationally affect (directly or indirectly) the assessment of the probability of the existence of a fact in issue in the proceeding.[60]

The connection is used as a basis for the rule that makes all relevant material admissible as evidence (subject to the rules of exclusion) and all irrelevant material inadmissible. In Thayer's words: "[n]one but facts having rational probative value are admissible."[61] Relevance is a primary threshold, and the proposition that relevant material is admissible and irrelevant material is not admissible is the 'principal' rule

---

[59] Weinstein J B, "Some Difficulties in Devising Rules for Determining Truth in Judicial Trials" (1966) 66(2) *Columbia Law Review* 223-246 at 243.

[60] Compare Fed. R. Evid. 401: "Relevant evidence means evidence having any tendency to make the existence of any fact that is of consequence to the determination of the action more probable or less probable than it would be without the evidence."

[61] Thayer, "Presumptions and the Law of Evidence" (1889) 3 *Harvard Law Review* 141-166 at 144-145. See also Wigmore, Note 5 at §9.

of evidence.[62] In Australia, section 56 of the *Evidence Act* 1995 (Cth) enacts this rule in the following terms.

> **Relevant evidence to be admissible**
>
> (1) Except as otherwise provided by this Act, evidence that is relevant in a proceeding is admissible in the proceeding.
>
> (2) Evidence that is not relevant in the proceeding is not admissible.[63]

Fed. R. Evid. 402 also provides a concise statement of the proposition.

> All relevant evidence is admissible, except as otherwise provided by the Constitution of the United States, by Act of Congress, by these rules, or by other rules prescribed by the Supreme Court pursuant to statutory authority. Evidence which is not relevant is not admissible.

The relevance rule is properly regarded as serving a dual function.[64] It is on one hand a mechanism of exclusion. It restricts the volume of material that can be presented in a proceeding to that which is essential, or at least useful, to legal fact finding. The rule also has an inclusionary aspect. To the material that it accepts, the rule provides

---

62 Twining, Note 32 at 114, referring to the rule as the "first rule of admissibility". At one time the 'best evidence' rule was a contender for 'principal rule' status. The conventional formulation of that rule comes from Lord Hardwick's judgment in *Omychund v Barker* (1745) 1 Atk 21 at 49; 26 ER 15 at 33. "The judges and sages of the law have laid it down that there is but one general law of evidence, the best that the nature of the case will allow." The rule, which was applied predominantly to establish requirements for the admission of documents, is said to have lost almost all of its significance, see Brown R A, *Documentary Evidence in Australia* (Sydney: Law Book Company, 2nd ed, 1996) 116-119; Heydon J D, *Cross on Evidence: Sixth Australian Edition* (Sydney: Butterworths, 2000) paragraphs [1465] to [1480]. As Brown observes, the effect of section 51 of the *Evidence Act* 1995 (Cth) is to abolish the rule for the jurisdictions to which the Act applies: Brown at 379. For a contrary view of the importance of the rule, see Nance D A, "The Best Evidence Principle" (1988) 73 *Iowa Law Review* 227-297.

63 The relationship between sections 55 and 56 of the *Evidence Act* 1995 (Cth) and the pre-existing common law has been considered by the High Court of Australia in a number of recent decisions. In *Smith v R* (2001) 206 CLR 650 a majority of Gleeson CJ, Gaudron, Gummow and Hayne JJ held, on the facts, that the evidence under consideration was not relevant to the proceedings which were the subject of the appeal (at 656). McHugh J expressed the view in *Papakosmas v R* (1999) 196 CLR 297, that "the statutory concept of relevance does not involve any real departure from the common law" (at 312). Gleeson CJ expressed a similar view in *Goldsmith v Sandilands* (2002) 190 ALR 370, noting that the definition of relevance in s 55 of the Act "is not materially different from that given by Sir James Stephen in his *Digest of the Law of Evidence*" (at 371). In other respects, however, McHugh J has regarded the Act as having "made substantial changes to the law of evidence" with the result that in the context of interpreting and applying the provisions of the Act "reference to pre-existing common law concepts will often be unhelpful": *Papakosmas v R* (1999) 196 CLR 297, 324. Other departures from the common law by the Act were noted in *Graham v R* (1998) 195 CLR 606 at 610, per Gaudron, Gummow and Hayne JJ.

64 This point is also made in Heydon, Note 62 at paragraph [1485], although in a slightly different context.

a measure of endorsement.[65] Relevant material is and, by the definition that the rule applies, must be of some value to legal fact finding.

An important consequence of this endorsement is the need for any rule that seeks to exclude relevant material from use to carry with it an appropriate justification. Even if such a rule is centuries old—as some are—it cannot be regarded as self-supporting. This point has been demonstrated in the results of a number of recent evidence law reform initiatives.[66]

*Predominant rules of exclusion*

The main categories that the rules of exclusion cover are hearsay, opinion, character or tendency, credibility and privilege.[67] The rules of exclusion are predominant because they comprise a large portion of the rules of admissibility. They also possess a prescriptive character and they are largely inflexible.[68] These attributes give them a 'presence' that overshadows other aspects of fact finding, most notably questions about the weight of evidence.[69]

The identification of a single explanation or rationale for the rules of exclusion is an elusive aim.[70] Stephen's view, which sought to explain all exclusionary rules of evidence as being applications of considerations of relevance,[71] has not prevailed.[72] The rules exist for reasons that are discrete from considerations of relevance, although what those reasons are is difficult to state. Thayer's 'child of the jury' thesis is one attempt to do this.

---

65 Twining W L, "What is the Law of Evidence" in Twining, Note 1, 178-218 at 190.

66 One example is the extent to which such reform initiatives have dramatically altered the operation of the rule against hearsay. This is discussed below.

67 See for example, Australian Law Reform Commission, Report 26, *Evidence (Interim)* (Canberra: Australian Government Publishing Service, 1985) volume 1, at paragraph 38.

68 In the sense that the exceptions apply only under particular prescribed conditions. There is no general discretion to admit evidence in spite of the applicability of a rule of exclusion.

69 Twining, Note 65 at 191.

70 For an account of attempts to do this see Twining, Note 1; Twining, Note 65 at 185-189.

71 Hearsay material, for example, is excluded under this theory simply because it is deemed not to be relevant: Stephen, J F, A *Digest of the Law of Evidence* (London: Macmillan, 1893) 15. See also Twining, Note 1 at 56.

72 Twining, Note 65 at 188.

> Some things are rejected as being of too slight a significance, or as having too conjectural and remote a connection; others, as being dangerous in their effect on the jury, and likely to be misused or overestimated by that body; others as being impolitic, or unsafe on public grounds; others on the bare ground of precedent. It is this sort of thing, as I said before,—the rejection on one or another practical ground, of what really is probative—which is the characteristic thing of the law of evidence; stamping it as the child of the jury system.[73]

Despite its prominence, this view has not passed without challenge.[74] As Damaska points out, a more contemporary view regards common law rules of exclusion as being of different parentage; they are "first and foremost the child of the adversary system."[75] As discussed in chapter one, the rules may also be seen as operating within an adversary system as exceptions to a general principle of freedom of proof.[76]

As difficult as it is to explain the rules of exclusion at any point in time, it is even more difficult to develop an explanation that accounts for changes in attitudes to those rules. Recent expressions of those changes in attitude generally favour a relaxation of the effect and scope of the exclusionary rules. In 1997, for instance, the Law Commission for England and Wales endorsed the view that "as a general rule all [relevant] evidence should be admissible unless there is a *good* reason for it to be treated as inadmissible."[77] Although recently expressed, this position is not new. A parallel sentiment was expressed over one hundred and forty years ago in the judgment of Cockburn CJ in *R v Birmingham Overseers* (1861) B & S 763, 767; 121 ER 897, 899 (QB).

> People were formerly frightened out of their wits about admitting evidence less juries should go wrong. In modern times we admit evidence and discuss its weight.

---

73 Thayer, Note 9 at 266 (citation in Twining, Note 65 at 190).

74 Twining, Note 1 at 58.

75 Damaska, Note 18 at 2. This view casts doubt upon the contention that the decline of the significance of the jury, a phenomenon that is referred to in contemporary critiques of the common law rules of evidence (see in this context Damaska, Note 18 at 6; Twining, Note 65 at 187) is, of itself, a sound basis for revision of some of the long-standing rules of exclusion.

76 An interpretation attributed to Thayer: Twining, Note 65 at 194.

77 Law Commission, *Evidence in Criminal Proceedings Hearsay and Related Topics* (London: H.M.S.O., 1997) paragraph 1.4 citing Scottish Law Commission, *Report 149: Evidence: Report on Hearsay Evidence in Criminal Proceedings* (Edinburgh: H.M.S.O., 1995) paragraph 2.3.

Whether this was an entirely accurate assessment of practices then or in the following years is not relevant for present purposes. There has been, in any event, a general trend toward relaxation of standards of admissibility which can be illustrated by the example of the rule against hearsay. The rule against hearsay prohibits the use of a statement that is made other than during witness testimony in particular proceedings to prove the truth of matters asserted by that statement.[78] The most recent revisions of the hearsay rule must be far outside the scope of what was thought possible or prudent in 1861. To date, section 1(1) of the *Civil Evidence Act* 1995 (UK) appears to mark the high point of reform with the simple edict that "[i]n civil proceedings evidence shall not be excluded on the ground that it is hearsay."

Other contemporary reforms have not gone as far, but they have nevertheless effected substantial changes to the premises for the operation of the rule. At one time the rule was applied as a rigid prohibition upon the reception of material, subject only to exceptions that focused upon particular qualities that could be associated with that material[79] or upon specific circumstances that justified its displacement. In some jurisdictions, this approach has now changed substantially. Sections 63-66 of the *Evidence Act* 1995 (Cth) are one example of such change. While section 59(1) of the Act is a standard expression of the exclusionary rule, the structure of the exceptions in sections 63-66 significantly alter the scope of the rule. Those provisions apply to 'first hand hearsay'[80] that is sought to be used in both criminal and civil proceedings. They also apply to cases in which the maker of the hearsay statement is available to give evidence in the proceedings as well as in cases in which the maker is not available to give evidence in the proceedings.[81] Although the extent of the exception varies in each of these four cases, the combined effect of the provisions is to alter the rule from one that applies automatically, to one that will apply only under certain conditions.

---

78 Various justifications have been claimed for it, although the evaluation of these is irrelevant for present purposes. For one discussion see: Law Commission, Note 77 at paragraphs 3.1–3.38.

79 For example, that the material had been prepared in the course of a regularly conducted business activity. See for example *Evidence Act* 1995 (Cth), s 69; Fed. R. Evid. 803(6).

80 Which is a representation made by a person who has personal knowledge of a fact as opposed to a representation made by a person who was merely the recipient of a representation by another person: see *Evidence Act* 1995 (Cth), s 62.

In civil proceedings, those conditions relate, in the first instance, to the availability of the maker of the hearsay representation rather than to the circumstances of production of the material. Under section 64 of the *Evidence Act* 1995 (Cth), the exception to exclusionary rule is available in civil proceedings even where the maker of the representation is available to give evidence in the proceedings, but does not do so, merely because

> it would cause undue expense or undue delay, or would not be reasonably practicable, to call the person who made the representation to give evidence.

What determines the availability of the exception under this provision are the circumstances of availability of the maker of the representation *vis-à-vis* the proceedings, rather than the circumstances in which the representation was made, or the form in which it is provided as evidence.[82] It is, however, the latter considerations that have been relied upon as a basis for the making of an exception to the exclusionary rule in earlier contexts. This is evidenced by the following comments of the Advisory Committee on Rules, which were made in respect of Fed. R. Evid 802 and 803.[83]

> The present rule proceeds upon the theory that under appropriate circumstances a hearsay statement may possess circumstantial guarantees of trustworthiness sufficient to justify non-production of the declarant in person at the trial even though he may be available. The theory finds vast support in the many exceptions to the hearsay rule developed by the common law in which unavailability of the declarant is *not* a relevant factor.[84] (Emphasis added.)

The significance of contemporary exceptions to the rule against hearsay, such as those provided for by *Evidence Act* 1995 (Cth), is that they focus upon questions of convenience, cost and potential delay in the proceedings, rather than the

---

[81] See *Evidence Act* 1995 (Cth), s 63 (civil, maker not available), s 64 (civil, maker available), s 65 (criminal, maker not available) and s 66 (criminal, maker available).

[82] Under section 64(2), evidence of the representation may be "given by a person who saw, heard or otherwise perceived the representation being made" or in a document that contains the representation.

[83] These rules are, respectively, an enactment of the exclusionary rule and its exceptions for the United States federal jurisdictions.

[84] Federal Rules of Evidence, *Historical Notes and Legislative Commentary* <http://www2.law.cornell.edu/cgi-bin/foliocgi.exe/fre/query =[jump!3A!27acrule802!27]/doc/{@774}> (visited 2 June 2003).

circumstances of production (and trustworthiness) of the material in question.[85] These provisions were not, however, the first to exhibit this change in emphasis. Section 55 of the *Evidence Act* 1958 (Vic) and section 92 of the *Evidence Act* 1977 (Qld) also introduce mechanisms for the reception of hearsay in civil cases[86] in which the maker of the hearsay representation is not available.[87] Brown notes that this mechanism is borrowed from the *Criminal Evidence Act* 1965 (UK).[88]

Clear in direction though this course of change may be, at least in relation to the rule against hearsay, it has not been without limitations. Some traditions continue to assert themselves. For instance, the English, Australian Commonwealth and New South Wales and Queensland approaches to the hearsay rule still exhibit a differential approach to the rule for civil and criminal proceedings. This differentiation has been said to be motivated by the "different nature and objectives of the civil and criminal trial".[89] Moreover

> [b]oth are adversary systems, but the former is a system for resolving disputes and the latter is an accusatorial system in which the State accuses the defendant of breaking the law. Individual liberty and civil liberties are at stake in criminal trials.[90]

The demarcation between civil and criminal litigation remains a settled part of the contemporary landscape in which evidence law must operate. In 1999, the Western Australian Law Reform Commission observed in this context that "[w]hile there is ample reason to move towards uniform laws of evidence throughout Australia, the distinction between civil and criminal matters remains important."[91]

---

85 Examples of exceptions to the rule which *do* focus upon the issue of circumstances of production include: the 'business records' exception (as to which see Note 79), contemporaneous statements (see for example *Evidence Act* 1995 (Cth), s 72) and admissions against the interest of the maker (see for example *Evidence Act* 1995 (Cth), s 81).

86 A slightly different approach is taken when the material is to be used in criminal proceedings: *Evidence Act* 1977 (Qld), s 93.

87 The *Criminal Justice Act* 1988 (UK) is also of note for its early adoption of this kind of approach, see in particular s 23 and the discussion at Law Commission, Note 77 at paragraph 2.14.

88 Brown, Note 62 at 222 and 267-279.

89 Australian Law Reform Commission, *Report 38, Evidence* (Canberra: Australian Government Publishing Service, 1987) Report Summary, paragraph 9.

90 Australian Law Reform Commission, Note 89 at Report Summary, paragraph 9.

91 Law Reform Commission of Western Australia, *Review of the Criminal and Civil Justice System* (Perth: The Commission, 1999) paragraph 20.3.

How far the process of relaxation of the exclusionary rules will progress is difficult to predict. The hearsay rule has, so far, been the subject of the most substantial revisions. Other rules of exclusion have been less affected.[92]

*Subjacent rules of presumption or inclusion*

Rules that deal with presumptions operate to displace altogether the need for proof of a particular matter. To this extent they can be treated as belonging to a class of rules that is outside the scope of rules about admissibility. This is the analysis adopted by Wigmore, who suggested that they (and certain other rules) were located at

> the borderline of what is in strictness the law of evidence. They involve and rest upon certain larger aspects of procedure that are independent of the evidential material.[93]

Rules about presumptions are mentioned here for two reasons. First, there are contexts in which they can operate to provide a foundation for the operation of exceptions to rules of exclusion. To this extent, they have the potential to be an important aspect of any approach to evidentiary treatment. Second, beyond this general significance, presumptions have a particular importance for the subject matter with which this thesis is concerned. This importance is due to their use by one of the principal existing approaches to the evidentiary treatment of computer-produced material.

'Rules of inclusion' are rules about admissibility properly so called. These rules operate to render particular material admissible as evidence. In doing this, they operate to displace the rules of exclusion that would otherwise apply to given material. Rules of inclusion are relatively rare in the law of evidence, and they receive remarkably little attention in commentaries about the law of evidence. Like

---

92 An example is the rule relating to character evidence in criminal cases. See The Law Commission, *Evidence of Bad Character in Criminal Proceedings* (London: H.M.S.O., 2001). As appears from the discussion in paragraphs 1.12 to 1.23 of the report, although changes to the law are recommended, these are by no means uniform changes that relax the exclusionary rule. See also the discussion at paragraphs 6.3 to 6.7, which argues strongly against the automatic admission of a defendant's prior criminal record in all criminal cases, even though that course is conceded to be one that would 'simplify' trials.

93 Wigmore, Note 5 at §3. For a contemporary treatment of presumptions see Heydon, Note 62 at paragraphs [7235-7320].

rules about presumptions, rules of inclusion also have the potential to alter in a substantial way the operation of the more prominent rules of exclusion. They operate to make material admissible despite the existence of rules of exclusion that would otherwise apply to the material in question. As with rules about presumptions, these rules too have been employed in existing approaches to the evidentiary treatment of computer-produced material.[94]

2.3.1.2 Discussion

The description of the concept of, and rules about, admissibility in section 2.3.1.1 has a number of implications in the present context. The first is that rules about admissibility can properly be seen as a mechanism for the attainment of the goals of legal fact finding. As such, particular rules should not be seen as sacrosanct or sustaining merely by reason of their existence. The development of an appropriate regime for the evidentiary treatment of particular material may—and in some instances will—require alterations to certain rules about admissibility.

This means that an approach to the evidentiary treatment of particular material must do more than merely attempt to 'fit' that material within existing rules of admissibility. The objective in formulating any such approach must be related to the goals of legal fact finding, not to the structure of the rules of evidence as it may appear at any given point in time. The position might be different if the rules about admissibility *were* sacrosanct or at least immutable, but they are not. The question that must be considered here is not whether particular material is or is not admissible under existing rules. It is instead whether, having regard to the goals of legal fact finding, particular material ought to be admissible and if so, under what conditions. The answer to this question may indicate that there is a need to alter the operation of existing rules about admissibility or, perhaps, to introduce new rules.

---

[94] Approaches to evidentiary treatment that employ rules of inclusion and rules about presumptions are examined in chapter five.

These considerations properly locate rules about admissibility as a mechanism that can be adapted to facilitate the rational identification of truth.[95] Because the rules have effect to either permit or exclude the use of particular material, such adaptation will be directed to the question of what ought to be excluded from use and what ought to be permitted to be used. Having regard to the considerations that were canvassed in section 2.2, it is clear that the likelihood that given material will reflect an underlying empirical truth is a central consideration in this context. Beliefs about this likelihood must, for the reasons given in section 2.2, have a rational foundation.

Attempting to discriminate between probative and other material introduces a fundamental problem. Criteria that are imposed to effect the necessary discrimination might not always operate in the manner that is desired. Probative material may be excluded, and material that has no probative value may be admitted. This is the case particularly when documentary material, such as computer-produced material, shows no obvious signs of error on its face. This implies that a considered attempt at discrimination must be made, and that it must have a rational basis.

The dangers that are involved in embarking upon any attempt at discrimination are, as has been said, twofold. There is on the one hand a risk that given conditions will be too restrictive, leading to the loss of probative material. There is also a risk that given conditions will be too liberal, leading to the introduction of material that is not appropriate for use in an attempt to discover the truth by rational means.

## 2.4 Constraints

Sections 2.2 and 2.3 of this chapter have examined, respectively, the goal of rational truth identification and the principal mechanism that is available to promote the attainment of that goal. This section explores the principal constraints that apply to the formulation and application of a regime of evidentiary treatment. The premise for this exploration is the proposition that legal fact finding in common law jurisdictions

---

[95] Subject to the constraints considered below at section 2.4.

employs particular 'ways of doing things' that impose boundaries upon the operation of existing (and proposed) regimes of evidentiary treatment.

These constraints are paradigms that exist as overlays upon the goals of legal fact finding, and account must be taken of them in the present context for two reasons. First, they have the potential to prevent the use of particular means of fact finding that may otherwise be indicated. Second, they are sufficiently entrenched in common law culture that the goal of rational truth identification must be pursued within the confines that they impose. The two matters of constraint that are most relevant to the present context are the adversarial mode of trial, and the resistance to attempts to prescribe formal mechanisms for decision making about facts.

### 2.4.1 The adversarial mode of trial

#### 2.4.1.1 Description

An adversary system is one

> in which procedural action is controlled by the parties and the adjudicator remains essentially passive. In the fact-finding domain, this implies that the litigants and their counsel decide what facts shall be subject to proof.[96]

Other accounts paint a picture that includes party antagonists and the court as umpire.[97] It is fundamental that the process revolves around a dispute in which the parties can and do take differing positions on questions of fact and law.[98] It has been suggested that

> [t]he common law courtroom trial is a forum in which arguments of the disputing parties are pitted against each other. ... The defining characteristic of adjudication in common law systems is its adversarial nature, reflected in the practice and culture of litigation.[99]

---

96 Damaska, Note 18 at 74.

97 Twining, Note 65 at 183, referring to Jacob J I, *The Fabric of English Civil Justice* (London: Stevens & Sons, 1987) 156.

98 Law Reform Commission of Western Australia, Note 91 at paragraph 6.1.

Yet despite this "defining characteristic", it is the case that a 'pure' adversary system cannot exist in a legal context. This is due (at least) to the fact that there must be a non-adversarial method of identifying the 'winner' of litigation.[100] Since litigation is not conducted by means of deadly or overwhelming physical force, but by means of reason and argument, that method must involve more than the mechanical and passive observation of some patent, objective indicia of victory or defeat. It must involve some active participation by the adjudicator in assessing and deciding the outcome.[101]

A 'pure' adversary system is also undesirable in the legal context because there is a fundamental incompatibility between processes that are wholly adversarial and processes that are regulated. Control by law is antithetical to conflict in pure form: *inter arma enim silent leges* (in time of war, the law falls silent).[102] Whilst this does not mean that an unregulated contest could not be utilised as a means of determining disputes, there are persuasive arguments why it would be undesirable to do so. After conducting an extensive review of the operation of the system of civil justice in the United Kingdom, Lord Woolf came to the view that there was a "need to bring the uncontrolled features of the adversarial system under proper discipline."[103] Lord Denning may well have seen the same need, having earlier expressed a view that "it is not possible to dispense with the rules of evidence and procedure altogether. Rough justice may become so rough that it ceases to be justice."[104]

Adversarial fact finding can be distinguished from its principal alternative: the inquisitorial mode of trial. What distinguishes the adversarial mode from the

---

99 Australian Law Reform Commission, *Issues Paper 20: Review of the Adversarial System of Litigation* (Canberra: Australian Government Publishing Service, 1997) paragraph 2.9.

100 Jolowicz J A, *On Civil Procedure* (Cambridge: Cambridge University Press, 2000) 175.

101 Jolociwz, Note 100 at 175.

102 Originally from Cicero's oration in defence of Milo at the latter's trial at Rome in 52 B.C.E. The popular form of the maxim transposes the phrases in the original text. See, for example, Colson, F H (ed) *Cicero: Pro Milone* (Bristol: Bristol Classical Press, 1980) 5.

103 Woolf H K, *Access to Justice, Final Report to the Lord Chancellor on the Civil Justice System in England and Wales* (London: H.M.S.O., 1996) at chapter 9, paragraph 1.

104 Denning A, *Freedom Under the Law* (London: Stevens & Sons, 1949) 89-90. The Australian Law Reform Commission considered a range of the issues and problems that arise in the context of an 'adversarial system' of justice. See: Australian Law Reform Commission, *Report 89, Managing Justice: A Review of the Federal Civil Justice System* (Canberra: Australian Government Publishing Service, 2000), in particular at paragraph 5.152. Similar matters were canvassed by the Law Reform Commission of Western Australia in its enquiry. See Law Reform Commission of Western Australia, Note 91 at paragraph 18.3.

'inquisitorial' mode,[105] is the control that the parties (as opposed to the adjudicator) have over information gathering.[106] It is in this information gathering phase—rather than the subsequent decision making phase—that adversarial fact finding differs most from inquisitorial fact finding. In this context, the parties do not collaborate; they participate in information gathering in an oppositional manner.[107] In addition to efforts to gather material that is most favourable to its own case, a party will attempt, as far as possible within the relevant rules, to ensure that information that is favourable to their opponent is not admitted as evidence.[108] In the inquisitorial mode the adjudicator is, by contrast, responsible for both information gathering and decision-making.

This relegation of information collection and presentation is supplemented by the right of the parties to contend, themselves or by an advocate, for particular factual conclusions that should be reached on the basis of the evidence that is admitted, and to present argument in support of those contentions. It is this feature of opposing argument that encapsulates much of what is termed the 'adversarial culture' that this mode of trial is said to engender.[109]

The picture of the adversarial mode of trial is completed by the imposition of criteria that allow the 'winner' of litigation to be identified. These criteria comprise an onus

---

Commission of Western Australia in its enquiry. See Law Reform Commission of Western Australia, Note 91 at paragraph 18.3.

105 It has been argued that this mode of enquiry also cannot be practised in a pure state, since the existence of a dispute that calls for fact finding implies that the parties will have opposing interests: Jolociwz, Note 100 at 175.

106 Except with respect to matters that are the subject of judicial notice, or which are presumed as a matter of law. See generally Heydon, Note 62 at paragraphs [3005], [7235]. As to arguments for and against greater application of the personal knowledge of judges in litigation see Jolociwz, Note 100 at 243-268.

107 Damaska suggests a further division, a "fission of fact-finding into two contrary cases": Damaska, Note 18 at 78.

108 This might imply that the parties, in addition to gathering favourable evidence, would also destroy or tamper with that which is unfavourable, or perhaps that they might fabricate evidence altogether. In a true adversarial environment this might follow, but conduct of this kind is prohibited in the common law jurisdictions and usually carries criminal sanctions. The authority for imposition of these sanctions may be a specific statutory provision, see for example *Crimes Act* 1914 (Cth), s 36 (fabricating evidence); s 39 (destroying evidence); *Crimes Act* 1900 (NSW), s 327 (perjury); 18 U.S.C. §1512 ('witness tampering'); 28 U.S.C. §41 (perjury). Conduct of this kind may also constitute the more general offence of perverting the course of justice: see Miller C J, *Contempt of Court* (Oxford: Oxford University Press, 3rd ed, 2000) 8. It may also found an independent (civil) cause of action in some jurisdictions for the tort of 'intentional spoliation of evidence', see for example: *Cedars-Sinai Medical Center v Superior Court* 18 Cal 4th 1, 74 Cal Rptr 2d 248 (1998).

or burden of proof and a standard of proof. In a civil case, the moving party bears the onus of proving 'on the balance of probabilities' facts that establish a cause of action. In a criminal case, the moving party bears the onus of proving 'beyond reasonable doubt' facts that establish the commission of an offence. In both cases,[110] the burden is subject to whatever assistance can be obtained from available presumptions.[111] The responding party bears no onus, except in respect of matters of defence that it wishes to raise.[112] It may prevail in a given case simply because the moving party fails to discharge the onus of proof that is allocated to it.

2.4.1.2 Implications for truth identification

The adversarial mode of trial impacts upon the goal of rational truth identification in three principal respects. First, whilst the identification of truth is related to the intended results of an adversarial trial, there are respects in which it is treated as only an *incident* of the trial process. Whilst rational truth identification is plainly a principal goal of legal fact finding, it is not the expressed goal of an adversarial trial. Second, freedom to influence the execution, scope and to an extent the outcome of the process is given to parties who may have interests that do not coincide with the discovery of truth. Third, notwithstanding these two limiting factors, there is very little scope for a new regime of evidentiary treatment (or any other revision of the law of evidence) to depart from the adversary paradigm. These matters are considered in turn.

---

109 See Australian Law Reform Commission, Note 99 at paragraphs 2.23, 11.

110 For an explicit statutory expression of the standards see *Evidence Act* 1995 (Cth), s 140(1) (civil standard) and s 141 (criminal standard). In the former case, s 140(2) goes on to prescribe certain matters that the Court must take into account in determining whether the standard has been met. In the United States the criminal standard of proof has been held by the Supreme Court to be an aspect of the 'due process' requirement of the fifth amendment to the constitution: *In re Winship* 397 U.S. 358, 361-362 (1970).

111 See for example Heydon, Note 62 at paragraph [7095].

112 In criminal cases, the defendant normally bears a lighter burden with respect to matters that comprise defences to a particular charge. See for example: *Criminal Code Act* 1995 (Cth), Schedule, s 13.3.

*A distinction between the goals of the adversarial trial and the goals of legal fact finding*

The use of an onus and standard of proof that do not refer directly to the demonstration of truth has a profound impact upon the goal of rational truth identification. In a real sense, the identification of truth is not an ultimate goal of the adversarial trial. Rather, as Dawson J observed in *R v Whithorn* (1983) 152 CLR 657, 682

> [a] trial does not involve the pursuit of truth by any means. The adversary system is the means adopted and the judge's role in that system is to hold the balance between the contending parties without himself taking part in their disputations. It is not an inquisitorial role in which he seeks himself to remedy the deficiencies of the case on either side.[113]

What seems to be a more immediate priority is the making of decisions about disputes in a manner that is generally palatable. Goodpaster argues in the context of criminal cases that "[t]he essential criminal trial need is to generate decisions in a manner and with results that the community can accept and use".[114] Such an object may well conflict with the identification of truth[115] because it gives priority to dispute resolution as an ultimate purpose.[116] As has also been argued, the

> adversarial system seems to put so much of the search for truth at risk, and as an experienced commentator has recently said, 'the deepest tyranny in these matters consists in being determinedly ignorant of the truth.'[117]

It is noteworthy that, despite such concerns, debates about the efficacy of the adversary system tend to focus more closely upon other potential shortcomings. For instance, Lord Woolf's review of the problems that were said to be associated with

---

113 This passage is cited in Law Reform Commission of Western Australia, Note 91 at chapter 6, 'Introduction'. Nor is it the role of the court to seek to uncover information that a party seeks to withhold even if it would assist that party: *Air Canada v Secretary of State for Trade* [1983] 2 A.C. 394, 434 per Lord Fraser.

114 Goodpaster G, "On the Theory of the American Adversary Criminal Trial" (1987) 78 *Journal of Criminal Law and Criminology* 118-153, 152 and see also at 118, referring to the need to "generate decisions about disputes."

115 Macchiarola F J, "Finding the Truth in an American Criminal Trial: Some Observations" (1997) 5 *Cardozo Journal of International and Comparative Law* 97-111, 101.

116 Damaska argues that this is the principal objective: Damaska, Note 18 at 120-124.

117 Macchiarola, Note 48 at 111, referring to Anastaplo G, "On Crime Lawyers and O.J Simpson: Plato's Gorgias Revisited" (1995) 26 *Loyola University Chicago Law Journal* 455, 469.

the system of civil justice in the United Kingdom identified matters of cost, delay, and complexity as the paramount concerns.[118] No mention was made of any failure in the system to decide cases on the basis of the true facts that underlie disputes. When concerns about truth finding are mentioned, it is the partisan culture of adversary procedure that attracts comment. Delisle, for instance, observes that

> [w]hether the adversary method will more closely approximate the truth is certainly open to question. The lawyer is trained to seek success for his client; to win the game. His goal is to present the best picture of his client's position and not the most complete picture at his disposal. Also the adversary system presupposes some equality between the parties; when this is lacking the 'truth' becomes too often simply the view of the more powerful.[119]

*Party interests*

Discovery of truth is important to legal fact finding, but will the discovery of truth always coincide with the interests of all of the parties to an adversarial proceeding? Such proceedings are, in most instances, commenced against the backdrop of a dispute as to the existence of particular facts. The point of initiating proceedings is to seek an application of substantive law that is congruent with a particular version of those disputed facts. There is, for the opposing party, a motivation to dispute the moving party's version of the facts. This motivation is more than merely abstract or intellectual.

In the civil sphere, a grant of the remedies to which the moving party lays claim equates to a cost to the responding party.[120] Whether in the prospect of being ordered to pay damages or of becoming the subject of an injunction, the economic interests of the responding party are at risk. The responding party is therefore motivated to avoid

---

[118] Woolf H K, *Access to Justice, Interim Report to the Lord Chancellor on the Civil Justice System in England and Wales* (London: H.M.S.O., 1995) chapter 3, paragraph 1.

[119] Delisle R J, *Evidence: Principles and Problems* (Toronto: Carswell, 1984) 2. For further comments about inequality of party resources see Law Reform Commission of Western Australia, Note 91 at paragraph 7.14. Compare Jackson's argument that "there remains a deep commitment to ... the idea of the contested trial as the paradigm example of a procedure designed to maximize accuracy": Jackson J D "Analysing the New Evidence Scholarship: Towards a New Conception of the Law of Evidence" (1996) 16(2) *Oxford Journal of Legal Studies* 309-328 at 327.

[120] Administrative law proceedings fall outside this general proposition, at least on the 'public' (decision maker's) side of the record. What is at stake in such proceedings is the continuing validity of a course, or proposed course, of governmental action. In some cases this may have revenue

an outcome that involves the grant of any remedy against it. The accused in a criminal case inevitably seeks also to avoid the adverse outcomes that can apply in those proceedings. It is obvious enough that a defendant who does not plead guilty wants to avoid—or to minimise as far as possible—liability to criminal penalties.

It is natural for such parties to prefer an outcome that favours their interests, irrespective of whether that result is congruent with the truth. Moral imperatives or simple pragmatism might encourage some parties to concede the validity of the claims or accusations that have been made against them. If this happens, it is likely that a course that is appropriate to those views would be pursued, such as a compromise of the proceedings, or the entering of a plea of guilty. Those courses dispense with the need for a fact finding exercise to be undertaken.

Parties that are not influenced by moral or practical concerns might press on in the hope of securing an outcome that is advantageous to them, even if it is at the expense of the truth. It is conceivable that this behaviour might, at least in civil cases, be exhibited by the moving party too. This is because the point of bringing such proceedings is to secure a remedy to which the moving party presumably attaches some value.[121] The significance of this point emerges from the probability that the truth is likely to favour one of the parties in a large number of cases. In those cases the interests of that party will be coincident with the goal of truth identification, but not so the interests of the other party. In other cases, the truth may lie rather more between the interests of the opposing parties, instead of directly favouring one or the other.

---

implications, but these will be felt collectively, rather than directly by the relevant government officers whose action might be reviewed.

[121] Such behaviour is less likely in criminal cases, because the duty of a prosecutor in a criminal case is not to seek to secure a conviction at all costs but "to present the case against the accused fairly and honestly": *King v R* (1986) 161 CLR 423 at 426 per Murphy J. Aspects of this duty include the obligation to disclose to the defence material that may be harmful to the prosecution case. For a discussion of the duty of disclosure in Australia and the United Kingdom see Hinton M, "Unused Material and the Prosecutor's Duty of Disclosure" (2001) *Criminal Law Journal* 121-139. For a recent consideration of this duty by the High Court of Australia see *Grey v R* (2001) 184 ALR 593. In limited circumstances, the duty of a prosecutor to present a case fairly may have an impact upon decisions about whether or not a particular witness should be called. This aspect of the prosecutor's duty will have application if, "when viewed against the conduct of the trial taken as a whole", the decision not to

This potential for divergence between the interests of the parties and the identification of truth is important because the adversarial trial is influenced to a degree by the way in which the parties participate in it. Even within the confines of what is permissible[122] a party can exercise choices in the course of participating in given proceedings. It may, for instance, choose to tender particular material as evidence or to refrain from doing this and it may choose to object to the tender of particular material by an opponent. In some jurisdictions it may choose to waive the application of certain parts of the law of evidence with respect to particular material.[123] It may choose to employ a particular depth of cross-examination of a given witness or to employ no cross-examination.[124]

The adversary system provides to a party who is motivated to obscure the truth certain scope to act upon that motivation. More importantly, there are respects in which the party may act legitimately and lawfully to do so. This is an inevitable consequence of the fact that parties with opposing interests are afforded a degree of control over the conduct of the fact finding process. Whilst a party's conduct in a given case may not be particularly effective, it is the case that this feature of the adversary system has a potential to conflict with the goal of rational truth identification.

*Prospects for departure from the adversarial paradigm*

The implications of the two matters that have just been examined might suggest that a regime of evidentiary treatment should seek to alter the basic adversarial structure under which legal fact finding is undertaken in common law jurisdictions. The reason why such a course is not open is that there is a strong affinity for an adversarial mode of trial in common law jurisdictions.[125] So much is clear from three recent reviews of common law systems of justice in the Commonwealth of Australia,[126] Western

---

call a particular witness is "seen to give rise to a miscarriage of justice": *R v Apostilides* (1884) 154 CLR 563 at 575 per Gibbs CJ, Mason, Murphy, Wilson and Dawson JJ.

122 That is, excluding unlawful conduct of the kind referred to at Note 108.

123 See for example: *Evidence Act* 1995 (Cth), s 190(1). See also s 191 as to the ability of a party to agree to facts, and the consequences of such agreement.

124 These choices are subject to relevant ethical constraints upon the lawyer conducting the cross-examination.

125 Albeit that what is employed in practice is, as has been discussed, not a 'pure' adversary system.

126 Reported in Australian Law Reform Commission, Note 104.

Australia,[127] and the United Kingdom.[128] Each of these reviews has recommended retention of the adversary system.[129]

The report of the most recent of these reviews[130] contains a sustained case for retention of the adversary system. The matters that are canvassed in support include the elusiveness of an "adversarial-non adversarial construct" as a basis for formulating change,[131] a convergence between adversarial and inquisitorial approaches to civil justice,[132] considerations of culture and tradition,[133] and possible constitutional constraints upon departure from the adversarial mode.[134] None of these matters is really compelling. Their importance lies rather in the fact that they have been marshalled in support of a sentiment that recoils unmistakably from abandonment of the adversary system.

What is important for present purposes is not so much why the adversary system commands such loyalty. It is enough that it does. Legal fact finding has to be undertaken, for the time being at least, within its constraints. Regimes for the evidentiary treatment of particular material must, if they are to find support, be fashioned in a way that does not require the alteration of those constraints. This is the case even though departure from an adversary system might be thought likely to remove potential impediments to the discovery of truth, such as the matters that have been considered in this section or matters that may arise in the context of the treatment of particular kinds of material.

It may be then that the least rigorous of the matters raised—the consideration of tradition—best illustrates why the adversary system will not easily be displaced in common law jurisdictions in the foreseeable future.

---

127 Reported in Law Reform Commission of Western Australia, Note 91.

128 Reported in Woolf, Note 103 (final) and Woolf, Note 118 (interim).

129 As Jolowicz notes, although the Woolf reports contain an exceptionally sharp criticism of the adversary system, explicit mention is made in the final report of a desire not to abandon that system: Jolowicz, Note 100 at 388. Jolowicz nonetheless questions whether the reforms that arose out of the Woolf reports have not had a significant impact upon the adversary system: at 389.

130 By the Australian Law Reform Commission: Note 104.

131 Australian Law Reform Commission, Note 104 at paragraph 1.112.

132 Australian Law Reform Commission, Note 104 at paragraph 1.126-1.130.

133 Australian Law Reform Commission, Note 104 at paragraph 1.132-1.134.

134 Australian Law Reform Commission, Note 104 at paragraph 1.123-1.124.

> [T]he best argument in favour of an adversary process is pragmatic. The process is not divinely inspired nor are all others essentially corrupt; it is simply our tradition and it probably is not worth trying to eradicate it.[135]

Against this, Jolowicz questions whether the prospect of change will always be so distant. He suggests that the 'case management' reforms that have been implemented in the United Kingdom give judges a platform from which they can and will assert a more active role in the conduct of proceedings.

> It is, however, difficult to believe that the old philosophy of the adversary system and the refusal to acknowledge that there may be value in a judicial attempt to find the truth, can survive much longer ... Is it too far fetched to suggest that future judges will take advantage of the opportunity thus provided to play a more active role in the preparation of the trial, over and above the management role that is to be cast upon them?[136]

For the present at least it seems that regimes of evidentiary treatment must be formulated on the basis that they must exist within the framework that is referred to as 'adversarial' and which exhibits the kinds of characteristics (and shortcomings) that have been considered in this section.

### 2.4.2 The process of decision making about facts

Legal fact finding involves the making of decisions about disputed facts. Those decisions are expressed in terms that are congruent with the criteria for factual findings in a particular jurisdiction. In common law jurisdictions, those criteria relate to the applicable onus and standard of proof.[137] The pathway to this final outcome can be considered to involve the following stages.

(a) The genesis and definition of a factual dispute, by means of the institution of proceedings and, in civil cases, the exchange of pleadings and particulars.

---

135 Cromwell T, "Dispute resolution in the Twenty-first Century" in *Canadian Bar Association Systems of Civil Justice Task Force* (Toronto: Canadian Bar Association, 1996) 90-91.

136 Jolowicz, Note 100 at 385.

137 See section 2.4.1.

(b) The offer of material to the fact finder by the parties.

(c) The admission of some portion of that material as evidence, by means of the application of rules about admissibility, or with the consent of the parties.

(d) The evaluation of that material, by means of the process of allocation of weight.

(e) The making of a decision by the fact finder.

(f) The expression of that decision, by means of a verdict (in the case of a jury) or a verdict and reasoned judgment (in the case of a judge).

Steps (a) – (c) are well defined in the sense that they are undertaken within a formal framework in which relevant controlling rules are articulated with relative clarity. Step (d) is less well defined in the sense that much about the process of allocating weight is not governed by rules that are explicitly stated. Yet it is not only the first four steps which are important in the context of the goal of rational truth identification. Also important is step (e), which involves the fact finder moving from the body of evidence to a decision about the facts in dispute.

This decision making step is important to the objective of rational truth identification because it is central to the quality of the outcome of the fact finding process. It is the mechanism within which the ideal of rationality may finally be secured. The antecedent steps in the fact finding process may give rise to the *potential* for the identification of truth, but it is only at the decision making stage that such potential will be realised, if it is to be realised at all. A natural incident of the goal of rational truth identification will therefore be a concern to regulate the process of decision making to ensure that it functions in a manner that is optimal with respect to that goal. This concern gives rise to an important question: is this decision making step amenable to regulation of any kind?

This is a question that has given rise to considerable interest and debate. It has engendered, among other things, a search for "new models of proof and new concepts

of evidence to facilitate them".[138] Underlying this search is undoubtedly a concern about the absence of a definition of the allied processes of proof and decision making. Wigmore thought that what was missing, and what was therefore required, was a 'science of proof'.

> [T]here is, and there *must* be, a probative science—the principles of proof—independent of the artificial rules of procedure; hence, it can be and should be studied... The procedural rules for Admissibility are merely a preliminary aid to the main activity, viz., the persuasion of the tribunal's mind to a correct conclusion by safe materials... What is wanted is simple enough in purpose, namely, some method which will enable us to lift into consciousness and to state in words the reasons why a total mass of evidence does or should persuade us to a given conclusion.[139]

The framework within which attempts have been made to define more fully and to prescribe a process of decision making about facts is often referred to as the 'new evidence scholarship'. This term describes an interdisciplinary treatment of evidence law that emphasises how facts are proved and how "inferences should be drawn from a mass of evidence".[140] It favours this emphasis over concerns that are focused only upon the rules of evidence.[141] It is concerned with identifying and defining relationships between evidence and the issues that are to be resolved. It is also concerned to promote the use of these relationships in the legal fact finding process. New evidence scholarship is rightly regarded as 'new' not only because it is of recent origin,[142] but also because orthodox evidence law says virtually nothing about these relationships.[143]

---

138 Jackson, Note 119 at 328.

139 Wigmore J H, *The Science of Judicial Proof* (Boston: Little Brown & Co., 3rd ed, 1937) §§1-2.

140 See Jackson, Note 119 at 309, referring generally to Twining W L and Stein A, 'Evidence and Proof' in *The International Library of Essays in Law and Legal Theory* (Aldershot: Dartmouth, 1992) and attributing the phrase to Lempert R, "The New Evidence Scholarship: Analyzing the Process of Proof" (1986) 66 *Boston University Law Review* 439-477. See also Twining W L, "Rethinking Evidence" in Twining, Note 1, 341-372 at 349-350.

141 Jackson, Note 119 at 328.

142 This is a relative assertion. The work of John Henry Wigmore that is referred to here was first published in 1913: Wigmore J H, *The Principles of Judicial Proof as Given by Logic, Psychology, and General Experience and Illustrated in Judicial Trials* (Boston: Little Brown & Co., 1st ed, 1913). The work was renamed *The Science of Judicial Proof* for the publication of its third edition, see Wigmore, Note 139.

143 As has been discussed previously, there is a single—and in this context somewhat unenlightening—rule of relevance that merely seeks to identify a broad capacity for material to assist in the resolution of factual issues, and there are virtually no rules about the weight or quality of evidence.

A significant part of new evidence scholarship is concerned with the characterisation of processes of proof and decision making about facts in mathematical terms. This involves the use of mathematics to characterise the strength of relationships between discrete items of evidence and particular hypotheses about ultimate factual issues.[144] Yet the attempt to introduce mathematical concepts to legal fact finding has not been well received. This is best illustrated by debates about, and judicial reaction to, the use in decision making of the set of mathematical relationships that are described by Bayes Theorem.

Bayes theorem describes the way in which a particular item of evidence can modify or update knowledge, or belief, about the existing probability of the correctness of an hypothesis about a given fact. The existing probability is generally called 'the prior probability'. In the forensic setting, the hypothesis would be aligned with some ultimate fact that is to be proved. For example, an hypothesis to be evaluated may be that an accused was the killer of a particular individual.

At the core of Bayes theorem is a measure called the 'likelihood ratio'. This ratio is the probability that a particular item of evidence (blood stains on the clothing of an accused, for instance) would exist if an hypothesis were true, divided by the probability that the same item of evidence would exist if that same hypothesis were false.[145] Bayes theorem holds that the updated probability, which is generally called 'the posterior probability', of the existence of a given item of evidence if the hypothesis is true is the product of the prior probability and the likelihood ratio for that evidence.[146] The theorem can be used successively in respect of a series of items of evidence. Such use provides a means by which discrete and otherwise unrelated items of evidence can be linked together to determine the cumulative effect that they have upon a particular hypothesis.

---

144 For another description see Jackson, Note 119 at 311.

145 Kaye D H "What is Bayesianism?" in Tiller P and Green E D (eds) *Probability and Inference in the Law of Evidence : The Uses and Limits of Bayesianism* (Boston: Kluwer Academic Publishers, 1988) 1-19 at 9.

146 Note 145. For another description see: Dawd P "Probability and Proof", Appendix to Anderson T and Twining W, *Analysis of Evidence: How to Do Things with Facts, Based on Wigmore's Science of Judicial Proof* (Boston: Little Brown & Co, 1991) 385-441 at 421-422.

Mathematical devices such as Bayes theorem have a significance in the present context because they are said to have direct application to the decision making process. They provide a means of associating discrete items of evidence that is itself referable to the ultimate issue about which a decision is to be made. There is, however, a fundamental division of opinion over whether Bayes theorem and other mathematical expressions of probability generally are or should be applicable to legal fact finding.[147] The principal point of contention is whether mathematical expressions can accurately produce the connections between items of evidence and factual issues that have to be made when fact finding is undertaken for legal, and therefore essentially social, purposes.[148]

What is also important in the present context is that judicial opinion is substantially opposed to the use of devices such as Bayes theorem.[149] Comments by the English Court of Appeal in *R v Adams* [1996] 2 Cr App R 467 capture the prevailing judicial attitude. In that decision, Bayes theorem was said to

> [trespass] on an area peculiarly and exclusively within the province of the jury, namely the way in which they evaluate the relationship between one piece of evidence and another.[150]

A further complaint, recently expressed by Justice Gray of the South Australian Court of Appeal in *R v Karger* (2002) 83 SASR 135, is that Bayes theorem

> can only operate by giving a numerical percentage to each separate piece of evidence ... The percentage chosen is a matter of judgment. The apparently objective numerical figures used in the theory may conceal the element of

---

147 Twining, Note 140 at 350. Debates on this point are said to have had their genesis in the attempt to use statistical expressions of probability in *People v Collins* 68 Cal.2d 319, 438 P.2d 33 (1968), although an earlier decision, *Smith v Rapid Transit, Inc.* 317 Mass. 469, 58 N.E.2d 754 (1945) has also received attention in this context. The most significant opposition to the use of mathematical relationships to define and to regulate decision making appears in Tribe L H, "Trial by Mathematics: Precision and Ritual in the Legal Process" (1971) 84 *Harvard Law Review* 1329-1393. Other important contributions include Cohen J, *The Probable and the Provable* (Oxford: Clarendon, 1977). Cohen J, "The Logic of Proof" [1980] *Criminal Law Review* 91-103; Williams G "The Mathematics of Proof" [1979] *Criminal Law Review* 297-308 and 340-354. Eggleston R, "Beyond Reasonable Doubt" (1977) 4 *Monash Law Review* 1-22; "Probabilities and Proof" (1963) 4 *Melbourne University Law Review* 180-211.

148 See generally Jackson, Note 119 at 316; Shaviro D, "Statistical-Probability Evidence And The Appearance Of Justice" (1989) 103 *Harvard Law Review* 530-554 at 531.

149 For a discussion see Heydon, Note 79 at paragraphs [9090] – [9095].

150 [1996] 2 Cr App R 467, 481.

judgment on which it entirely depends. The use of Bayes' Theorem has been firmly rejected by the courts.[151]

It is noteworthy that the use of mathematical expressions about probability seems only to be objectionable when it is used between items of evidence. It is an objection at the 'inter-item' level. The expression of the significance of a single item of evidence in mathematical terms is a different matter. This is the case even though that expression involves the linking together of several 'intra-item' elements. The presentation of DNA evidence, for example, relies very heavily upon mathematical expressions of probability to explain the combined significance of matches across two or more samples of molecular patterns at not one, but several, distinct loci on the DNA molecule.[152]

This involves the use of arithmetic manipulation[153] of the expected frequencies with which particular alleles are observed within sample segments of a population. That manipulation produces a single statistical expression of the effect of simultaneous matches at multiple loci. It involves 'linking' the significance of the individual matches together via a 'product rule' and it is what generates the very large expressions of probability[154] that have come to be associated with DNA evidence.

Although this linking is not undertaken via the Bayes theorem, the mathematical processes are very similar[155] and certainly do not draw upon any non-mathematical considerations. In *Adams*, the prosecution used the product rule to characterise DNA evidence that encompassed the combined statistical significance of matches for nine

---

151 (2002) 83 SASR 135, 181. See also *R v GK* (2001) 83 NSWLR 317, 322-323 per Mason P. For an extra-judicial comment by Justice Hodgson of the New South Wales Court of Appeal which appears to favour the use of Bayes Theorem, see Hodgson J, "A Lawyer Looks at Bayes' Theorem" (2002) 76 *Australian Law Journal* 109-118, 118.

152 For an account of DNA analytical techniques and terminology see chapter one.

153 This is done by the so called 'product rule', which multiplies the individual probabilities of random matches for individual alleles: Inman K and Rudin N, *An Introduction to Forensic DNA Analysis* (New York: CRC Press, 1997) 93. Debate about the validity of the use of the product rule was a central feature of the 'DNA wars', as to which see the references cited at chapter one.

154 In *R v Adams* it was said that the DNA test result indicated a 1 in 297,000,000 chance of a random match between the crime scene material and an individual *other than* the accused: [1996] 2 Cr App R 467, 469.

155 In *R v GK* (2001) 83 NSWLR 317, 323 Mason P described Bayes Theorem in terms that drew no material distinction between it and the product rule. His Honour said that "[t]he [Bayes] Theorem underpins the statistics professed by the experts based upon what was described as the product rule."

loci,[156] and did this without comment. It was the attempt by the defence to use Bayes theorem to link four discrete and heterogeneous items of evidence[157] that drew judicial criticism.

It is of course arguable that the presentation of DNA evidence really involves the offer of a discrete item of evidence which the fact finder can really only deal with as a singular source of information. It should, however, be recalled that the empirical foundation for DNA evidence is the observation of discrete 'matches' in some *number* of loci. What is being presented to the fact finder is not an account of a single observation, but rather information about a series of discrete observation for a given sample of DNA material. In cases in which the sample that is tested contains the DNA of more than one individual, the report of a 'match' at any given locus may refer in fact to DNA material which is discrete from the DNA material for which matches at other loci may be reported.

The apparent paradox in judicial response to the use of mathematical techniques for 'inter' and 'intra' item assessment does not, however, need to be resolved for present purposes. It is sufficient to note that the mere attempt to invite a fact finder to adopt a mathematically founded decision making process for the entire body of evidence invites the reaction that has been described here. It seems that to go further and prescribe some formalised decision making methodology within the context of a regime of evidentiary treatment for particular material would encounter similar or more vehement opposition.

It is, as a result, not practicable to formulate a regime for the evidentiary treatment of given material that prescribes a particular decision making methodology. Equally, it is not possible to assume for the purposes of the operation of any such regime that the process of decision making that a fact finder employs will conform to any particular methodology. This restriction has the important consequence that the capacity of

---

For a similar conclusion, see Atchison B, "DNA Statistics May be Misleading" (2003) 41 *New South Wales Law Society Journal* 68-70.

156 Referred to in the judgment as "nine bands of DNA": [1996] 2 Cr App R 467, 468.

157 The items of evidence were the failure by the victim to identify the accused at an identification parade, the accused's own evidence as to non-commission of the offence, alibi evidence by the accused's girlfriend and evidence about the accent of the perpetrator: [1996] 2 Cr App R 467, 468.

material to aid rational truth identification must be assessed in terms of the inherent characteristics of the material, not in terms of a specific decision making methodology that might be employed to assess that material.

This does not mean that a poorly defined process of decision making is necessarily less likely to be able to identify truth. All that is suggested is that the absence of a formal definition of the decision making process renders it impossible to make assumptions about how material will be dealt with. In particular it cannot be assumed that the process of decision making will exhibit a supervening capacity to identify and to disregard information that is inaccurate. Equally, it cannot be assumed that the process will have no such capacity. The position that obtains is instead one of uncertainty about this capacity.

These considerations might have less significance if the goals of legal fact finding did not include the identification of truth by rational means. Yet truth identification is a principal goal of legal fact finding. For this reason, careful attention must be given to the kinds of material that are appropriate for use by the fact finder. This is necessary to compensate for the uncertainty that is imposed by a process of decision making that is not well defined. The ultimate consequence of this is to highlight further the important role that the application of rules about admissibility has to play in legal fact finding and, therefore, in any regime of evidentiary treatment.

## 2.5 Conclusions

This chapter has examined aspects of the primary goal of legal fact finding: the identification of truth by rational means. The 'truth' which this goal seeks is empirical. It is an account of observable events and transaction in a world that is predominantly physical, but in which social constructs can be the subjects and objects of such events and transactions. The rationality to which the goal aspires is a process of logical reasoning that is primarily inductive. It involves the development of inferences that are based upon sources of information. Those sources of information must be ones that have a capacity to indicate truth. This requisite capacity may be

expressed in terms of a likelihood that information in question is accurate with respect to the matters with which it deals. The strength of the likelihood of accuracy that is required is not, however, absolute. Were it so, the use of virtually all sources of information would fall outside what would be considered 'rational' in the relevant sense.

Something less than a guarantee of accuracy is required, but it is clear that the goal of rational truth identification requires more than mere ambivalence about the accuracy (or inaccuracy) of the information that is used in legal fact finding. It is possible that, for some kinds of material, certain assumptions about the likelihood of accuracy will need to be adopted. In these cases, there will be no way of quantifying precisely the risk that particular material contains information that is inaccurate. What the requirements of rationality impose in such circumstances is the need for a logical basis for making the assumptions that have to be made about the material in question. The standard against which the requisite basis is judged may not, however, be an exacting one. In this sense what is sought is rather more a 'coarse' rationality than a 'fine' one. As Wigmore argues

> [a]gain, wherever a rule or principle may be adopted in the effort to employ the recognized tests of reasoning, no attempt can be made to furnish ideal tests. Details, refinements, contingencies, and exact distinctions, which the ideal principle will demand, may be and must often be neglected so that the test may be serviceable. Perhaps it may be necessary to take a mean of convenience and lay down a specific and unshifting rule that will sometimes operate arbitrarily or unequally.[158]

This chapter has considered the principal mechanism that is available for the realisation of the goal of rational truth identification: the application of rules about admissibility. The application of rules about admissibility is required to provide a basic, but not comprehensive or absolute, assurance that information that is to be used by the fact finder is likely to be accurate.

The extent to which rules about admissibility should restrict or condition the use of given material is—and must be—determined by the nature of the information that the

---

158 Wigmore, Note 5 at §27.

material contains.[159] In some cases, few if any restrictions will be appropriate. In other cases a strict regime of admissibility will be indicated.[160] In all cases, the objective that should be sought is to apply rules that reflect rational beliefs about the nature of the information in question and the likelihood it will be accurate in any given case. The application of rules about admissibility will, as a discrete process, always have a role when the concern is to identify truth by rational means. Although the rules that are applied can and should vary from case to case, it is incorrect to suppose that the process of applying rules about admissibility can be abandoned in particular instances. It is also, as Thayer observed, incorrect to suppose that "general admissibility of what is logically probative is ... a necessary presupposition in a rational system of evidence."[161]

If the process of allocation of weight is to have an effective role to play in the rational identification of truth, then a necessary assumption is that some process of quality control will be applied to the material that is admitted into the relevant evidence pool. That process is the application of rules about admissibility. As has been stated, those rules need not apply an exacting standard. At the same time, they should operate with more than mere ambivalence about the likelihood that given information will be accurate or inaccurate.

This chapter has considered the constraints that apply to the expression of regimes of evidentiary treatment. These constraints appear largely to be the product of the 'culture' that is associated with legal fact finding in common law jurisdictions. They are not practical or logistical constraints. They simply are limits upon what might be considered 'acceptable' and to this extent they are material concerns for the formulation of new regimes of evidentiary treatment. They may not be optimal to the facilitation of rational truth identification, but they exist and they are well established features of the fact finding landscape. To this extent, they have to be recognised and

---

159 What is also relevant in this context is the extent to which an opposing party can test the evidence. Direct oral testimony is highly susceptible to critical scrutiny, because it is amenable to cross examination. This reduces the role that rules about admissibility need to play. Consistently with this view, direct oral testimony is widely admissible. Factors such as potential or actual bias or weakness in faculties of observation or recall do not render testimony inadmissible, but may be explored within cross examination.

160 Such as when the material consists largely of opinion that is based upon scientific or technical principles that are not well established.

considered. A regime for evidentiary treatment could not, for instance, seek to introduce mechanisms that subvert, or are substantially incompatible with, the adversary system.

Similarly, a regime of evidentiary treatment must take account of the reality that common law tradition resists attempts to regulate or even to define the process of decision making about facts. It would not be open to a regime of evidentiary treatment to seek to impose upon the fact finder a formal methodology for the making of decisions about facts, or more precisely about whether a particular onus of proof had been discharged. Perhaps more importantly, it would also not be open to a regime of evidentiary treatment to make assumptions about the way in which material that may be admitted under it might be used by the fact finder in the decision making process.

When the subject of interest is a particular kind of material,[162] the consideration of greatest importance will be the likelihood that the information that it contains will be accurate. Whilst the range of factors that have been considered in this chapter will contribute to shape an 'optimal' regime for the evidentiary treatment of that material, the starting point must be this fundamental issue.

In the case of computer-produced material, it is clear that the process of production will be important in this regard. This follows from the information transformation model that was introduced in chapter one. For this reason, an examination of the creation and characteristics of accuracy of computer-produced material is necessary for the purposes of this thesis. That examination is undertaken in the following two chapters.

---

161 Thayer, Note 61 at 144.

162 Such as material that has been produced by a computer.

# 3. Computer elements and operation

## 3.1 Introduction

Chapter two demonstrated that there is a fundamental connection between rationality in the sense applicable to legal fact finding and a *physical* reality. The next two chapters undertake an examination of the characteristics of accuracy of computer output in a physical context. This chapter considers the operation of the computer. Its foci are the elements of the computer that were identified in chapter one (hardware, software) and the physical aspects of their operation. The relative significance of the roles of these elements is also explored.

A question that is addressed in this chapter is whether a computer can be considered to be a member of a category of comparable and largely interchangeable devices that meet the basic dictionary definition that was given in chapter one,[1] or whether it is instead necessary to treat a given combination of hardware and software as unique in some material respect. More specifically, this chapter examines whether computer-produced material may itself be treated as an homogeneous class of material, or whether it must be considered and dealt with on a discrete basis.

---

[1] Namely "an electronic device, usually digital, for storing and processing data (usually in binary form), according to instructions given to it in a variable program": Moore B (ed), *The Australian Oxford Dictionary* (Melbourne: Oxford University Press, 1999) 275.

This chapter is ultimately concerned to establish the foundations for examining the reliability of computers in the sense in which that concept was introduced in chapter one. The concept of reliability bears directly upon the accuracy of the information that is contained in computer-produced material and, in turn, upon the object of rational truth identification. There is a need to assess the reliability of computers in general (or on a case by case basis) because of the connection between accuracy and reliability. At the same time, the discussion in chapter two makes it clear that any such assessment must be rational in its own right.

The question of reliability is considered in chapter four, following the examination in this chapter of the principles that underlie the operation of the elements of a computer. This chapter commences by reiterating the roles of those elements and highlighting the distinction between them. It then examines the role and function of each element. In doing this, it identifies the matters that need to be understood in order to consider how well—and therefore how reliably—those elements might function.

## 3.2 The elements of a computer

*The extent of the examination that is required*

A discussion of the subject that is dealt with in this chapter might be undertaken at any one of a number of different levels of detail. This thesis goes considerably beyond a cursory treatment of the functioning of computers. The decision to do this permits a more detailed statement to be given of the underlying principles upon which subsequent analysis and argument will depend. The object in this regard is to avoid the need to rely upon assumptions that may or may not be correct, or for which no basis or justification is given. Such reliance would not present an adequate foundation for decision making about regimes of evidentiary treatment and, ultimately, could not meet the requirements of rationality. It would instead commit the thesis to an examination of the accuracy of computer output without a sound basis for undertaking that examination. Apart from the possibility that this could lead to

conclusions about computer output that did not have a rational foundation, it might also increase the possibility that incorrect assumptions might be made.

The significance of the course that has been foreshadowed is that it is distinctly at odds with the approaches that are taken in much of the prior (legal) literature. Colin Tapper's approach in *Computer Law* illustrates the orthodox position, which is to avoid any detailed treatment of the physical aspects of the operation of computers. Tapper argued that this was justifiable on the basis that his work was

> concerned with computers as we all understand them. No attempt is made to define computers, nor does the book contain any account of their operation. While this was, perhaps a bold step to take in 1978, by 1988 it could more readily be justified on account of the vastly increased penetration of computers into everyday life.[2]

This distinction between the thorough exploration of the principles underlying the operation of computers, and the preparedness to overlook that subject or to deal with it on only a superficial level is an important one. As is demonstrated subsequently in the thesis, the approach taken here and the level of detail that is included are necessary if a rational foundation for dealing with computer-produced material is to be established. Conversely, a failure adequately to deal with the details and complexities that are inherent in the operation of computers diminishes the likelihood that any subsequent examination of the question of evidentiary treatment will be based upon an adequate foundation.

*Hardware and software are distinct elements*

Chapter one introduced some of the concepts that are foundational to computing. Drawing upon a standard dictionary definition, it characterised the computer as a device that deals with information and also as a device that operates according to variable instructions that are provided to it. The latter characteristic establishes a

---

2 Tapper C, *Computer Law* (London; New York: Longman, 4th ed, 1989) xliii.

fundamental distinction between the tangible device (the 'hardware')[3] and the intangible instructions and the information with which they deal (the 'software').[4]

The *IEEE Standard Dictionary of Electrical and Electronic Terms* maintains this distinction. It defines hardware as the "physical element used to process, store or transmit computer programs or data"[5] and software as the "computer programs, procedures and possibly associated documentation and data pertaining to the operation of a computer system".[6]

This distinction between hardware and software introduces the possibility that the way in which these elements affect the characteristics of output will itself be distinct. In particular, it may be the case that problems with the operation of one element may affect the accuracy of output, even though the other element operates flawlessly.

## 3.3 Hardware

Irrespective of the particular software that is used in a given case, it is the hardware that must, in the end, provide the functionality that the computer exposes to the real world. The 'real world' is the environment in which the computer is operated. This environment is the source of the information that the computer is to process and it is where use is made of any material that the computer produces. The real world contains the individuals and organisations that make use of the computer as a tool or aid in the attainment of some purpose(s) of interest.[7] These purposes could relate to commercial activity, research or even recreational pursuits. They may include legal fact finding, which is the case that is of present interest.

---

3 See for example Lucas H C, *Introduction to Computers and Information Systems* (New York: Macmillan: 1986) 79 and Lee G, *From Hardware to Software* (London: Macmillan, 1982) 210.

4 Note 3.

5 Institute of Electrical and Electronics Engineers, *The IEEE Standard Dictionary of Electrical and Electronic Terms* (New York: Institute of Electrical and Electronics Engineers, 6th ed, 1996) 475.

6 Institute of Electrical and Electronics Engineers, Note 5 at 1006.

7 The generic term 'user' is adopted here to describe an individual who makes use of the computer at a given time.

Hardware is traditionally arranged according to a 'von Neumann architecture'. It is named for John von Neumann (1903-1957) who is credited with the conception of the basis for modern computer design.[8] Under this architecture, the hardware exists as a number of discrete, connected elements. The elements that facilitate the reception of instructions or information (termed 'input') and the expression of results (termed 'output') are separate from the elements that are involved in the actual processing of information.[9] Of the latter elements, two are most directly responsible for providing the characteristic functionality of the computer, which is the ability to accept and execute variable instructions. Those elements are the central processing unit and the memory.[10] A typical von Neumann architecture is depicted in figure 3-1.

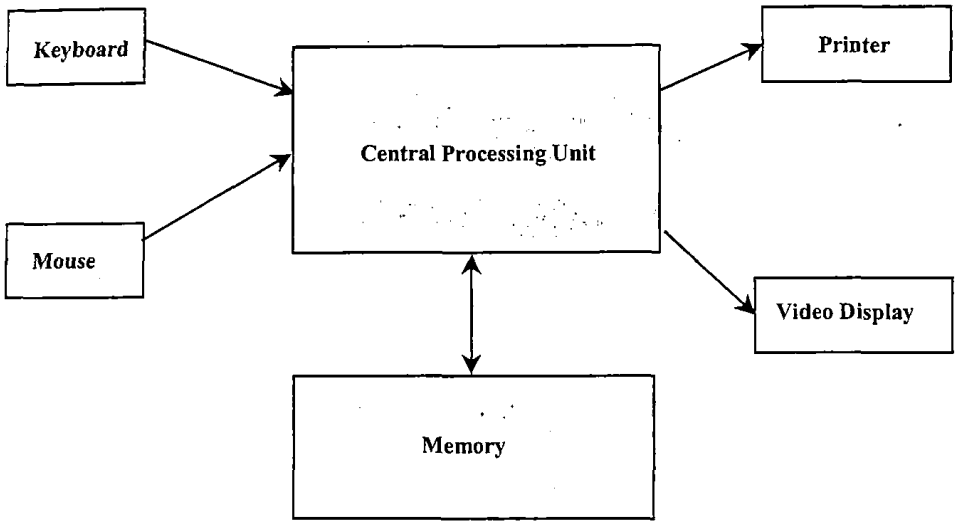



**Figure 3-1: A typical von Neumann architecture**[11]

The central processing unit is the element that is directly responsible for taking information and instructions,[12] processing them and transferring the results to the

---

[8] Baer J L, *Computer Systems Architecture* (Rockville, Maryland: Computer Science Press, 1980) 77; Lucas, Note 3 at 53; Lee, Note 3 at 1.

[9] See for example the accounts of the von Neumann architecture that are given in the references cited at Note 8.

[10] Clements A, *The Principles of Computer Hardware* (Oxford: Oxford University Press, 3rd ed, 2000) 212-214; Biermann A W, *Great Ideas in Computer Science: A Gentle Introduction* (Cambridge, Massachusetts: MIT Press, 2nd ed, 1997) 323.

[11] Diagram adapted from Clements, Note 10 at 13.

[12] Typically, instructions are not carried out in isolation but involve operation upon some pieces of information: Biermann, Note 10 at 259. An example is the comparison of two values to determine which is the greater. The comparison operation is distinct from the values being compared.

other elements.[13] It provides the essential functionality that is characteristic of computer hardware. The memory supports the operation of the central processing unit by facilitating the *persistent* storage of information. Once information is stored in memory, it remains there until it is specifically altered.[14] Memory holds the instructions and information that are to be processed. It also holds the results of the various operations, including any intermediate results that might be produced by a series of related processing operations.[15] Although this is a critical function, it is ancillary to the operation of the computer, since a memory device has no computational or processing abilities.[16]

Four aspects of the operation of the central processing unit are fundamental determinants of the way in which the hardware operates. The first is the reliance upon electric quantities and circuits to represent, process and transfer information in binary digital form. The second is the fact that the instructions that the central processing unit can accept must have a logical foundation. The third is the fact that the central processing unit can accept only a finite, predetermined range of instructions. The fourth is the manner in which the central processing unit realises a decision making ability, which is through the selective or conditional execution of instructions.

### 3.3.1 The use of electrical quantities to represent information

*An internal representation*

In order that a computer can be useful in most real world contexts—and certainly in legal fact finding—it has to provide information that is comprehensible to humans. This requirement applies, however, only to the information that is contained within the material that the computer produces. No such constraint applies to the

---

13 Clements, Note 10 at 212.

14 Provided that the operating conditions that are required by the memory hardware are maintained. Principal amongst these is electrical power; most memory devices cannot retain information after the hardware is switched off, because they use the state of electrical quantities within an electric circuit to record information. See Clements, Note 10 at 462.

15 Clements, Note 10 at 461.

16 Biermann, Note 10 at 259.

representation of information within the central processing unit, or within the other elements of the hardware.

These elements can use internal forms for the representation of information that are not necessarily comprehensible by a human. It is only the output devices (printers, video displays) that have to abide by the restriction that ties them to a form of representation that can be understood in the real world. In the result, these elements are really dedicated, as are the elements that accept input, to the conversion of information between internal and external forms of representation.[17]

Because the central processing unit is a wholly internal hardware element, it can deal with information that is expressed exclusively in an internal form. The internal form of representation that has proven to be most suited to computing is based upon the use of electric quantities that arise within electric circuits. This is the form of representation that is used in 'electronic' computers.[18] Electronic computers exploit the fact that it possible to vary and to measure electrical quantities within electric circuits at arbitrary points in time. This possibility therefore gives rise to a means by which information may be expressed and perceived. The particular electrical quantity that is used is electrical potential difference.[19]

The use of electrical quantities has advantages and drawbacks. The advantages include those which are commonly associated with modern computing: high speed, miniaturisation and low component costs.[20] The principal drawback is that the ability

---

17 See for example Biermann, Note 10 at 324.

18 The distinction between 'electronic' and 'electric' components and devices is a minor one for present purposes. Normally an electric circuit is opened and closed by a mechanical switch. The switches that turn ordinary household appliances 'on' and 'off' are one example. By contrast, electronic components can be used to open and close electric circuits without any need for mechanical switching. For a discussion see, for example, Bobrow L S, *Fundamentals of Electrical Engineering* (New York: Oxford University Press, 2nd ed, 1996) 768-769 and Biermann, Note 10 at 256-257.

19 Electrical potential difference is an expression of the electrical energy that is expended upon the movement of a quantity of electric charge. The unit of measurement of electrical potential difference is the volt, named for Alessandro Volta (1745-1827): Bobrow, Note 18 at 3-4. The existence of electrical potential difference between two points or regions is the property that causes electrical current to flow: Bobrow at 6. This can occur in a conventional electric circuit, when a conducting path exists between the points of electrical potential difference: Bobrow, Note 18 at 8, 346.

20 These advantages accrue especially because of manufacturing technologies that allow large numbers of circuits and circuit elements to be located on a single integrated circuit 'chip': see for example Bobrow, Note 18 at 890-981. Also important is the fact that electric current flows at a very high speed within the material from which these 'chips' are constructed. That speed is approximately

to vary and to measure electrical potential difference has limitations. It is not absolutely precise. In the context of miniature, high speed circuits it can be accomplished with confidence only when the possible values of this quantity[21] are limited to two discrete choices, rather than a continuous range of possible values.[22]

These choices are either a relatively high value that is within a range that is typically close to 5 volts, or a relatively low value that is within a range that is typically close to 0 volts.[23] Restricting the choice of representation to these two disparate ranges permits the expression of information in a way that maximises discrimination between the two states. Minor fluctuations of voltage within each of these ranges, which are phenomena that can often be encountered in electric circuits, do not lead to ambiguity or inaccuracy in the expression or perception of information.[24]

*The use of binary forms of expression*

When the choices for expression are limited to two values or states, information must be represented in a binary form.[25] The typical notations for the choices are '0' (corresponding with a relatively low voltage) and '1' (corresponding with a relatively high voltage). Using this notation, information can be expressed in a numerical, digital form. A single binary digit is called a 'bit', a sequence of eight bits is called a 'byte' and larger sequences have other names, although they tend not to follow a uniform convention.[26]

Binary representation confines choices for the expression of information to numbers.[27] Despite this, other information such as letters and other symbols can also

---

70 percent of the speed of light, or around 210,000,000 metres in one second: see Clements, Note 10 at 634.

21 Measured and expressed in volts: see Note 19.

22 See for example: Clements, Note 10 at 18.

23 Clements, Note 10 at 18-19, 633.

24 See for example: Clements, Note 10 at 19.

25 Bobrow, Note 18 at 749. In a binary system, the radix or 'base' of the number set is two and the digits are the elements in the set {0,1}. The universally familiar decimal system has a radix of 10 and digits in the set {0,1,2,3,4,5,6,7,8,9}. See for example: Bobrow, Note 18 at 751.

26 See for example: Clements, Note 10 at 151; Biermann, Note 10 at 255.

27 It does not entail, however, only a total of two choices for the representation of information. As is the case with a decimal system, digits may be combined in a series to represent a large number of states. For example, the value '10011001' would be a valid selection in a system accommodating up to

be represented. This kind of information can be represented in a derivative form in which the items that are to be represented are allocated an unique numerical code. One of the best-known codes for representing alphabetical characters numerically is the American Standard Code for Information Interchange, or 'ASCII'.[28] ASCII assigns a number to each character of the (English) alphabet and also to a range of printing and punctuation marks. Because the numbers increase with alphabetical sequence,[29] the code incorporates not only a means of expressing individual characters through the use of binary digits; it also imparts knowledge of alphabetical order.[30]

There are other, more complex schemes for representing images and sounds within a binary, numerical framework.[31] Less formal schemes are also possible. Things that seem to be subjective and inherently non-numerical might also be represented by such schemes. The only essential requirement for any such scheme is the assignment of a separate numerical value to each unique element that is to be represented.

The use of binary, electrical quantities for the internal representation of information by the central processing unit and other elements of the hardware adds some additional complexity to the process of expressing information. It does not, however, restrict the kind of information that can be represented. What is restricted is the way in which information can be *processed*.

---

eight digits. The position of each digit signifies the power of the base that the digit is multiplied by to yield the represented value. In binary, $10 = 1 \times 2^1$, (2 in decimal) $100 = 1 \times 2^2$ (4 in decimal) just as in the decimal system $10 = 1 \times 10^1$, $100 = 1 \times 10^2$ and so on. See for example Bobrow, Note 18 at 751, Clements, Note 10 at 154-155 and Biermann, Note 10 at 240-241.

28 See for example Clements, Note 10 at 606; Biermann, Note 10 at 255.

29 For example: the character 'A' is 65 (or 1000001 in binary: see Note 27), 'B' is 66, 'C' is 67 and so on.

30 This has direct applications to the sorting of text. ASCII goes further by assigning separate codes to upper and lower case letters. Whereas 'A' is 65, 'a' is 97. This has still further applications to text manipulation and it is the basis of a range of operations that are provided by word processing programs.

31 See Clements, Note 10 at 162. For a discussion of such schemes and their applications to matters such as image and speech recognition see Kurzwell R, "When Will HAL Understand What We are Saying? Computer Speech Recognition and Understanding" and Rosenfeld A, "Eyes for Computers: How HAL Could 'See'" both in Stork D G (ed), *HAL's Legacy: 2001's Computer as Dream and Reality* (Cambridge, Massachusetts: MIT Press, 1997) 131-169 and 211-235.

*Restrictions upon the processing of electrically represented information*

Electronic components within an electric circuit behave in a manner that alters electrical quantities in response to particular conditions.[32] This behaviour can be controlled so that it realises a particular purpose. This in turn provides a foundation for processing of information according to some predetermined scheme.

It is possible to treat this kind of information processing as involving some number of discrete 'operations'.[33] An operation applies particular criteria that prescribe a required information outcome (or 'output') for each possible initial (or 'input') state. For example, an inversion operation can be defined. It has the purpose of swapping or 'inverting' whatever input information is supplied. For binary digits, this operation will cause an input of '0' to yield an output of '1' and an input of '1' to yield an output of '0'.

Such an operation seems limited in usefulness because it operates on only one piece of input information. Other operations that operate upon multiple inputs can also be defined. For example, an operation might compare two values and yield an output of '1' if one (but not both) of the inputs is '1'. If both inputs are '1' or both inputs are '0', the output is '0'. In the language of computing, this operation is designated

---

[32] See generally Bobrow, Note 18 at 765-772; Clements, Note 10 at 19-25. Typically, many such transformations or alterations have to be performed in order to carry out even a rudimentary task. For this reason, the electrical potential difference that is used to represent information within the central processing unit is delivered to electric circuits within the central processing unit in 'bursts' or 'pulses' that establish and maintain the required voltage level for only so long as is needed for each element within the electric circuit to effect the transformation. For the most common elements, this period is small but finite. It is of the order of 5-25 nanoseconds (1 nanosecond is equal to one billionth of a second) for each 'logic gate' (see Note 35) in the circuit: Bobrow, Note 18 at 820; Clements, Note 10 at 634-635. To facilitate the delivery of electrical potential difference in pulses and to maintain synchronisation between large numbers of electric circuits, a single 'clock' is implemented within the central processing unit. It does no more than produce a voltage 'signal' that continuously alternates between relatively low and relatively high voltages (that is, binary 0 and binary 1): see for example Clements, Note 10 at 218. The 'speed' of the clock cycle, which is the number of times that the signal switches from 0 to 1 and back to 0 in one second, is typically over one billion times. It is this figure that is popularly associated with the speed and therefore the processing 'power' of a computer. Clock speeds for contemporary personal computers now typically exceed two gigahertz (or two billion cycles per second).

[33] The difference between 'instructions' and 'operations' is subtle. The terminology that is adopted here is to use the term 'instruction' to refer to those items that can be referred to in the software. An 'instruction' can be nominated directly in a program of instructions: see Clements, Note 10 at 213. As discussed in section 3.3.3, the execution of a single instruction invariably involves the execution of a number of operations.

'exclusive OR' and it is used very frequently in a range of information processing operations that computers perform.[34]

The operations that can be defined for a system that uses binary electrical quantities for the representation of information are limited by the fact that both the inputs and outputs have to be expressed as binary digits. The electronic components that actually alter electrical quantities (usually called 'gates')[35] can only accept electrical potential difference as inputs and they can only produce electrical potential difference as an output.[36] Operations that are realised by logic gates must have a basis in binary or 'Boolean' logic.[37]

### 3.3.2 Instructions that have a logical foundation

Boolean logic involves operations that combine binary input information with fixed rules about the relationship that is to apply between input and output.[38] They produce defined (binary) outcomes in a manner that is synonymous with the production of a deductive conclusion from stated premises. They share common features with deterministic[39] mathematical relationships. It has been said that Boolean operations are an "algebra of logic".[40] Even when basic Boolean operations are combined to produce more complex relationships,[41] those relationships will still retain this logical

---

34 See Bobrow, Note 18 at 769; Clements, Note 10 at 29-30.

35 Or 'logic gates': Bobrow, Note 18 at 769. Normally a gate is named for the logical operation that it realises. For example there are (among others) 'AND', 'NOT', 'OR' and 'exclusive OR' gates: Bobrow at 768-772,

36 See for example: Bobrow, Note 18 at 768-772, and for a detailed discussion of the way in which these alterations are carried out, at 443-450. See also Biermann, Note 10 at 224-230.

37 Named for George Boole (1815-1864).

38 The terms 'input' and 'output' are used here in a special sense. They are relative to the operation that is to be carried out. They refer to input and output information that is developed internally, not to information that passes between the hardware and the real world.

39 That is, relationships that determine or define outcomes within a system. For example, the rule that determines that the result of the addition operation '1 + 1' will always be '2'. They are distinguished from probabilistic relationships, which merely predict outcomes. Also distinguishable are random outcomes, which occur in a system in which there is no identifiable relationship between the elements.

40 Clements, Note 10 at 50.

41 Such as those that have direct applications to mathematical calculations. See for example Bobrow, Note 18 at 772; Biermann, Note 10 at 240-248; Clements, Note 10 at 50-58.

character. Boolean operations make up the constituent parts of every instruction[42] that a central processing unit can accept and execute.[43]

A typical 'instruction set'[44] for a central processing unit includes instructions for transferring information to and from memory, basic arithmetic operations, shifting operations[45] and comparisons.[46] All of these kinds of operations have a logical basis. They are quite distinct from, for example, the kinds of decision-making that are usually regarded as 'human' in nature. This distinction limits the extent to which a computer can be programmed to emulate human thought and human decision-making.[47]

This limitation is illustrated when an attempt is made to have a computer produce random numbers. Applications such as cryptography require large, unique random numbers that cannot be reproduced. Because the computer can use only deterministic operations to produce output, a series of numbers that a computer produces can often be predicted and replicated.[48] Computer hardware is tied to logical rather than arbitrary operations and it lacks the kind of functionality that is required to give rise to truly random output.

While other real world tasks do not call for complete arbitrariness, they do call for behaviours that model complex natural processes. Such modelling requires

---

42 These are the instructions that will appear in a given software program.

43 The ability of the hardware to execute those instructions is created by the combination of electrical circuits that implement fundamental Boolean operations. For a detailed discussion of the techniques that are employed, see Bobrow, Note 18 at 808-853.

44 It is customary to refer to the range of instructions that a particular central processing unit can deal with as its instruction set. See for example Clements, Note 10 at 245.

45 In which digits in a series are shifted so as to increase or decrease the order of magnitude of the number that is represented. For example, the instruction to shift the binary number '00001100' (decimal 12) one place left would result in the number '00011000' (decimal 24). The order of magnitude of increase is two, because the representation is binary, or 'base' two. Shifting left increases the magnitude, shifting right decreases it. A variant 'circular' shift operation will take any digits that are removed at one end of the sequence and insert them at the other. For a discussion of shifting operations see Clements, Note 10 at 276-278.

46 See for example Clements, Note 10 at 269-285; Biermann, Note 10 at 265.

47 For a discussion of the effects of this limitation on the performance of computers that have been programmed to play the game of chess see Campbell M S, "An Enjoyable Game: How HAL Plays Chess" in Stork, Note 31 at 75-98.

48 The problem is discussed in Schneier B, *Applied Cryptography: Protocols, Algorithms, and Source Code in C* (New York: John Wiley & Sons, 2nd ed, 1996) 44. Attempts to solve this problem frequently resort to an external source of random information that the hardware can use as an input. See Schneier at 423-425.

organisation of the logically founded instructions that the hardware can carry out in such a way that the resulting function is much more complex and less determinative than its constituent parts.[49] Achieving this can be very difficult. Frequently the limiting factor is the extent to which such processes can be broken down into operations that require only the simple logical manipulations that the hardware can perform.[50] This limitation contrasts with other aspects of operation of computers, which surpass natural human abilities. Those include the ability of computers to deal with large quantities of information[51] and to do so at great speed.[52]

### 3.3.3 A finite set of instructions

The central processing unit is a physical element that has a fixed design and construction. The set of instructions that it can execute is also fixed, and it is finite.[53] If a particular instruction is not included in the design of a central processing unit, then it is not possible to include that instruction in any program[54] of instructions that is to be executed by that central processing unit.

---

[49] A problem of long standing interest is the task of providing computers with linguistic ability: the capacity to understand and to use language for the meaning that it communicates. This is an aspect of attempts to create systems that exhibit so-called 'artificial intelligence'. The classic test of such intelligence relates to linguistic ability and was proposed by Alan Turing (1912-1954) in 1950. Under this test a machine is 'intelligent' if, in communicating with it at a distance (say via teletype), a person could not determine whether they were in fact communicating with a machine or another human, see Turing A, "Computing Machinery and Intelligence" (1950) 59 *Mind* 433-460. For a further discussion of the problem see Schank R C, "I'm sorry, Dave, I'm Afraid I Can't Do That: How Could HAL Use Language?" in Stork, Note 31 at 171-190. For a discussion of related problems that focus on the problems of imparting differing forms of human knowledge to a computer see Biermann, Note 10 at 455-462; Kurzwell, Note 31 at 131-169.

[50] For a discussion of how this might be done for the task of recognising images such as facial features and expressions see Rosenfeld, Note 31 at 211-235.

[51] Despite the fact that computers use binary digital representation, the combination of a series of only 32 binary digits in the manner described in Note 27 permits the representation of $2^{32}$ (or over 4 billion) unique values.

[52] See Note 32.

[53] This proposition does not ignore the existence of programmable logic elements that allow for configuration after they are manufactured, as to which see for example Clements, Note 10 at 86-92. What is suggested is that at the point at which software is to be executed by the hardware, there is a single configuration for which the set of instructions that are valid (and can therefore be executed) is unchangeable.

[54] It is common to refer to 'programs' of instructions. A program is a list of instructions that is to be executed by the hardware: see generally Biermann, Note 10 at 10; Clements, Note 10 at 1.

A single instruction, such as one that calls for the multiplication of two numbers, requires the performance of a number of logical operations. Multiplication is carried by the hardware in a manner that approximates the 'pencil and paper' approach that would be taken by a human.[55] The essential features of the task are the breaking down of the problem into a series of multiplications that involve each individual digit, and a final tallying of the resulting partial products.[56] Each individual multiplication operation involves the comparison of two individual binary digits. Each comparison produces, as its result, a single binary digit.[57]

The operations that make up a single instruction have to be implemented in an electric circuit in the manner described in 3.3.1. This circuit will contain the logic gate that is appropriate to the operation to be performed. There must also be a mechanism for transferring the information that is to be processed to these circuits. That mechanism has to ensure that the individual operations occur in the correct order and that the results of each operation, reflected in changes to the original information, are propagated correctly through the process.

The required mechanism is implemented by a distinct component within the central processing unit. It is called the 'control unit'.[58] Section 3.3.1 described the use of electronic components to facilitate switching actions within electric circuits without the use of moving parts.[59] The same approach is used within the control unit. The control unit makes use of logic gates together with appropriate sequences of electrical potential difference.[60] The result is that the control unit can direct the electrically

---

55 For a discussion of different approaches to binary multiplication using logical operations see Clements, Note 10 at 194-200.

56 An important part of the process is to ensure that the correct order of magnitude is allocated to each partial product (i.e. that they are located in the correct 'columns'). The 'shift' operation that was described at Note 45 can be applied to this end.

57 The four possible binary multiplication operations involving two single bits are: 0 x 0 = 0, 1 x 0 = 0, 0 x 1 = 0, 1 x 1 = 1. This is equivalent to the logical comparison operation that is known as 'AND'. It can be implemented by a single electrical circuit that comprises the appropriate electronic elements that make up the 'AND' logic gate: see Clements, Note 10 at 194.

58 Clements, Note 10 at 218.

59 See in particular Note 18.

60 See for example Clements, Note 10 at 218, 239-243. As discussed at Note 32 the sequences comprise short 'pulses'. They are a combination of the repeating clock pulses and a unique sequence for the particular instruction, yielding an unique pattern that is used to 'switch' the required information (which is also made up of pulses) to the required circuits.

represented information that is to be operated upon (normally called the 'operands')[61] to the circuits that are appropriate to the operations that relate to the chosen instruction.

Some design choices exist for the implementation of the control unit. A dedicated circuit can be provided to cater for the required instruction as a whole. This will typically be a large, complex circuit that contains a number of logic gates that are arranged in the sequence that follows the order that is required for the operations that are to be carried out. Under this design, there will be one such circuit for each instruction that the central processing unit supports.[62] Once the circuit that is appropriate to the instruction to be executed is selected, the electrical quantities that represent the operands are applied to that circuit by means of the switching functionality of the control unit. The outcome of this process is the production of a series of values of electrical potential difference by the circuit. Those values represent the result of the operation.[63] Usually, they will be transferred to memory pending further processing.[64]

An alternative design approach for the control unit involves a circuit that can sequentially choose the required number of smaller logical operations in the correct sequence. These smaller logical operations are combined to implement the actual

---

[61] The instructions that a central processing unit can understand typically operate upon one or two operands at a time. See for example the kinds of instructions that are discussed in Clements, Note 10 at 269-285.

[62] The circuit will typically accommodate the processing of an operand that is represented as a series of binary digits of some length. The length is called the 'word length' and varies between different hardware. A value of 32 is typical for modern computers: Clements, Note 10 at 151. This requires that the circuit have sufficient paths to process all of the bits of the word in 'parallel'. This is a necessity for operations that make up arithmetic instructions because these apply the usual arithmetic processes of 'borrowing and 'carrying' between digits. For a discussion in relation to the addition operation see Clements, Note 10 at 172-175.

[63] The values are presented in parallel and make up a word that represents the result of the operation in the manner described in Note 62.

[64] This point needs to be understood in the context of typical memory architecture. Memory has a number of forms. Among these are memory elements that are called 'registers'. Registers are special circuits within the central processing unit. They use a feedback mechanism that maintains particular voltage levels for more than one clock cycle. In this sense registers facilitate the persistent storage of information. Registers use parallel paths to hold simultaneously all of the digits that make up a given value. Information can readily be switched between the registers and the circuits that carry out the processing that each instruction requires. Often, the registers are used as an intermediate store for the results of operations. These results are combined in turn with the results of other operations. Particular registers also hold information about the next instruction that is to be executed and the point in the

instructions as they are required.[65] This approach is said to result in a "computer within a computer"[66] in which the instructions are carried out by way of "microinstructions".[67] In central processing units that implement this design, a specification by software that a particular instruction is to be carried out is satisfied by the control unit causing the execution of the appropriate sequence of microinstructions. Even under this approach, there is a clear boundary between the hardware and the software. The microinstructions are arguably part of the hardware rather than the software, because a person who creates a program of instructions cannot access them directly,[68] nor can they alter the order in which individual microinstructions are executed.

The fact that the instruction set is finite is important because an instruction set is the only interface between the hardware and the software. It specifies what the software (and, ultimately, the person who creates the software) can ask the hardware to do. Software can attempt to build more complex functions by combining these instructions, but in doing this it is always limited to using the set of instructions that the hardware can understand.[69]

### 3.3.4 Conditional execution of instructions

It is said that the power of computers lies in their ability to make decisions.[70] In this sense, 'decision-making' involves selective action that is based upon the satisfaction of some criteria at some point(s) during the execution of a program of instructions. Because the only action that the hardware can take is to accept and execute instructions, the actions that it takes in order to give effect to decisions must involve the execution of one or more instructions.

---

software program that the central processing unit has reached. For a further discussion see Clements, Note 10 at 266-269 and 464-465.

65 For a discussion of the alternatives, see Clements, Note 10 at 231.

66 Clements, Note 10 at 231.

67 Clements, Note 10 at 231.

68 Clements, Note 10 at 213.

69 See in this context: Parsons T W, *Introduction to Compiler Construction* (New York: Computer Science Press, 1992) 1.

As software comprises a group or list of instructions,[71] it seems reasonable to expect that hardware would ordinarily execute instructions one at a time in the order in which they are provided. Decision-making by the hardware really involves decision-making about which instruction will be executed next. This requires the ability to deviate from the order in which instructions appear in a particular program. Hardware has this ability because its design provides for the execution of instructions that can modify the sequence in which other instructions are executed.[72] These are known as 'conditional' instructions.

Conditional instructions operate by causing the central processing unit to test for a certain condition and, depending upon the results of that test, to move to some other portion of the program in lieu of simply executing the next instruction. As with the other instructions and operations that the central processing unit can carry out, the testing process is a logical one. It is limited to a comparison of values that are represented in binary digital form.[73] The decision-making capability of all hardware is therefore limited to criteria of logic, rather than instinct or intuition.

The use of conditional instructions allows certain parts of a program of instructions to be excluded from execution on some occasions, but not others. It also allows parts of a program to be executed repetitively. These abilities introduce considerable flexibility in the way in which a program of instructions is executed and, as a result, the kinds of tasks that a program can achieve. When decision-making is tied to the state of initial input data, a program becomes very versatile. It can be used to carry out a given task in a variety of situations.

This versatility is also a drawback because the use of only a small number of conditional instructions greatly increases the number of unique sequences in which instructions might be executed. This is especially the case when conditional statements are 'nested' within other conditional statements. Then, the evaluation of one condition leads not to a set of statements that are to be executed sequentially and

---

70 Clements, Note 10 at 220-221.
71 See Note 3.
72 Clements, Note 10 at 221.

unconditionally but rather to further conditional statements. These present still further alternative paths for the execution of the program to take. A program containing just 25 sequentially nested conditional statements each presenting two paths for further execution will create $2^{25}$ (in excess of 33 million) unique sequences of possible program execution.[74] Under such conditions, predicting how the program will behave and, consequently, what will be produced when it is executed becomes very difficult.

### 3.3.5 Ancillary functionality

The characteristics of hardware that have been discussed are those that are most directly connected with the characteristics that are unique to the computer. They have centred naturally upon the fundamental aspects of hardware, being those which most directly influence the manner in which information is represented and processed by the computer.

The components that are directly involved in the realisation of these characteristics are, however, only a part of the wider hardware system. A variety of supporting mechanisms and components also contribute to the operation of the hardware. They have responsibility for operations such as the timing and sequencing of operations, the transmission of information between the various hardware elements, the arbitration of requests by the various hardware components for interaction with the central processing unit and for the temporary and long term storage of information, to name just a few.[75]

---

[73] These values may have been supplied as input information from the 'outside world' via an input device. or they may have been created and/or modified at an earlier stage of program execution.

[74] This result is reported in McCabe T J, "A Complexity Measure" (1976) 2 *IEEE Transactions on Software Engineering* 308-320, 308.

[75] For a discussion of these and related topics see Clements, Note 10 at 82-85, 220-223, 231-245 and 466-495.

## 3.4 Software

### 3.4.1 The role of software

*Control of hardware*

In the 'real world', computers are recognised for their ability to perform a variety of functions.[76] This ability represents the realisation of the potential for the physical device—the hardware—to accept and to act upon variable instructions. The kinds of instructions that the hardware can accept are, as was discussed in section 3.3.3, limited to those for which its design has provided. What lies behind the potential of the hardware to be a multi-purpose device is the fact that even a limited number of instructions can be combined in a very wide variety of ways. Different combinations of instructions can implement different functions. This, coupled with the fact that different operands[77] can be used with each instruction, is what gives rise to the variable functionality of the computer.

While it is the hardware that must provide the functionality that the computer exposes to the real world, it is the software that controls the hardware. The finite range of instructions that the hardware can recognise must always limit what the software can direct the hardware to do, but the hardware does not determine what ultimate functionality the computer will exhibit. This critical role is reserved to the software, making it a very important element.

*Realisation of a 'real world' purpose*

When a given program is executed, the tangible result is control of the operation of the hardware. A specific program of instructions gives rise to a specific hardware behaviour. In general, this behaviour represents the processing of information. When

---

[76] See for example Brown R A, *Documentary Evidence in Australia* (Sydney: Law Book, 2nd ed, 1996) 355: "[a] computer is a universal machine, capable of performing any computable function."

[77] Frequently a program will obtain the variable information by making provision for the reception of that information from an input device. For example, a spreadsheet program performs calculations based upon the information that is provided to it via a keyboard and (usually) some form of pointing device such as a mouse.

the hardware includes peripheral devices that can accept and produce information in a human readable form, the behaviour encompasses everything that comprises a real world purpose. It covers the reception of initial input information and the production of output that can be comprehended (and used) in the real world.

In many contemporary real world contexts, the processing of some information is a useful end in itself. This is true even when the information processing is constrained to operations that have a logical foundation, as is the case with electronically based hardware. Aside from purely mathematical computations, hardware can deal with textual, pictorial and symbolic information. Even though the representation of this information is tied to numerical foundations hardware can, via suitable coding mechanisms,[78] perform a broad range of operations with it.

The role of software is to impose purpose upon the behaviour of hardware. It moulds from that behaviour a composite functionality that has both significance and utility in the real world. The concern that arises in this context is very frequently not whether a computer can realise a useful real world function, but how well it can do so. A criterion for assessing how well a computer operates in this real world context is whether it performs the task or function that a user expects it to perform. This consideration is patently critical in the context of the information transformation model that was introduced in chapter one. The balance of this chapter deals with how software is created and how it functions. The question of how *well* software (and hardware) may operate is explored in chapter four.

### 3.4.2 Software languages

#### 3.4.2.1 A linguistic analogue

Hardware imposes certain constraints upon how instructions can be included in a software program. There is, for instance, only a finite set of instructions that can be nominated in any program. Software that is to operate on given hardware has to

---

[78] Such as the ones discussed in section 3.3.1.

confine itself to expressions from within the instruction set for that hardware. This is a consideration that is comparable to the lexical constraints of communication in a 'natural' language, such as English.[79]

There are also syntactic limitations. Hardware has to be able to recognise and to distinguish between the different instructions, and between the instructions and the operands or information to which the instructions are to be applied. In addition, hardware has to be able to associate the appropriate operands with the particular instructions for which they are intended. For this reason, the form in which the program is expressed must follow a formal syntax. In natural languages, syntax is dictated by grammatical rules[80] and it is possible to carry the linguistic analogy further to say that software is subject to grammatical constraints as well.[81] A further constraint that also has a linguistic analogue involves meaning. In order that software can be effective, it has to express some intended 'meaning' or purpose. That meaning relates to the function that the execution of the software is expected to fulfil. This requirement is, by analogy, a semantic one.[82]

These requirements can be regarded as prerequisites for the effective communication to the hardware of the task or function that is desired by the person who uses the software. The requirements, and the analogy of communication, make it natural to refer to forms of expression for software as programming 'languages', and this nomenclature is common, if not universal, in computing.[83] Due to the constraints that were discussed in section 3.3.1, the only 'language' that the computer hardware can understand is one that consists entirely of elements that are expressed in binary digital form.[84] Such a language is commonly called 'machine language' or 'machine code'.[85]

---

79 See in this context Parsons, Note 69 at 64-68.

80 Parsons, Note 69 at 68.

81 See for example Parsons, Note 69 at 68 and for further details of work in this area at 70-82.

82 See Parsons, Note 69 at 69 and 197-168 and also Biermann, Note 10 at 280-283.

83 See for example Biermann, Note 10 at 10 and Clements, Note 10 at 213.

84 Parsons, Note 69 at 1. The variables also have to be presented as electrical quantities. This highlights the importance of input devices such as keyboards. These devices are essential to facilitate the conversion of instructions and information that are provided by humans to electrical quantities that the other hardware can use.

85 Biermann, Note 10 at 268; Clements, Note 10 at 213.

3.4.2.2 Machine and assembly level languages

Different instructions can be distinguished by their functional descriptions. For example, one function adds numbers, another divides them, and another compares some values and takes some action depending upon the result. Classified in this way, the instructions that might form part of the instruction set for given hardware are really elements that have non-numeric descriptions. As is the case with other such sets, the instruction set and its elements can be represented to and by the hardware with a numeric coding scheme.

The use of such a coding scheme in software is what allows a particular desired sequence of instructions to be expressed to the hardware. The codes for individual instructions are referred to as 'operation codes'.[86] Electrical representations of the operation codes are used by the control unit to select the electrical circuits that are needed for the performance of each instruction. Machine code therefore consists entirely of binary numbers. Both the instructions and the data[87] are expressed as numeric values. Consequently, producing software in machine code is difficult for a human programmer. This difficulty is addressed by the use of software that can itself recognise non-numeric expressions and translate them into numeric machine code.

At the simplest level, the non-numeric expressions are abbreviated labels that designate particular instructions on a 'one for one' basis. The instruction that adds two numbers together can, for example be refereed to by the label 'ADD' instead of its numeric operation code (which might be '010'). This form of expression is a

---

[86] Clements, Note 10 at 218. This illustrates the ambiguity between the terms 'instruction' and 'operation'. A single operation code refers to an instruction, but as is discussed at Note 33 each instruction would normally involve a number of logical operations.

[87] The information (operands) can be referred to in a program either directly, as an absolute value (for example the binary value '10001101'), or by reference to a location in memory at which the value is stored. The latter approach is much more common. It allows different values to be used at different times because what is referred to is the value that is *presently* stored in memory, not a fixed value. If the program contains suitable provisions, variable input information can be stored in memory such that a single program can operate with a variety of input information in the manner that was described in chapter one, section 1.1.2.1. Such a program is not limited merely to performing the same instructions with the same information every time it is executed. As is the case with the other hardware elements, memory uses electrical quantities. It is able to store specific information for an arbitrarily long period in specific and unique locations or 'addresses'. These addresses are used in the instructions that are provided to the central processing unit. The instructions themselves are stored in memory also. See generally Clements, Note 10 at 213-214, 468-470.

software language itself. It is known as 'assembly' language.[88] Although labels are used in this language to refer to instructions, most references to operands or to the memory locations in which they are stored still have to be in numeric form.[89]

3.4.2.3 Higher-level languages

*Benefits of higher-level languages*

Because the relationship between machine code and assembly language is still one of basic substitution of labels for numbers, the availability of an assembly language does not substantially reduce the programming difficulties that are involved. Any task that is to be implemented in an assembly language still has to be broken down into a sequential list of 'statements'.[90] These can be no more complex than the very simple instructions that are implemented in the target hardware.[91]

Efforts to simplify the task of writing software have led to the creation of so called 'higher-level' languages.[92] As with assembly language, these languages are again implemented by means of translating software. This software takes the higher-level language program that the author of the software creates and produces a machine code expression of it that can be understood by the hardware. Higher-level languages can typically convert single expressions into an arbitrary number of instructions at the assembly language level, and then into machine code. There is, therefore, a departure from the 'one to one' relationship that exists between assembly language and machine code. This represents a considerable reduction in the effort that is involved in the creation of software.

---

88 Parsons, Note 69 at 2; Clements, Note 10 at 213; Biermann, Note 10 at 221.

89 An exception is that the registers (see Note 64) that are used to store information can be referred to in assembly language by their special designations, for example 'D0', 'D1' and so on. Some assembly language instructions may therefore involve no absolute numerical references. For example the instruction to add the values presently in the registers D0 and D1 is written 'ADD D0, D1'. For a discussion see Clements, Note 10 at 229.

90 Software is made up of instructions and operands, but it is common to refer to a single element, usually an instruction together with the operands that it is to use as a program 'statement'. See for example, Biermann, Note 10 at 12-13.

91 That is, the central processing unit upon which the software is to be executed: Parsons, Note 69 at 3.

92 Parsons, Note 69 at 2.

It is possible to express in a higher-level language statement a notation that resembles a traditional mathematical equation. For example, a widely used higher-level language is known as 'C++'.[93] The following program statement is permissible in C++.

$$Z = X + Y;$$

In this statement, the items X, Y and Z represent 'variables'. Like mathematical variables, they are labels for quantities that can have differing values.[94] The range of values that is valid for a variable depends upon the information 'type' that is assigned to that variable and upon the word length[95] that the hardware is using. In typical implementations of the C++ language one type that can be assigned to variables is referred to as the 'signed integer' type. It can represent whole number (integer) values between -32768 and 32767.[96] Many other data types are catered for in higher-level languages.

The effect of the program statement shown above is to direct that the present value of Z be replaced by the sum of the present values of X and Y. To achieve this outcome using assembly language or machine code requires three statements. One possible implementation in assembly language that will achieve the same outcome is as follows.[97]

---

93 The C++ programming language is popular, and very widely used: Khoshafian S, *Object Orientation: Concepts, Analysis & Design, Languages, Databases, Graphical User Interfaces, Standards* (New York: Wiley, 1995) 267. The language is defined in International Organization for Standardization, *Standard No 14882 Programming Languages -- C++*, International Organization for Standardization, 1998.

94 For a general discussion see Biermann, Note 10 at 49.

95 See Note 62.

96 The signed integer information type is allocated 16 binary digits. Of these, one is used to indicate whether the value is positive or negative. The range of values that can be represented is limited to $2^{15}$, or 32768 (see Note 27). This range can be expressed as either positive or negative values. Since the expression of the value zero also has to be allowed for, it takes one of the possible positive values. The highest positive number that can be represented is therefore 32767.

97 Adapted from Biermann, Note 10 at 221-222.

| Assembly Language Instruction | Explanation |
|---|---|
| COPY AX, X | Copy the value of X into register[98] AX |
| ADD AX, Y | Get Y and add it to the quantity in AX. |
| COPY Z, AX | Put the result into Z. |

Many tasks and functions that involve considerably more complex operations and larger numbers of variables can be written in C++ and in other higher-level languages as single program statements. These statements produce, or are replaced by, larger numbers of assembly language and, ultimately, machine code statements.[99]

Other benefits that are offered by higher-level language include the opportunity to create sub-programs within the 'main' program. Such sub-programs are usually referred to as 'subroutines'.[100] Specifications for some higher-level languages use the terms 'functions' and 'procedures' as well. A subroutine commonly performs a defined function upon stated input parameters and produces or 'returns' an output that is used by the main program. The function that is performed will typically be a task that is not directly provided for in an instruction that the higher-level language implements, but which is required on several occasions within the program of instructions. The availability of subroutines offers the advantage that portions of the software can be reused. It also allows particular tasks within a program to be isolated from others.[101]

Some higher-level languages expand the opportunity for the segregation of programming tasks much further. They do this by facilitating 'object-oriented programming' in which abstract information structures (called 'objects') can be created within a program.[102] These structures hold information that relates to

---

98 See Note 64.

99 This leads to commensurate advantages in productivity: Parsons, Note 69 at 2.

100 Biermann, Note 10 at 130.

101 Biermann, Note 10 at 132, noting that the isolation of tasks can simplify the process of designing software to meet the needs of a particular problem that the software is to solve, and see also at 126-127.

102 Such objects should not be confused with graphical elements (windows, menus, icons) that are commonly presented on a visual display device. Those elements are not objects in the object-oriented

entities[103] that have a role in the real world task that the program attempts to implement.[104] As well as holding data, these structures also contain programming segments—which are largely equivalent to subroutines—that can process that data. Object orientation has been said to have an "intuitive appeal [because] it provides better concepts and tools with which to model and represent the real world as closely as possible."[105]

A basic tenet in the philosophy of object-oriented programming is 'encapsulation'.[106] Viewed from the level of the main program, the interaction between the objects is minimalist. Each object exposes some limited functionality that the other objects can call upon. The internal workings of the object are hidden from the other objects, and from the main program. Information that is internal to the object can be accessed only by means of a deliberately exposed functionality.[107] Under this approach, the program is viewed as

> a set of interacting objects, with their own private state, rather than as a set of functions ... Objects communicate by passing messages to each other and these messages initiate object operations.[108]

Object-oriented programming is said to be advantageous because it removes the risk that some other portion of the program will erroneously alter the workings and information of an object.[109] It also offers opportunities for reuse. Objects are created from templates, called 'classes'.[110] The main program must contain a definition for each class that sets out the behaviours and structure of objects that belong to that class. An object is then created within the program by a single statement that assigns

---

sense, even though an object may have some role in their creation and presentation. Objects are purely abstract elements that do not have any visible form.

103 See Biermann, Note 10 at 180, referring to "computational entit[ies]".

104 Khoshafian, Note 93 at 41.

105 Khoshafian, Note 93 at 7.

106 Biermann, Note 10 at 183.

107 For a further discussion, see Sigfried S, *Understanding Object-Oriented Software Engineering* (Piscataway, New Jersey: IEEE Press, 1996) 46-50.

108 Sommerville I, "Software Design for Reliability" in Rook P (ed), *Software Reliability Handbook* (London: Elsevier, 1990) 21-49 at 29, noting various advantages that are said to accrue from the use of object-oriented methods. The C++ language (Note 93) supports object-oriented programming.

109 Biermann, Note 10 at 183.

110 Sigfried, Note 107 at 34-35.

to it a particular, previously defined class. That object is but one 'instance' of the class, because other objects of the same class can be produced by the same means.[111]

New class definitions can be derived from existing classes. When this is done, the new class (called a 'subclass') takes on all the existing behaviours and structure of the 'parent' class in a process that is called 'inheritance'.[112] Object-oriented programming extends this genealogical analogy further by the concept of 'polymorphism' in which subclasses that derive from the same parental structure can distinguish themselves in the way in which they deal with messages that are sent to them. Objects from the same class structure but differing subclasses will recognise the same messages, but will respond to them in different ways.[113]

Both the implementation of object-oriented techniques and the more straightforward recognition of subroutines creates an opportunity for standard 'libraries' of program content to be created and published for reuse in different software.[114] The availability of such software libraries can have a positive impact upon the cost of software production.[115] At the same time, the use of these libraries diminishes the extent to which the behaviour of a program can be specified, predicted or controlled by an individual software author. It also relies upon an assumption that the behaviour of the components that have been taken from the pre-existing libraries is fully known, documented and understood.

*Translation from higher to machine level language*

The kinds of program statements, definitions and structures that have been discussed in connection with higher-level languages involve orders of complexity that far

---

111 The program statement is said to 'instantiate' an object of the particular class: Sigfried, Note 107 at 35.

112 Biermann, Note 10 at 183.

113 *Biermann, Note 10 at 183. The process is (loosely) analogous with the appearance of genetic variations or 'polymorphisms' within the chromosomes of organisms that share a common heritage.*

114 For example, the well known global software producer, Microsoft, publishes a comprehensive library of classes known as the 'Microsoft Foundation Classes'. The Microsoft Foundation Classes cover programming tasks for a wide range of functions: Khoshafian, Note 93 at 418, 423.

115 The rationale behind the commercial production and distribution of such libraries is that the cost of their development is amortised over a large number of users, allowing them to be sold at a price that is equal to only a fraction of the development cost: Boehm B W, *Software Engineering Economics* (Englewood Cliffs, New Jersey: Prentice-Hall, 1981) 648.

exceed the simple machine code program statements that hardware can process. Consequently, the translation of these higher level elements to machine code involves a significant effort. The task of translation is itself performed by a special kind of program that is known as a 'compiler'.[116] In order that a compiler can perform its translation function, the higher-level program (often called the 'source code') that is provided to it must adhere to defined lexical, syntactic and semantic conventions.

In the case of higher-level languages, the lexical constraints do not relate to the limits of instruction set for the target hardware. They relate instead to the set of instructions that the compiler has derived from combinations of the basic instructions that hardware can recognise. For example, a higher-level language instruction might be provided to compute the statistical mean of a series of numbers, even though the target hardware does not provide an individual instruction that has this function. These derived instructions will constitute the set of instructions that is valid for use in the higher-level language and which, consequently, will be recognised by the compiler.

The provision for derivation of such complex instructions in a higher-level language is possible because the compiler is able to formulate the complex functions that are required from the rudimentary instructions that the hardware can understand. It does this by automatically generating a sequence of machine code instructions that, when executed, will implement the required higher-level instruction. Lexical constraints still apply, because the instructions that are valid for a higher-level language are only those which have been incorporated in the design of the compiler software.[117] Despite this, the availability of complex instructions (especially instructions that have applications to mathematical calculations) makes the use of higher-level languages very attractive.[118]

---

[116] Biermann, Note 10 at 11. See also Parsons, Note 69 at 1-15. The creation of the first compiler, in 1952, is generally credited to Grace Mary Hopper (1906-1992): Slater R, *Portraits in Silicon* (Cambridge, Massachusetts: MIT Press, 1987) 224.

[117] This design also includes software libraries that the compiler software has reference to when it is executed, see Parsons, Note 69 at 2.

[118] Biermann, Note 10 at 306.

Higher-level languages are attractive also because their design seems to recognise some of the problems that the production of software entails.

> Any program that has to deal with the real world will be complex. The real world teems with special cases, exceptions, and the general untidiness that distinguishes it from worlds like that of mathematics. Much of the evolution of programming languages has consisted of finding new ways of dealing with complexity that minimize its impact on the programmer.[119]

The kinds of syntactic and semantic requirements that are associated with machine and assembly level languages apply to the higher-level languages as well. It is the compiler rather than the hardware that imposes these, but they are equally important. In the semantic context, the requirements are critical to the effective implementation of the function that the software is intended to realise. In this regard, it is noteworthy that the role of the compiler is one of mere translation. It endeavours only to translate a form of expression between languages; it does not infer meaning. Of equal importance is the fact that the adherence to syntactic and semantic requirements is what makes it possible to employ a rule based translation process.[120] Such a process does not require the making of any intuitive judgements or attempts to interpret the programmer's intent.[121]

The outcome of the 'compilation'[122] of source code is machine code or, as it is also called in the context of language translation, 'object code'.[123] It is possible that the object code will be one of several possible *correct* translations of the higher-level language program, but it may not be the most *efficient* translation.[124] It is therefore common to apply some automated method of 'optimising' the object code. Optimisation is mainly directed to finding and removing operations that are unnecessary because they produce information that has already been produced elsewhere in the object code.[125] Optimisation involves, like compilation, further

---

119 Parsons, Note 69 at 2.

120 This is Biermann's characterisation of the compilation process: Biermann, Note 10 at 309.

121 Were this not the case, the use of software (and in turn, hardware) to perform the translation process would be impossible. For a detailed discussion of techniques used in the translation process see Biermann, Note 10 at 273-312.

122 That is, the execution of the compiler software with given source code to produce a lower-level product.

123 Parsons, Note 69 at 3.

124 Parsons, Note 69 at 213.

125 Parsons, Note 69 at 214.

alteration of the code that was originally produced by the author of the higher-level language program.

Compilation is typically augmented by the availability of software libraries that implement commonly needed functions. These functions could be created using the higher-level language itself but, because they are used so often, it is more efficient to provide them in a 'ready made' package.[126] For example, a library might contain the program code that implements a range of statistical functions that are not part of the standard instruction set for a given higher-level language. Unlike a library that contains software components that can be used within an object-oriented programming environment, these library functions are incorporated into the program directly at the machine code level. This is achieved by a process that is known as 'linking'.[127] Linking takes place after compilation, but usually in conjunction with it.[128]

### 3.4.3 Operating systems

#### 3.4.3.1 Support for higher-level language software

A compiler of the kind discussed in section 3.4.2.3 must be created in a way that meets the requirements of specific target hardware. The purpose behind the use of the compiler program is to produce machine code. Yet it is the hardware, and specifically the central processing unit, that dictates the instruction set that can be used and the format that must be adopted. The compiler therefore has to be able to produce machine code that fits with the lexical, syntactic and semantic conventions of the specific central processing unit in the hardware on which the machine code is to be executed.

This is not, however, the only hardware specific consideration. The other hardware elements, principally the input, output and information storage devices (which are

---

126 Parsons, Note 69 at 4-5.
127 Parsons, Note 69 at 5-6.
128 Parsons, Note 69 at 5.

sometimes called the 'peripheral' devices)[129] impose additional requirements. In order that they may fulfil their respective roles, the peripheral devices have to be able to communicate with the central processing unit, and it is software that must direct this communication. This means that any software that is to use particular peripheral devices must include in its program content a provision for communication between the central processing unit and these devices. Such communication must take place under the protocol(s) that each peripheral device is designed to use.

The need to interact with peripheral devices is important at the outset of the execution of any software. This is because any software that is to be executed by hardware must first be transferred to the memory, and then to the central processing unit before execution can commence. Yet it is the software that has to facilitate communication between the input devices and the central processing unit in the first place. This gives rise to a circular or 'bootstrap' type problem.

Such difficulties are due in part to the expectation that given hardware will be used to execute different software at different times.[130] The universal approach to their solution involves the use of a general software program, which is called an 'operating system'. The operating system runs whenever the hardware is in operation.[131] A principal function of the operating system is to provide facilities to control the input, output and storage devices.[132] These facilities can be accessed in a higher-level language program by simplified program statements.

---

129 See for example Clements, Note 10 at 400. The devices are peripheral because they are at the boundary between the real world and the computer.

130 Which is an expectation that applies especially to general-purpose computers, but not to specific devices that although they meet the definition of a computer, are designed only to ever operate a single piece of software. The latter devices are generally part of larger equipment that has a mechanical rather than an information processing functionality. These devices may not even provide a means by which the software that they use can be updated or replaced. The electronic devices that are part of the fuel injection system of a car, the meter in a petrol pump or the automatic control system in an aircraft are examples of such 'embedded systems'. For a discussion of these examples, see Clements, Note 10 at 7-8.

131 The initial 'bootstrap' problem of loading the operating system software when the hardware is activated is solved by providing for circuits within the hardware that can automatically search for and load the operating system from a storage device when electrical power is applied to the hardware, see Biermann, Note 10 at 326.

132 Biermann, Note 10 at 325; Clements, Note 10 at 213.

In this way, a higher-level language program can be tailored to operate with or 'on' a particular operating system and central processing unit architecture (often collectively called a 'platform'). It can be designed without regard to the different protocols for communication and interaction that specific kinds of input, output and storage devices might require. It is only the operating system that must have regard to these protocols and their variants. Although the operating system is ultimately a single program itself, it is one that is directed to the general operation of the hardware, rather than a specific real world application or end use. Programs that do not make up the operating system software and that are designed to produce some specific real world result are called 'application' programs or 'applications'.[133]

3.4.3.2 Collective operating system functionalities

In addition to addressing fundamental tasks that would otherwise have to be dealt with by each application program individually, an operating system provides functionalities that operate collectively. By their nature, these functionalities are ones that could not be implemented within a discrete application program. They relate to the operation of the hardware on a more general level. Two collective functionalities that are very commonly implemented are multitasking and memory management.

*Multitasking*

Multitasking is a technique that allows central processing unit time to be shared between more than one program.[134] Multitasking is facilitated by periodically storing in memory the 'environment' of a particular program. A program environment consists of the contents of registers, the next instruction to be processed and the values of any intermediate information that the program is using.[135] Storing the environment of a program allows its execution to be suspended and later resumed. It represents storage of the processing work that the hardware has devoted to the

---

[133] Biermann, Note 10 at 212.

[134] For descriptions see Clements, Note 10 at 539-540, and Biermann, Note 10 at 328-330, referring to the functionality as one of 'time sharing'.

[135] Clements, Note 10 at 541.

program up to the point of suspension. In turn, the previously stored environment of another program can be transferred to the central processing unit from storage. This facilitates the resumption of execution of that other program for some interval of time. If small intervals of time are used, the exchange between the central processing unit and storage programs occurs relatively frequently. This gives to the user the appearance that more than one program is being, or has been, executed *simultaneously*.[136]

Multitasking does not increase the total quantity of information processing work that a central processing unit can carry out in any time period. The process of switching between programs in fact involves a certain wasted overhead. This overhead is represented in the finite amounts of time that are devoted to the continual saving and loading of the information that makes up the state of each program. Despite this, multitasking is especially advantageous in many cases. It can, for instance, eliminate 'bottlenecks' that arise when many different programs are subject to delays in the acquisition or delivery of external information.

A program may, for example, require that information be obtained from or sent to a peripheral device. Peripheral devices involve some degree of mechanical operation and, often, some intervention by a human user. Both of these matters entail delays to which the wholly electronic central processing unit is not subject.[137] Unless the central processing unit is provided with other processing tasks to continue with while the slower peripheral devices are in operation, then it will remain idle. Multitasking is an efficiency mechanism that operates on the premise that different programs will be subject to delays at different times. On this premise, a preferable outcome is obtained when relatively small intervals of processing time are allocated to each program on an alternating basis.

Whenever an operating system is used with hardware, some degree of multitasking is mandatory. The operating system software itself requires some use of the central

---

136 Biermann, Note 10 at 328-329.

137 This is due in part to the wholly internal role of the central processing unit, which shields it from delays encountered at the interfaces between the hardware and the real world.

processing unit.[138] It must arbitrate not only the demands for processing time that are made by various application programs, but its own requirements for processing as well. Without the ability to share processing time between application programs *and* the operating system software, the hardware could only execute the latter.

*Memory management*

Memory is a persistent information storage medium that is used to support the operation of the central processing unit. In machine code form, each program will refer to particular memory addresses[139] for the storage of particular information. It is therefore possible for a single program to keep track of the memory locations that it has used. This is necessary for the prevention of problems that could arise from inadvertently overwriting information that should be retained in memory, or attempting to retrieve previously stored information from the wrong location in memory.

When the execution of more than one program is contemplated, as occurs when multitasking functionality is employed, additional problems arise. Just as the programs make use of a single central processing unit, they must also make use of a single 'pool' of memory. This necessitates the use of some system of management that will reserve portions of memory to particular programs. Without such a system, a given program could attempt to make use of specific memory locations at which information had been stored by other programs.

'Memory management' describes techniques by which memory addresses that are specified by a program are treated as conceptual, rather than literal.[140] When a program specifies a particular address, it is translated into an actual physical address.[141] When a particular memory location, say '1234', is referred to in one program, it is translated to a different physical location than the one that corresponds with the location '1234' that is 'seen' by a different program. The location '1234' is

---

138 Biermann, Note 10 at 329.
139 See Note 87.
140 *Clements*, Note 10 at 552.

said to be a 'logical', rather than a 'physical' address.[142] The use of logical addresses involves the concept of so called 'virtual memory'.[143] The memory to which a program has access is only a representation of physical memory. In this way, portions of physical memory can be reserved to different programs, and a program can use any logical address without the risk of corrupting the memory that is being used by another program.[144] This is also important because there is no way of knowing, at the time at which a program is written, how many other programs will be executed in tandem with it on given hardware at any given time.

The use of virtual memory also allows the size of the available physical memory store to be expanded by using portions of secondary storage—typically a 'hard disk'—as locations for physical memory. The translation process that a virtual memory mechanism employs can accommodate the distribution of stored information between memory and secondary storage devices.[145] A qualification to this is that the time that is required to access information from a secondary storage device is greater than the time that is required to access information from a memory device. As a result, the distribution process involves the retention in memory of the most recently used information and the periodic exchange of information between memory and secondary storage. The process is designed to increase the likelihood that information that is going to be required at a particular stage of program execution will be located in a memory device when it is needed, rather than in secondary storage.[146]

*Other collective functionalities*

Operating system software can provide for a range of other collective functionalities. Examples include a filing system, under which records of information are stored or 'saved' on a secondary storage device (such as hard disk or magnetic tape) after the

---

141 The translation is carried out by a special piece of hardware called a 'memory management unit', which can be part of the central processing unit: Clements, Note 10 at 552.

142 Clements, Note 10 at 552.

143 Clements, Note 10 at 555.

144 Clements, Note 10 at 555.

145 Clements, Note 10 at 555-556. For another description, see Biermann, Note 10 at 343-344.

146 Clements, Note 10 at 555-556.

completion of program execution.[147] Such a system presents to a program a common method for identifying, storing and retrieving collections of information within computer 'files'. Because common methods and formats are used, a file that is created by one program may be accessed by other programs.

A filing system can also facilitate the communication of information, through the storage of files on removable media (such as 'floppy' or 'compact' disks) or through transmission over an electronic network. When a network is used as the transmission medium, there is a need for all computers that use the medium to adopt a common protocol for communication. This protocol has to be able, among other things, to distinguish the individual files that are to be transmitted. Protocols of this kind include the Hyper Text Transfer Protocol[148] and the Simple Mail Transfer Protocol,[149] which are, respectively, the operational bases of the popularly known and utilised mechanisms of the World Wide Web and of electronic mail. Typically, such protocols are also supplied by the operating system as a collective functionality that is available for use by all application programs.

*Access to operating system functionalities by application programs*

In order that an application program may make use of the functionalities that the operating system provides, some reference to those functionalities has to be included in the program when it is written. Normally, this reference involves the nomination of a precompiled software library that contains the machine code that implements the relevant functionalities.

Ordinarily, the machine code that is contained in a software library is added to an application program in conjunction with compilation.[150] Many applications need to use the same or similar operating system functionalities. For example, functionalities that provide access to the filing system are used by almost every kind of application

---

147 Files are also the repositories for programs themselves, not just for information that particular programs have created: see generally Biermann, Note 10 at 335.

148 Described in Network Working Group, *Request for Comments: 2068, Hypertext Transfer Protocol HTTP/1.1* (Reston, Virginia: Internet Engineering Task Force (Secretariat), 1997).

149 Described in Postel J B, *Request for Comments: 821, Simple Mail Transfer Protocol* (Marina del Rey, California: Information Sciences Institute, University of Southern California, 1982).

program, as are the functionalities that provide access to peripheral devices, such as printers.[151] This common need for access to the same functionalities code implies that the same library code would have to be copied into the software for many different application programs.

The potential for inefficiency and wasted storage space that this common need introduces can be avoided by allowing for fully compiled programs to link to the operating system library code that they need only when those programs are executed. This technique is called 'dynamic linking'.[152] It requires that only one copy of the relevant library code be available. Different application programs can access that single copy as and when they require access to the functionality that it implements.

### 3.4.4 The software 'environment'

Section 3.4.1 characterised the role of software as one that involves the control of hardware. This role, which flows from the fundamental distinction between hardware and software, endures in the contemporary computing environment. Yet in this contemporary environment, one individual program is unlikely to be wholly responsible for that controlling function. Control is rather more likely to be divided between various programs that operate in collaboration.

The use of dynamic linking to software library functions in the manner that was described in section 3.4.3.2 is one example of the collaborative operation of different programs. The operating system software shares control of the hardware in a number of instances too. It has, for example, a distinct role in the control of the peripheral devices. This means that the hardware behaviour that a user perceives is due to the combined operation of both a specific application program and the operating system software. Therefore "most [application] software does not interact directly with

---

150 See section 3.4.2.3.

151 Since the advent of widespread use of the Internet, application programs also often require access to the functionalities that provide for connection to and communication across networks of computers.

humans; instead all inputs come from an operating system and all outputs go to an operating system."[153]

The operating system software collaborates with application software in other ways as well. Most fundamentally, it interposes itself between the application software and the central processing unit. As was described in section 3.4.3.2, the operating system software determines when the application software will be executed and, in a multitasking or time-sharing context, when that execution will be halted or resumed. Additionally, the use of memory management techniques of the kind described in section 3.4.3.2 have the effect that access to memory is also obtained by application software through the operating system software. In this case too, the means by which control of the hardware is realised involves the combined operation of different programs.

## 3.5 Conclusions

An examination of the elements of a computer in a physical context reveals that hardware and software are overwhelmingly discrete and heterogeneous items. Their contribution to the operation of a computer and therefore to the production of material by it is distinct for each element. This suggests that there is a need to consider how problems with the operation of one element might affect the overall functioning of the computer. This is one of the issues that are considered in the next chapter.

Computer hardware is characterised by 'basic' building blocks (circuits, logic gates) that appear to have a substantial foundation. The use of rules of logic and a finite, well defined instruction set suggests that a degree of confidence in the correct operation of hardware might be appropriate. Whether or not such confidence is

---

[152] For a description of the process see Wegner P, "Capital-Intensive Software Technology" in Biggerstaff T J and Perlis A J (eds), *Software Reusability Volume 1: Concepts and Models* (Reading, Massachusetts: ACM Press, 1989) 43-97 at 61-62.

[153] Whittaker J A and Voas J, "Toward a More Reliable Theory of Software Reliability" (2000) 33 *IEEE Computer* 36-42, 41.

actually warranted,[154] caution must be exercised in equating the correct operation of hardware with the correct operation of the computer. The examination of hardware in this chapter plainly indicates that there is no such equality. The correct operation of hardware in fact carries no significance in the 'real world'. If given software specifies a sequence of instructions that does not correctly specify a means of performing a function or task of interest, then it does not matter that the hardware correctly performs such tasks.

It is the software that realises the ultimate function or task that is of interest in the real world. Yet when the design and operation of software is considered in detail, it is evident that very few constraints apply that would serve to ensure that a given item of software will implement a given function or task correctly. Here the parallel with the composition of words in a natural language is compelling. The enforcement of syntactical and grammatical rules does not guarantee that those words will have any meaning, much less that they will convey a meaning that is accurate in any relevant sense.

A consideration of the process of creation of software reveals that too much significance can be attached to the logical, highly structured and well defined nature of hardware. These characteristics do not apply to software and they impose very few constraints upon it. The attempt to model the relevant aspects of the real world to which the software is to relate benefits from no guarantee that it will be successful. More importantly for present purposes, there is no inherent assurance that simply because given software has been designed to perform a given task and is expressed in a format that can be executed by hardware, it will produce output that is accurate in any relevant real world context.

A further consideration that is relevant in the present context is the complexity of the software environment and the extent to which a given application program must itself depend upon other software to function. Here, successful operation depends upon the efforts of the creators of a variety of other programs such as compilers, software

---

[154] It should be noted that the impact of hardware design errors and materials failure have not yet been considered.

libraries and operating systems. This does not mean that dependence upon other software *necessarily* lessens the prospects of success. All that is suggested is that there is a vast distinction to be drawn between the computer operating as a combination of hardware and a variety of software programs on the one hand and a simple, deterministic device that functions according to fixed rules of logic on the other.

Perhaps the most important distinction between hardware and software is one of versatility. Properly viewed, hardware performs only one function; it executes instructions that are given to it in a format that it can recognise. The real versatility that a computer exhibits is due solely to the variable nature of software. An important aspect of this versatility is the ability of software to use the faculty of conditional execution to respond differently to different input information. This versatility and the degrees of freedom that it implies have a substantial drawback: there is much less to constrain the operation of the software to any given benchmark or standard, such as the standard that the output that is produced must be accurate.

When this factor is considered with the supervening role of software in determining the real world function that a computer will perform, it is clear that great care needs to be exercised in approaching computers from the point of view that they are members of an homogeneous class. The examination in this chapter indicates clearly that there is a very great potential for a given combination of hardware and software to be unique in a variety of respects.

The matters considered in this chapter have a direct bearing upon the question of how well or how *reliably* a computer may operate. What has been demonstrated is that great caution needs to be exercised in treating computers as comparable. What is also evident is that a consideration of the functioning of software is likely to be crucial to the question of the overall reliability of a given computer, or indeed of computers generally. What is most relevant in the context of evidentiary treatment and legal fact finding is that views within the 'legal' environment are demonstrably at odds with the conclusions that are expressed here. Those views are examined in chapter five. This follows an examination of the reliability of computers, which is undertaken in chapter four.

# 4. The reliability of computers

## 4.1 Introduction

This chapter continues the examination of physical aspects of the operation of computers that commenced in chapter three. Here, attention is directed from how computers operate to the question of how well they operate. The purpose of this chapter is therefore to consider the reliability of computers, in the sense in which that term was introduced in chapter one.[1]

The chapter commences by briefly revisiting the relationship between the concepts of 'correctness of operation', reliability and accuracy of output. It then establishes that the reliability of a computer can, and must, be considered to be governed by the reliability of the constituent elements that were examined in chapter three, namely hardware and software. The extent to which each element can impact adversely upon overall reliability is also considered.

It is argued that these elements are arranged in a 'series' configuration for the purposes of reliability analysis. This has the effect of producing a 'weak link' scenario in which shortcomings in one element degrade the overall performance of the computer from the standpoint of reliability. Whereas shortcomings in the other

---

[1] See chapter one, section 1.2.1.

element can degrade the performance of the computer yet further, the perfect operation of that element can not remedy deficiencies in the first element.

Against this backdrop, the chapter explores the reliability of each element of the computer in turn. The exploration commences with software. Several problems with the prediction of software reliability generally and the ascertainment of software reliability in a specific instance are identified. Due to the 'weak link' scenario that obtains for hardware and software, these findings reduce considerably the need to examine the reliability of hardware. This latter topic is, as a result, dealt with in considerably less detail.

## 4.2 Accuracy, 'correctness of operation' and reliability

### 4.2.1 Accuracy and 'correctness of operation'

Information is accurate when it is "[e]xact, precise, correct, nice; in exact conformity to a standard or to truth."[2] Accuracy is synonymous with correctness. To be correct is to be "[in] accordance with fact, truth, or reason; free from error; exact, true, accurate; right."[3] It is plain enough that there is a close connection between the accuracy of given information and its capacity to aid the identification of truth in a rational manner. Information that is accurate must, by definition, be indicative of the truth of the subject matter with which it deals.

Accuracy can be related to correctness of operation via the information transformation process model that was introduced in chapter one.[4] As has been said, the most important aspect of this model is the fact that, for particular input, the output that is expected is defined by the process that the hardware and software have been designed to implement. The accuracy of output is dependant upon the computer operating 'correctly' in the sense that the way in which the input information is dealt with

---

2 Simpson J A and Weiner E S (eds), *The Oxford English Dictionary* (Oxford: Clarendon Press, 2nd ed, 1989) volume one, page 92.

3 Simpson and Weiner, Note 2 at volume three, page 961.

4 See chapter one, section 1.1.3.2.

conforms to the task that it is expected to perform.[5] If, for example, the relevant task is stipulated to involve some specified numerical calculations, then the expectation of correctness of operation entails an expectation that the steps that are required to effect the specified calculations will be performed according to the mathematical rules that define them.

The expectation of performance of a previously defined function has two aspects. First, there must be some process that can be performed by a computer and that will, when furnished with input information, produce an accurate output in the context of the real world task that is of interest. Second, the computer in question must be able to carry out the process reliably.

The first requirement is therefore that the relevant real world task is one that can actually be implemented in software. The creation of software involves the development of an 'algorithm', which is a specification of some sequence of operations that constitute a method for achieving the real world task that is to be performed.[6] Since the goal is to produce a program in some programming language, the algorithm has to specify steps that can be implemented in program code. The algorithm can only be made up of constituent parts that are ultimately capable of synthesis to instructions that the target hardware is able to perform.[7] Those parts must therefore have a logical foundation.[8]

A further constraint that is of interest from an engineering standpoint is the size of the task that is to be carried out. Some tasks involve such a large number of information processing operations that they cannot be performed, even by hardware that is relatively fast, within any reasonable timeframe.[9] When it is possible for a task to be

---

[5] The accuracy of output also depends upon the intended transformation being appropriate to the context in which the output is to be used. This is the issue of the difference between a computer 'not doing the right job' and a computer 'not doing the job right' that was referred to in chapter one, section 1.2.3. As mentioned in that section, this thesis considers only the latter issue.

[6] Biermann A W, *Great Ideas in Computer Science: A Gentle Introduction* (Cambridge, Massachusetts: MIT Press, 2nd ed, 1997) 43.

[7] The processing capabilities of any hardware are limited to a finite set of instructions: see chapter three.

[8] A related factor is the need for the information that will be used to be in, or to be capable of conversion to, binary digital form: see chapter three, section 3.3.1.

[9] See for example the discussions in Biermann, Note 6 at 365-388; Schneier B, *Applied Cryptography: Protocols, Algorithms, and Source Code in C* (New York: John Wiley & Sons, 2nd ed,

implemented in software because it meets these constraints, it is said to be 'computable'.[10]

The second aspect of the stated expectation—that the computer carries out the process reliably—relates to what the computer actually does after it receives the required input information. It must, for the process of interest, carry out the appropriate series of operations in the appropriate sequence and with the appropriate information.[11] It is only when this is done that the computer can be said to operate 'correctly'.

### 4.2.2 Correctness of operation and reliability

It is clear that confirmation that a computer has operated correctly on a given occasion is essential[12] to verifying the accuracy of its output. It is the case, however, that such confirmation is not readily available in a legal fact finding environment. This is due to the fact that neither hardware nor software can be scrutinised in any direct way to ascertain whether or not they have operated correctly on a given occasion. This much is evident from the account of these elements that was given in chapter three.

In the case of hardware, the relevant physical components involve dimensions, media, quantities and speeds of operation that wholly preclude the chance of direct human observation. Scrutiny of the operation of software is equally difficult. As was also discussed in chapter three, various layers of complexity are interposed between a programming intent that is originally expressed in a higher-level language for an

---

1996) 237-242. These cases of so-called 'intractable' problems (Schneier at 239) are not relevant to the present thesis because a precondition to the potential use of computer-produced material in legal fact finding is that the task that produces the material is actually completed.

[10] Biermann, Note 6 at 425-427. This does not mean that for every computable task, there is only one program that will perform it. There are many choices that are available in the use of a programming language to create a program to perform a given task. These include the order of completion of independent operations (those which are not sensitive to the results of previous operations), the naming of operands and related matters. They are analogous with the choices of 'style' that are available for the expression of a particular piece of information in a natural language. There is also the choice of which programming language to use. These matters extend the linguistic analogue that was introduced in chapter three.

[11] In the present context the question of interest is not whether a user supplied the appropriate external information. It is whether, at each of the various stages of program execution, the appropriate information (be it initial information, intermediate results or some combination of both) is used for each particular operation that is carried out.

application program and the program code that is actually executed to fulfill that intent. Not least significant is the fact that the intent of the application programmer(s) is mediated by the structures that are engineered by the creators of the operating system and library software with which the application must interact. The opportunities that are available to the legal fact finding environment to scrutinise the performance of these elements are vastly inferior to the opportunities that are available for scrutiny of other sources of information.[13]

These considerations make it necessary to refer to a *likelihood* of correct operation, rather than to a certainty of it. This likelihood can be related to the reliability of the computer in question. This is clear from the definition that was introduced in chapter one. Under it, reliability is "the ability of an item to perform a stated function under stated conditions for a stated period of time."[14] Reliability quantifies the probability of failure free operation[15] and it is this probability that permits exploration of the likelihood that on a given occasion a computer operated correctly.[16] More significantly, reliability plainly describes an underlying physical reality of the kind that legal fact finding is required to recognise.[17]

### 4.2.3 Computer elements and reliability

*Hardware and software are distinct elements*

The discussion in chapter three established that hardware and software are distinct and heterogeneous elements. At the level at which information processing takes place, the hardware exposes a finite instruction set and predetermined syntactic requirements to

---

12 Though not sufficient, since the relevant input information may be inaccurate.

13 Such as the oral testimony of witnesses. Traditional forensic tools for the critical examination of oral testimony have no direct application to computers or to the material that they produce. For a discussion see Kelly M R, "Computer Generated Evidence as a Witness Beyond Cross Examination" (1995) 17 *Journal of Products and Toxics Liability* 95-115.

14 Institute of Electrical and Electronics Engineers, *The IEEE Standard Dictionary of Electrical and Electronic Terms* (New York: Institute of Electrical and Electronics Engineers, 6th ed, 1996) 904.

15 Lyu M R, "Introduction" in Lyu M R, *Handbook of Software Reliability Engineering* (Los Alamitos: IEEE Computer Society Press, 1996) 1-25 at 5.

16 In the sense that it performed a specified function without failure. A formal definition of failure is introduced below in section 4.3.1.

17 See the discussion in chapter two, section 2.2.2.

which software has to conform. The functionality that hardware provides is inflexible in the sense that it is limited to the execution of instructions. Consequently, the decision to use computer hardware to perform a particular task will require that that task be expressed in code that meets the requirements of the hardware. If necessary, the task must be adapted to fit with the kinds of instructions that the hardware will recognise, and not vice versa.

Hardware also has a physical manifestation. Its reliability is governed not only by the validity and integrity of its design, but also by the lifespan of its physical components. These components are subject to physical wearing out and associated failure over time.[18] They are also subject to destruction in adverse environments, such as those which exhibit excessive temperature, vibration or other physical stress.[19]

Software is very different. In each instance in which it is created, it must meet some real world function or purpose. It has to conform to the "special cases, [the] exceptions, and the general untidiness"[20] of the real world. The fact that software is required to conform to the requirements of the environment in which it is to be used means that its functionality will be very much more complex that the functionality of hardware.[21] As Lyu observes "software assumes a larger burden, while based on a less firm foundation, than hardware."[22]

Considerations of failure and reliability also differ for software. For instance, there is greater scope for software to fail to perform an expected function because that function was not properly specified or understood when the software was produced.[23] Software has no physical manifestation, so physical failure does not have to be considered in analysing the reliability of software. Despite this, it is said that "the

---

[18] For a description, see Lala P K, *Fault Tolerant and Fault Testable Hardware Design* (Englewood Cliffs, New Jersey: Prentice-Hall International, 1985) 2-3.

[19] See Singpurwalla N D and Wilson S P, *Statistical Methods in Software Engineering: Reliability and Risk* (New York: Springer-Verlag, 1999) 68.

[20] Parsons T W, *Introduction to Compiler Construction* (New York: Computer Science Press, 1992) 2.

[21] Brooks F P, "No Silver Bullet: Essence and Accidents of Software Engineering" (1987) 20(4) *IEEE Computer* 10-19 at 12. See also Rook P (ed), *Software Reliability Handbook* (London: Elsevier, 1990) at ix.

[22] See Lyu, Note 15 at 4.

[23] See "Appendix B: Review of Reliability Theory, Analytical Techniques and Basis Statistics" in Lyu, Note 15, 747-779 at 754.

complexity of [the failure modes for software] rivals or surpasses the difficulties in analyzing hardware failures."[24]

*Hardware and software make distinct contributions to reliability*

A computer is a combination of two elements: hardware and software. A reference to the operation of a computer is in reality a reference to the operation of these two components. A computer can therefore be regarded, for the purposes of reliability analysis, as constituting a system that comprises more than one component. As such, it is governed by the general principles that apply to such systems.[25]

Those principles dictate that, among other things, when components are arranged in 'series', the reliability of the resulting system is no better than the reliability of the least reliable component.[26] Components are properly regarded as having been arranged in series when "the success of the system depends on the success of all the system components ...[so that] all of them must succeed for the system to succeed."[27] In this scenario the failure of one component will cause the entire system to fail, which means that the system is dependent upon its 'weakest' link.[28]

This result can be expressed quantitatively. If the reliability of each component is expressed as a value between 1 (signifying perfect reliability) and 0 (signifying a complete absence of reliability)[29] then the overall reliability of the system is given by the product of the reliability of each component.[30] If, for example, the reliability of one component in a given system is expressed as 0.8 and the reliability of another component in the same system is expressed as 0.5, then the overall reliability of the

---

24 Note 23.

25 These are discussed in, for example, Ramakumar R, *Engineering Reliability: Fundamentals and Applications* (Englewood Cliffs, New Jersey: Prentice-Hall, 1993).

26 Ramakumar, Note 25 at 148. See also Lala, Note 18 at 8.

27 Ramakumar, Note 25 at 148.

28 The familiar scenario in which an entire 'string' of decorative lights fails when a single light in the string fails is an example of this principle. The converse situation involves components arranged in 'parallel'. In this situation, the success of just one element will permit the system to operate: Ramakumar, Note 25 at 150.

29 Within the terms of the definition that was given in section 1 above, reliability can be expressed in either qualitative or quantitative terms. Because it is often considered as a probabilistic measure, quantitative expressions of reliability (R) can usefully be expressed in the range: $0 \leq R \leq 1$.

30 Ramakumar, Note 25 at 149; Lala, Note 18 at 8.

system will be just 0.4 when the two components are arranged in series. If the reliability of all but one of the components is 1, meaning that they are perfectly reliable, then the overall reliability of the system will be the reliability of the remaining component. The reliability of the system can never be greater than the reliability of the least reliable component. It will, however, be less than this when more than one component is less than perfectly reliable.

From a reliability perspective, hardware and software are properly regarded as components that are arranged in series.[31] Hardware may operate exactly in accordance with its defined function by performing correctly each instruction that the software nominates. It cannot, however, compensate for or correct mistakes in the software such as those that cause the hardware to carry out operations or to use data that are inappropriate to the task that the software is supposed to perform. In turn, software operates on the basis that the hardware will carry out correctly each instruction that is stipulated. It is not possible for software to compensate for or correct violations of this assumption.[32]

Because hardware and software are properly regarded as components that are in series, a given computer system will only ever be as reliable as the least reliable of these two elements. Significant shortcomings in the reliability of just one element will translate to a significant shortcoming in the overall reliability of the computer. Similarly, if the reliability of one element cannot be determined, then a calculation of the overall reliability of the computer cannot be performed.

---

31 For a consistent analysis see Lyu, Note 15 at 7-8.

32 Any mechanism to verify the operation of the hardware that is implemented in software must make use of hardware in order to operate. The reliability of the verifying mechanism would then depend upon the reliability of the very same element that is supposed to be the subject of verification.

### 4.2.4 Exploring reliability

#### 4.2.4.1 Agenda

The development of a strategy for exploring the reliability of computers involves choices that are commonly presented when observable events, processes and phenomena are to be investigated. Foremost, there are a number of parameters and possible points of emphasis that could influence the choices that must be made. In the present setting the course and scope of the exploration that is appropriate to the present thesis is governed by the use to which the results of any such exploration are to be put. That use relates to the evaluation and development of approaches to the evidentiary treatment of computer-produced material. It is therefore necessary briefly to foreshadow the account of the existing approaches that is given in chapter five.

That account reveals that, among other things, the predominant approaches to the evidentiary treatment of computer-produced material regard reliability as a property that is shared by computers in a general way. It is taken to be capable of assessment for computers as a class of device. Consequently, reliability is thought not to be a property that will vary in any significant way from computer to computer.[33] This is obviously a significant assumption that has the capacity to take a fundamental place in the architecture of approaches to the evidentiary treatment of computer-produced material. The kind of approach that would be appropriate if the assumption were true would differ considerably from the kind of approach that would be appropriate if the assumption were false. This suggests that one of the principal aims of any exploration of reliability must be to discover whether there is a rational basis for this contention.

A related concern that also arises out of the account that is given in chapter five is the level at which reliability is generally experienced. If, as has been assumed in some (legal) settings, reliability does not vary in any significant way, then it must follow that there is some 'level' of general reliability for computers. Is this level high, low or intermediate? The answer to this enquiry also has the potential to influence in a substantial way the kind of approach to evidentiary treatment that will be appropriate

---

[33] Or, more accurately, between different combinations of hardware and software.

having regard to the objective of rational truth identification. It may be, for instance, that the answer is that reliability is generally high in the sense that it invariably meets or exceeds some high threshold. In that case, and subject to having a rational basis for this assertion, it would be possible to regard one of the two sources of inaccuracy[34] in computer output (namely incorrect operation) as a matter that warrants little or no attention. This is in fact a position that is reflected in the more prominent approaches to evidentiary treatment. It involves an assumption of considerable breadth and magnitude. Verification of this assumption must also be a principal focus of an exploration of reliability in the present context.

A final consideration that is relevant to the exploration of reliability that is to be undertaken here arises out of the possibility that it will be concluded that the 'generalist' view of computer reliability is not sustainable. The reliability of a given combination of hardware and software may be found to be more likely to be a unique property, rather than one that is comparable with the reliability of other such combinations. In this case, it will be important to ascertain whether the reliability of a specific combination of hardware and software can be assessed. If such a specific assessment is not possible, then addressing the issue of reliability will be a major difficulty for any regime of evidentiary treatment.

4.2.4.2 Prioritisation

The matters discussed in section 4.2.4.1 establish an agenda for exploration of the reliability of computers in the context of this thesis. The matters that are to be considered can be divided initially into the exploration of the reliability that is experienced for computers generally and the assessment of reliability in a specific case. These matters should be considered in this order, because the results of the first enquiry may influence the extent to which the second enquiry is necessary.

---

[34] The sources are inaccurate input information and incorrect operation: see chapter one, section 1.1.3.2.

In light of the discussion in section 4.2.4.1, a reasonable initial hypothesis to guide consideration of the first area is that the reliability of computers generally meets or exceeds some high threshold. This means that even if there is some variation in the reliability that is experienced from computer to computer, the reliability that is experienced for any given computer will never be less than this threshold. In this sense a 'high' threshold is one that, whilst not excluding the possibility of incorrect operation that affects the accuracy of output, renders it sufficiently remote that it warrants little or no attention. In other words, the existence of such a threshold might support existing approaches to evidentiary treatment.

The matters that were discussed in section 4.2.3 present further scope for prioritisation. They suggest that if the hypothesis that has been proposed is to be established then the reliability of *both* the elements of hardware and software must satisfy the condition of high reliability on a general basis. The failure of just one element to satisfy this condition will, for the reasons given in section 4.2.3, preclude satisfaction of the condition for computers generally. This suggests that the examination of the least reliable component should be undertaken first. If just one element fails to meet the stated criterion, then this will represent an early preclusion of the hypothesis.

There is of course a degree of circularity involved in attempting to determine in a non empirical manner which component is likely to be least reliable before examining the principles that govern its reliability. This difficulty can be overcome by making an intuitive guess as to which component might be less reliable.[35] More precisely, what is sought is the component that is *least* likely to be governed by principles that constrain its reliability to a high level. As between software and hardware, software satisfies the intuitive criteria for this 'weak link' status. Software is likely to be weaker in terms of reliability because it involves a greater functional complexity. Reliability is defined in terms of ability to meet a specified function, and software has

---

[35] An incorrect guess would not, however, invalidate the examination. It would merely impede its efficiency, because it would necessitate examination of the other component as well. In terms of the outcome of the examination, nothing turns upon the initial choice and it therefore need not be arrived at by rigorous processes.

a more significant and less uniform functional role than hardware.[36] This choice is further supported by Lyu's view that

> many existing large systems face the ... situation [that] software reliability is always the bottleneck of system reliability, and the maturity of software always lags behind that of hardware.[37]

For these reasons, the reliability of software is considered first, in section 4.3. An examination of the reliability of hardware follows in section 4.4.

4.2.4.3 Methodology

The hypothesis that the reliability of computers generally meets or exceeds some high threshold might be established on one of the two following bases.

- Demonstrating that computers are observed to operate with high levels of reliability.

- Demonstrating that computers have some inherent properties that limits departure by them from 'correct' operation to some low degree or extent.

The first case refers to the possibility that the operation of computers can be characterised by reference to suitable empirical data. These data would consist of observations of the operation of computers on discrete occasions such that 'correctness' or 'incorrectness' of operation could be ascertained and recorded for each occasion. The data might be used to infer the applicability and parameters of a probability distribution from which more general conclusions might be drawn.[38]

---

36 See Notes 20-22.

37 Lyu, Note 15 at 8.

38 For a general discussion of the statistical principles involved, see Kenkel J L, *Introductory Statistics for Management and Economics* (Boston: PWS Kent, 3rd ed, 1989) 151. For a discussion of the application of statistical techniques to the measurement of software reliability in a specific, as opposed to general, case see Littlewood B, "Modelling Growth in Software Reliability" in Rook, Note 21, 137-153 at 143-144.

The critical requirement in this regard is the availability of suitably representative data. The stated hypothesis relates to *all* computers; it is not limited to specific types of devices or to some limited number of operating contexts (e.g. use by large businesses or large public agencies). The range of data that is required is necessarily vast. The data must cover the spectrum of computer systems that might be involved in the production of material that could be offered for use in legal fact finding.

These exigencies have the consequence that the possibility for establishing the hypothesis by empirical means can be dismissed. The ground for dismissal is simply that the requisite sample data do not exist, or at least that they have not been published. The requirement that such data be suitably representative—a foundation that is critical to rational statistical inference in any context—is what excludes the likelihood that such data could ever be assembled.[39] Due to the very large numbers of hardware and software components that are available, many combinations are possible.[40] Yet each different combination must be represented within any general model of, or statement about, the reliability of computers. How could data that are representative of the characteristics of all of these combinations possibly be obtained?

A further problem involves the innovation and adaptation of new and existing design and production methods for hardware and software. The collection of some representative data set, if otherwise feasible, would be outdated by the use of new methods.[41] In the case of software, a still further consideration is the distinct

---

[39] The absence of published data of this kind should be contrasted with the abundance of reports that deal with reliability data for specific items, such as individual software programs. As to these see for instance the NASA data considered by Hatton: Hatton L, "Re-examining the Fault Density-Component Size Connection" (1997) 14 *IEEE Software* 89-97 at 91, the 1971 Japanese data reported by Akiyama F, "An Example of Software System Debugging" *Proceedings, International Federation of Information Processing Societies Congress, 1971* (Amsterdam: North-Holland, 1971) 353-358, the United States Navy and NASA data discussed in Jelinski Z and Moranda P, "Software Reliability Research" in Freiberger W (ed), *Statistical Computer Performance Evaluation* (New York: Academic Press, 1972) 465–484 at 483-484 and the data from the IBM, Hitachi Software, AT & T and Nortel companies that are referenced and discussed in Jones W D and Vouk M A, "Field Data Analysis" in Lyu M R, *Handbook of Software Reliability Engineering* (Los Alamitos: IEEE Computer Society Press, 1996) 439-489 at 464-471, to name just a few. By their nature, reports of this kind do not (and cannot) meet the condition that they are representative of the characteristics of operation of computers generally.

[40] For a more detailed discussion of the possible permutations, see chapter three.

[41] That differing design and production methods can have a significant differential impact upon software reliability is demonstrated by a study reported in Smidts C, Huang X and Widmaier J C, "Producing Reliable Software: an Experiment" (2002) 61 *Journal of Systems and Software* 213-224. In that study, identical specifications for the development of software to control a security access system

influence on the item that individual designers can have. It has been suggested in this regard that

> [t]he development of each particular software product is a complex intellectual and social process that will inevitably exhibit features unique to that product.[42]

If sample data are to be used to found inferences about the reliability of computers in general, then that data must be representative of all possible cases. Merely anecdotal accounts of favourable or unfavourable experiences of computer reliability are patently inadequate to constitute the requisite data. This is a matter that has been overlooked in some of the legal literature.[43] While vivid accounts of isolated, yet catastrophic, computer failures can be used to preface discussions of the subject of the reliability of computers to some dramatic effect, they are of no assistance in the present context.

A second possible basis for establishing the hypothesis that the reliability of computers generally meets or exceeds some high threshold was referred to above. This involves the identification of some principles that impart to computers a characteristic property that, as a general rule, limits their departure from correct operation to some low level. The existence of such principles would demonstrate a suitably high level of reliability for all computers. Those principles would make such levels of reliability an *inherent* aspect of the operation of computers.[44] In such a case, the contention that computers in general exhibit high levels of reliability would properly be treated as an axiom rather than an assumption. However it may be classified, the contention would possess an adequate logical foundation for use in the contexts of evidentiary treatment and legal fact finding.

---

were provided to two groups who were each to use a different development methodology. Both groups reported the completion of a reliable product, but upon testing the reliability experienced for each differed substantially.

42 Littlewood, Note 38 at 137.

43 See for instance Peritz RJ, "Computer Data and Reliability: A Call For Authentication of Business Records Under the Federal Rules of Evidence" (1986) 80 *Northwestern University Law Review* 956-1002 at 990-999.

44 Meaning that the characteristic is an *essential* incident of the operation of the device. For example, the consumption of electrical energy is an essential incident of the operation of any electrical appliance because a principle that governs the operation of all such devices is that they require electrical energy in order to function.

For the foregoing reasons, the exploration of reliability that is undertaken here must be limited to this second possibility for satisfying the hypothesis of interest. This requires an exploration of the principles that govern the reliability of computers. As was foreshadowed in section 4.2.4.2, the reliability of software is considered first.

## 4.3 The reliability of software

### 4.3.1 Reliability, faults and failures

Software reliability can, together with some ancillary concepts, be treated as an extension of the general notion of reliability that was introduced in chapter one.[45] The concepts that are primarily important to this extension are 'faults' and 'failures'. These concepts are defined in the *IEEE Guide for Use of IEEE Standard Dictionary of Measures to Produce Reliable Software.*

> **fault.** (1) An accidental condition that causes a functional unit to perform its required function. (2) A manifestation of an error in software. A fault, if encountered, may cause a failure. Synonymous with bug.[46]
>
> **failure.** (1) The termination of the ability of a functional unit to perform its required function. (2) An event in which a system or system component does not perform a required function within specified limits. A failure may be produced when a fault is encountered.[47]

Two matters require clarification. First, the references to 'units' in these definitions have to be understood to be references to conceptual units of software, rather than to literal or physical units. Second, it has to be emphasised that a failure does not necessarily result in a complete termination of the execution of the software and does not preclude the production of material. Rather, a failure implies that some operations that were to be performed by the software (via the hardware) have not been performed as expected. A software failure can result in the production of material that appears

---

[45] Software reliability is, however, a relatively recent area of research. The earliest substantive work in this area is probably Jelinski and Moranda's in 1972: Jelinski and Moranda, Note 39.

[46] Institute of Electrical and Electronics Engineers, *IEEE Guide for Use of IEEE Standard Dictionary of Measures to Produce Reliable Software* (New York: Institute of Electrical and Electronics Engineers, 1988) 15.

[47] Institute of Electrical and Electronics Engineers, Note 46 at 15.

regular on its face, but which contains information that is inaccurate in some respect. It is precisely this case of latent inaccuracy that presents the greatest challenge to legal fact finding. The absence of obvious irregularity on the face of given material means that scrutiny of that material alone cannot be an effective means of determining the accuracy of the information that it contains.

Another definition of software faults emphasises that they are conceptual, rather than physical defects. It refers to "[a]n incorrect step, process, or data definition in a computer program."[48] The genesis of a software fault is the making of a mistake or error in the design of the software, or in the preparation of the program code. The mistake is a human one, as a definition of the term 'error' indicates. Under this definition, error is

> [h]uman action that results in software containing a fault. Examples include omission or misinterpretation of user requirements in a software specification, and incorrect translation or omission of a requirement in the design specification.[49]

It is possible to describe this situation in terms of the variability of the input information. Particular inputs will trigger particular faults and lead to failures. These failures will have consequences (be they obvious or non-obvious) for the output that is produced. Other inputs will not trigger any faults and will therefore not produce failures.[50] This analysis of the fault-failure relationship is one that has been adopted in the software reliability literature.[51] It is useful in the present context because it is compatible with the information transformation model that was introduced in chapter one. This is depicted in figure 4-1, which is an adaptation of figure 1-1.

---

48 Institute of Electrical and Electronics Engineers, Note 14 at 394.

49 Institute of Electrical and Electronics Engineers, Note 46 at 15.

50 A consequence of this classification is that the capacity of particular input information to trigger a fault depends upon the characteristics of the fault and has nothing to do with the accuracy of that information.

51 See for example Boehm B W, *Software Engineering Economics* (Englewood Cliffs, New Jersey: Prentice-Hall, 1981) 373; Laprie J and Karama K "Software Reliability and System Reliability" in Lyu, Note 15, 27-69 at 29.

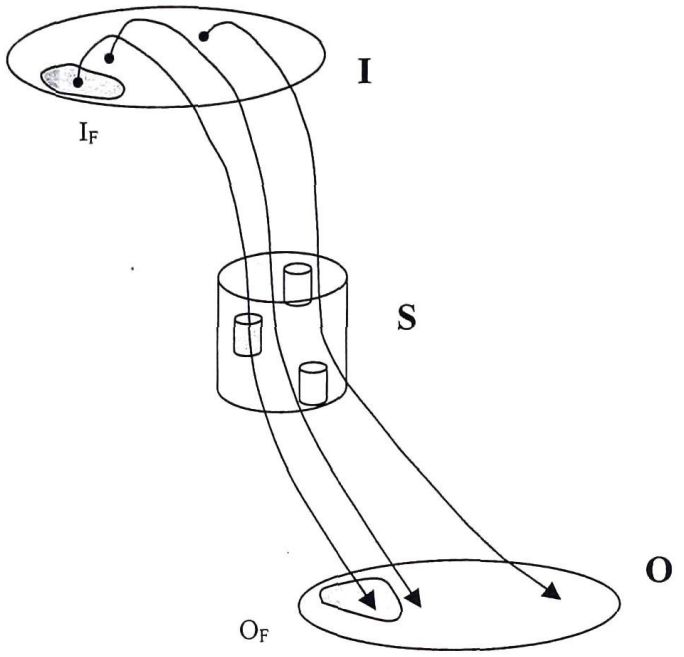



**Figure 4-1: Software faults and failures within an information transformation context**[52]

In figure 4-1, $I_F$ is a region within the input space I from which input will trigger a fault in the program S, which is a representation of particular software operating in conjunction with particular hardware. An input from within $I_F$ will result in an output that falls within the failure region $O_F$ in the output space O. Inputs that are not from within $I_F$ do not lead to failures, and their respective outputs fall outside the failure region $O_F$. Faults in the software are represented by small cylindrical regions. The input from $I_F$ interacts with, and therefore triggers, a fault. It can be said that the fault is 'sensitive' to inputs that lie within $I_F$. This region is referred to in this thesis as the 'input sensitivity region' for the software. Other inputs depicted in figure 4-1 are from outside $I_F$ and do not trigger any faults. These inputs will not cause any failure in the software.

This connection between faults, failures and software reliability is confirmed by the way in which the *IEEE Guide for Use of IEEE Standard Dictionary of Measures to Produce Reliable Software* defines the term 'software reliability'. It is

---

[52] Diagram based in part upon Boehm, Note 51 at 373, Figure 24-1.

> [t]he probability that software will not cause the failure of a system for a specified time under specified conditions. The probability is a function of the inputs to, and the use of, the system as well as a function of the existence of faults in the software. The inputs to the system determine whether existing faults, if any, are encountered.[53]

The level of software reliability that will be experienced is determined by the presence of faults and the likelihood that the program will receive the particular inputs that will trigger those faults. This highlights an important aspect of the special relationship between faults and failures in software. A failure is only ever caused by a fault, but not every fault will cause a failure.[54] This means that it is only software failures that can be detected; faults in software cannot be observed directly.[55]

Software faults are quite unlike the kinds of faults that might be encountered in physical objects. The periodic activation of software faults impacts adversely upon the reliability that is experienced, but does not degrade the software product toward total unserviceability.[56] As with computer output, software is not limited to states of obvious robustness or obvious defectiveness. In both cases, regard must be had for the potential impact of matters that are not always immediately apparent. As is demonstrated in chapter five, this matter has generally been overlooked in the legal literature.

That particular faults can be sensitive only to certain inputs is best illustrated by the well-known example of the 'Year 2000 problem'. That problem involved the conjecture that some software that utilised dates as input information would fail when the date inputs furnished to the program referred to points later in time than 31 December 1999. The mechanics of the problem, which relate to the way in which date information was represented internally in some software, are not material for

---

53 Institute of Electrical and Electronics Engineers, Note 46 at 16. Lyu adopts a comparable definition of software reliability. Under that definition, software reliability is "the probability of failure-free software operation for a specified period in a specified environment": Lyu, Note 15 at 5. For near identical definitions see Singpurwalla and Wilson, Note 19 at 67; Dale C, "Software Reliability Issues" in Rook, Note 21, 1-20 at 2.

54 Singpurwalla and Wilson, Note 19 at 67. See also Laprie and Karama, Note 51 at 28, referring to a fault as either 'dormant' or 'active'.

55 Singpurwalla and Wilson, Note 19 at 67.

56 Software can exhibit ongoing failures but unlike hardware, it does not wear out. See generally: Lala, Note 18 at 2-3; "Appendix B: Review of Reliability Theory, Analytical Techniques and Basis Statistics" in Lyu, Note 15, 747-779 at 753-754; Ramakumar, Note 25 at 59.

present purposes. The point of note is that despite the fact that these programs carried Year 2000 type faults from the time of their creation, the provision of pre-year 2000 date inputs would not trigger those faults.

The nature of the faults that exist in any given software will typically not be known, since they are regarded as accidental (and unwanted) defects in the software product.[57] This means that the size and content of the region $I_F$ will be unknown in a given case.[58] This is an important contributor to the uncertainty that is associated with the measurement of software reliability. In broad terms, however, poor reliability will be experienced for given software when the region of its input domain that is sensitive to faults is relatively large.

This is made clear when the two limiting cases are considered. A null sized sensitivity region corresponds with total reliability, since no possible input could trigger a fault and lead to a failure. A sensitivity region that is completely contiguous with the possible input domain corresponds with a total absence of reliability, since every possible input will trigger a fault and lead to a failure. The size of the input sensitivity region is plainly the critical factor.

At the same time, it is obvious that the way in which the software is used may moderate (or exacerbate) its reliability. If a given user happens almost always to provide inputs from the sensitivity region when using the software, then that user will experience very poor reliability. A user of the same software who fortuitously provides inputs that are always outside the sensitivity region will experience perfect reliability, although the software is imperfect in the sense that it contains faults. The subtlety with which faults interact with patterns of use of the software adds considerable complexity to the examination of the reliability of software on both an individual and general level.

---

57 Brooks, Note 21 at 11.

58 The depiction of the sensitivity region as a single closed area is a gross simplification of matters. To accommodate the combined effect of multiple faults, a more accurate representation would involve a variety of regions, some of which may overlap.

A further layer of complexity is added by the fact that the size and location of the input sensitivity region may also depend upon the environment in which the software is being executed. For example, it might be the case that given application software will be executed in a multitasking environment of the kind described in chapter three. This environment may impose, at various times, limitations upon the range and extent of memory and secondary storage (such as disk space) that is available to the software. If the software is unable to operate correctly despite such limitations then a potential for a fault-like sensitivity arises. This sensitivity may be related to given inputs, but it also depends upon the 'state' of the environment in which the software is being executed. A further possibility is that the operating system software may contain faults that cause it to fail to store or retrieve information to or from particular locations in memory or on disk at particular times. These factors may mean that the size and location of the input sensitivity region for given software will vary over time.

How is this analysis of the input-fault-failure relationship related to the question that is of present concern? What is sought here is to identify governing principles that will give rise to a high minimum level of reliability that is characteristically met or exceeded by software generally. The reliability of software is the result of the combined effect of patterns of use and fault content.[59] Attainment of the required level of reliability for all software can be seen as the result of a favourable manifestation of one or other of the following two factors.

- A relatively small[60] input sensitivity region, as a generally occurring phenomenon.

- Patterns of use of software generally which happen to avoid to a substantial extent the inputs for which faults in the software are sensitive.

A reasonable initial assumption is that the first factor might be associated with low fault content. This may in turn be the result of some characteristic of the process of

---

[59] Institute of Electrical and Electronics Engineers, Note 46 at 16. More precisely, it is the relative size of the input sensitivity region, rather than the number of faults.

software design and production. The level of faults in software may therefore be predictable, or at least verifiable. Conversely, the second factor seems to be a matter that is more likely to be purely fortuitous. Different items of software will have different functions. The scope and type (numerical vs. textual, for instance) of the input information that might be provided to them will vary from case to case. As has been observed, software faults are considered to be accidental defects in the software product. The range and location within the input domain of the input values for which they are sensitive must also be regarded as accidental and not capable of prediction. The prospect that these matters might somehow be constrained in all cases to produce patterns of use that minimize or eradicate the influence of a given number of faults in software seems to be remote.

The present objective is to demonstrate the existence of a particular condition, namely a high minimum level of reliability for software generally. It is appropriate and necessary in such circumstances to limit enquiry to that which is demonstrable. Of the two factors considered above, it is only the first that might be demonstrable. Even if the second factor is in fact manifested in the real world, there seems to be little scope to demonstrate that this is so, other than by reference to suitable empirical evidence. For this reason, further investigation of the present question is confined to the size of the input sensitivity region and to matters that might constrain it.

### 4.3.2 The size of the input sensitivity region for software generally

#### 4.3.2.1 Factors affecting the size of the input sensitivity region

The input sensitivity region is a manifestation of the cumulative influence of all of the faults that are contained in given software. A fault is 'sensitive' only to those inputs that precipitate failures that are referable to that particular fault.[61] Some faults may be

---

[60] What is referred to here is something that is not necessarily of infinitesimally small proportions. It will be of a size that translates to a sufficiently small risk of failure in the software such that reliability can be said to equal or exceed the 'high' level that was referred to in section 4.2.4.1.

[61] The sensitivity 'footprint' for a single fault does not have to be a continuous range of information, nor does it have to be able to be represented as a single region within the input domain in a diagram of the kind shown in figure 4-1.

sensitive to a large range of values for given input information. Other faults might be sensitive to only very few of the possible values for input information.

For a 'Year 2000' type fault of the kind described in section 4.3.1, there will be a very large range of dates that could, if supplied as input information, trigger the fault and lead to a failure in the relevant software. This kind of fault will make a comparatively large contribution to the input sensitivity region for given software. Another fault may perhaps be sensitive only to date inputs within the year 2000, but not before or after that twelve-month period. This hypothetical fault will contribute to the sensitivity region for the software to a far lesser extent. A scenario in which faults contribute to the input sensitivity region differentially is depicted in figure 4-2.

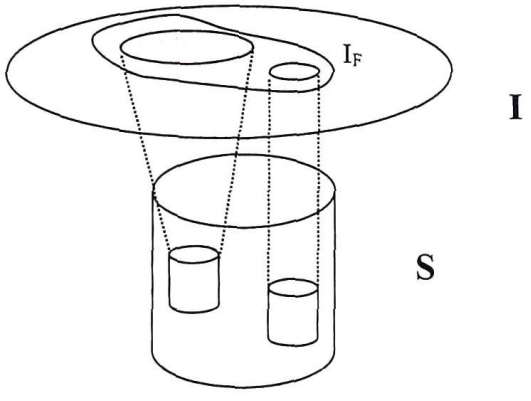



**Figure 4-2: Contribution to the input sensitivity region by individual faults**

In figure 4-2, two faults in the software S are depicted by cylindrical regions. The extent to which each contributes to the input sensitivity region $I_F$ is shown by the (unequal) elliptical regions inside $I_F$.

Despite this possibility for variation, it has been suggested that the number of faults in given software is related to its reliability.

> Although ... the relationship between perceived reliability and the number of faults in a software system is a complex one, it is generally true that the fewer the faults in a software system, the better the reliability will be.[62]

---

62 Sommerville I, "Software Design for Reliability" in Rook, Note 21, 21-49 at 21.

In reality, however, to associate the number of faults in software and the reliability of that software is to assume that the extent to which faults will contribute to the input sensitivity region will be somehow constrained to levels that are equal or closely comparable. If a general limitation upon the size of the input sensitivity region for software is to be demonstrated by reference to the number of faults in software, then there must be also be a comparability in the extent to which each fault adds to the size of that region.

It is appropriate to defer further consideration of this condition until after the question of fault numbers in software has been considered. If, for instance, it cannot be demonstrated that the number of faults in software is subject to a generally applicable limit, then the question of comparability will be redundant. The initial question of fault content, or the number of faults in software, is considered in sections 4.3.2.2-4.3.2.4.

4.3.2.2 Fault volume: inherent limits?

If a particular task can be implemented in software[63] then there is at least one program that can perform it correctly, in the sense that the expected task will be carried out unfailingly. Errors in production may, but not necessarily will, occur. The making of such errors is, to use the language adopted by Brooks, "accidental", rather than "essential", to the nature of software.[64]

While the inclusion of faults is not inevitable, there is very little about the process of programming that ensures that it will not happen. 'Compiler' software that is used to convert so called 'higher level' language software to 'executable' code[65] can detect and signal to the programmer that certain basic errors have been made.[66] These kinds of errors are limited to matters that would prevent the program from executing with any basic integrity. They do not encompass conceptual errors that may result in a

---

63 Meaning that the task can be expressed as some sequence of instructions of a kind that hardware can perform.

64 Brooks, Note 21 at 11.

65 See the description given in chapter three, section 3.4.2.3.

program that can be executed by hardware but which does not achieve the purpose that is intended by the creator of the software.[67]

The inability of compiler software to detect these kinds of errors is due to the fact that what a given program is intended to achieve in the environment outside the computer is not something of which the compiler software—or the hardware—has any cognizance. The linguistic analogues that were developed in chapter three illustrate this point. A new way of combining linguistic elements (such as individual words) might happen to be syntactically correct for some natural language, but this does not necessarily mean that the resulting combination will have a valid or significant meaning in the environment in which it is communicated.

A given number of faults in particular software represents the commission of some finite number of (presumably accidental) errors in a programming process. While the number of possible faults has a lower limit of zero, it has no corresponding upper limit. This is due to the fact that the definition of software carries with it no inherent upper limit upon fault content; an item of software that contains a ridiculously large number of faults is still an item of software. What must be considered is whether something about the manner in which software is produced may operate to impose its own limit upon fault numbers. This possibility is explored in the following section.

#### 4.3.2.3 Fault volume: other limits?

*Fault volume and program size*

The genesis of faults in software is human error. Such faults are

> ultimately human-made since [they] represents the human inability to master all the phenomena which govern the behaviour of a system."[68]

---

66 For a brief discussion of some of the techniques involved, see Parsons, Note 20 at 148-150.
67 For examples see Parsons, Note 20 at 206-207.
68 Laprie and Karama, Note 51 at 30.

This may imply that the actual extent to which faults are created is related to the extent to which the demands of a given programming task exceed the capabilities of the human programmer. Maurice Halstead explored this idea in *Elements of Software Science.*[69] Halstead's conjecture was that the extent to which human capacities would be exceeded by a given programming task could be related to the 'volume' of the program that was being created. The volume of a program was a metric that was derived by Halstead from other uniquely defined metrics that related to the number of distinct operands, distinct operators, total operands and total operators that appeared in a program.[70] Halstead then applied to this volume a parameter that was derived from assumed limitations upon the information handling capacity of the human brain and a parameter that measured the number of mental discriminations or choices that would have been required to produce the program in question.[71]

Halstead argued that the extent to which human information handling capacity was exceeded could be expressed directly in terms of the number of resulting mistakes that would be made in the programming activity. These mistakes would be manifested as 'bugs' or faults[72] in the software. The ability to associate program volume with the introduction of faults via the limiting factor of human information handling capacity was said by Halstead to lead to a relationship between the volume of a given program and the number of faults that it should be expected to contain. This relationship was said to be linear and direct.[73]

The existence of such a relationship would, if proved, mean that for a greater program volume, a higher number of faults should be expected. Halstead presented the results of an evaluation of his hypothesis against data relating to the incidence of faults in software in a large Japanese computer system that had been published by Akiyama in 1971.[74] That evaluation demonstrates a close correlation between the number

---

69 Halstead M H, *Elements of Software Science* (New York: Elsevier, 1977).
70 Halstead, Note 69 at 6, 19.
71 Halstead, Note 69 at 84-86.
72 Halstead used the terms 'bugs' and 'errors' as synonyms for faults in software: Halstead, Note 69 at 85, 87.
73 Halstead, Note 69 at 87.
74 See Akiyama, Note 39.

predicted by the hypothesised relationship and the number of faults reported to have been observed.[75]

Halstead predicted that the total numbers of faults should vary with program volume. Since program volume is plainly a variable quantity, this result does not support a contention for the existence of the condition that is of interest here, namely an upper limit upon the number of faults in software generally. Rather, the hypothesis tends to disprove the existence of the condition. Halstead's results have, in any event, been the subject of some criticism. Hamer and Frewin in particular present evidence of errors in Halstead's evaluation of his hypothesis against the 1971 data.[76] Based upon their own re-examination of that data, they argue against the existence of any linear relationship between fault counts and program volume.[77] Kitchenham[78] undertakes a more general review of Halstead's work and criticism of it, concluding that

> [t]here does not appear to be any sound evidence that the complex [Halstead] formulae are valid, or provide good measures of the product attributes they are meant to characterize.[79]

The importance of what Halstead attempted to do lies in the fact that it has been the only significant attempt to quantify the inclusion of faults in software on a general basis.[80] His hypothesis was to apply to software generally, not just to specific items of software for which empirical data had been collected. This notwithstanding, the weaknesses in the theory mean that it is of limited use in the present context. It can be taken only as indicative of the possibility that larger programs will lead to greater numbers of errors and, in turn, to higher numbers of faults. Criticisms of it do not rule out the possibility of a relationship; they merely point to an absence of evidence to

---

75 See Halstead, Note 69 at 90 (Table 11.3). Although offering other criticisms, Hamer and Frewin computed a coefficient of correlation of 0.98 for this information: Hamer P C and Frewin G D, "M. H. Halstead's Software Science: A Critical Examination" *Proceedings, 6th International Conference on Software Engineering, September 13-16, 1982, Tokyo, Japan.* (Long Beach, California: IEEE Computer Society, 1982) 197-207 at 201.

76 Hamer and Frewin, Note 75 at 201-202.

77 Hamer and Frewin, Note 75 at 202.

78 Kitchenham B, "Appendix C: Software Development Metrics and Models" in Rook, Note 21, 441-486 at 447-449.

79 Kitchenham, Note 78 at 449.

80 For an account of other work in this area see Fenton N E and Neil M, "A Critique of Software Defect Prediction Models" (1999) 25 *IEEE Transactions on Software Engineering* 675-689, 675-677.

support the existence of a direct linear relationship between program volumes (measured as a Halstead metric) and the number of faults.

In contrast, most subsequent analyses of the question of the rates at which faults are included in software have restricted themselves to specific cases. This work is relevant to the present discussion for two reasons. First, it provides evidence of a clear variation in the number of faults that have been observed in different items of software. Second, it indicates that to the extent to which any general inferences about the inclusion of faults can be made, those inferences do not support the proposition that there is some generally applicable upper limit on the number of faults that might appear in a given item of software.

The relevant studies have attempted to identify limits upon the 'fault density' of software. Fault density is a measure of faults per nominal unit of program size. These studies illustrate a potential for variability in the number of faults in given software, rather than the operation of any constraining principles that might yield an upper limit on fault numbers. One study reviewed fault data from a space satellite project.[81] The data that it used comprised information about the incidence of errors[82] in software modules of varying lengths. The authors reported that the data appeared to show that fault density tended to decrease with module length,[83] but acknowledged the possibility that the larger modules that had been studied contained additional faults that had not been detected.[84] Translating this result from fault density (number of faults per unit of program code) to the total number of faults indicates that the total number of faults should tend to increase with module size.[85]

---

81 Basili V R and Perricone B T, "Software Errors and Complexity: An Empirical Investigation" (1984) 27(1) *Communications of the ACM* 42-52.

82 The authors defined an error as "something detected within the executable code that caused the [software] module in which it occurred to perform incorrectly (i.e. contrary to its expected function)": Basili and Perricone, Note 81 at 43. This definition is synonymous with the definition of software faults that has been adopted here.

83 Basili and Perricone, Note 81 at 48.

84 Basili and Perricone, Note 81 at 48.

85 Basili and Perricone, Note 81 at 48, Table VII. The table shows a small deviation from this trend for modules of the size 150-200 lines of program code.

Another study reported a conclusion that minimal fault density is encountered in software modules of a fixed size (around 200-400 lines).[86] In that study there were two sets of data under review. Higher fault density was observed for both data sets for software modules that were either smaller or greater in size than the 'optimal' range of 200-400 lines.[87] Most notably, the fault densities for modules of similar size differed as between the two data sets.[88] Although these results arise out of very limited data, they are examples of two relevant phenomena:

- variation in the total number of faults in program modules when there are variations in program module size; and.
- variation in the total number of faults in program modules, irrespective of variations in program size.

Evidence of either phenomenon reduces support for a contention that there is a general upper limit on the number of faults that are contained in software. The fact that the studies referred to in fact provide evidence of both phenomena compounds the situation.

*Fault volume and program complexity*

A characteristic of software that may be related to the incidence of human error in performing the programming task is the complexity of the program that is being produced. This too has been associated with the presence of faults in software. The underlying idea is that the creation of 'more' complex software is more error-prone than the creation of 'less' complex software. It implies that the complexity of software can be measured in some meaningful way.

---

86 Hatton, Note 39 at 96.

87 Hatton, Note 39 at 96. For an extensive criticism of this conclusion, see Fenton and Neil, Note 80 at 681-682. That criticism is directed only to the optimum module size contention; it does not affect the contentions that are advanced here.

88 Hatton, Note 39 at 93, Figure 4.

Although the complexity of software might be measured in different ways,[89] the concept involves, in broad terms, a description of how complicated a given program is. So, for example, an early measure of complexity proposed by McCabe[90] sought to quantify complexity by reference to the number of possible paths or distinct routes that could be taken though the program code, having regard to the way in which the various conditional program statements might be executed.[91]

It appears that there are few data sets that directly support or discount the existence of a relationship between the complexity of a program and the incidence of faults in the resulting software. The existence of such a relationship is intuitively appealing and some have embraced this intuition. Munson and Khoshgoftar, for example, have argued that

> [m]easures of software complexity can be used as good predictors of software quality: for example, complex software modules are those likely to have a high fault count[92]

and that "[g]enerally, if a program module is measured and found to be complex, then it will have a large number of faults."[93] Fenton and Neil, who charactertised "size based metrics" as "poor general predictors of defect density"[94] appear prepared to acknowledge the existence of a relationship between complexity and fault volume, albeit that they observe that any such relationship "is clearly not a straightforward one."[95]

A recent study of perceptions within the software industry produced a broadly consistent finding. It suggested that there is a strong view within the software

---

89 For a review of two methodologies, see Kitchenham, Note 78 at 453-459.

90 McCabe T J, "A Complexity Measure" (1976) 2(4) *IEEE Transactions on Software Engineering* 308-320.

91 McCabe, Note 90 at 309. For a description of conditional program statements and paths of execution, see chapter three, section 3.3.4.

92 Munson J C and Khoshgoftaar T M, "Software Metrics for Reliability Assessment" in Lyu, Note 15, 493-529 at 495. See also Lew K S, Dillon T S and Forward K E, "Software Complexity and its Impact on Software Reliability" (1998) 14 *IEEE Transactions on Software Engineering* 1645-1655, 1645.

93 Munson and Khoshgoftaar, Note 92 at 510.

94 Fenton and Neil, Note 80 at 676. Fenton and Neil use the term 'defects' to describe "deviations from specifications or expectations which might lead to failures in operation" (at 675). This use is synonymous with the definition of software faults that has been adopted here.

95 Fenton and Neil, Note 80 at 680.

production industry that software reliability and complexity are related.[96] Out of thirty-two factors that were said to be capable of affecting the reliability of software, program complexity was ranked first by the study participants.[97]

The considerations reviewed here do not establish a relationship between complexity and the number of faults. They point merely to the possibility that one may exist. Complexity is a metric that can be expected to vary from case to case because the representation of dissimilar real world tasks as computer programs will necessarily involve different abstractions, different attempts to represent a solution in program code and, ultimately, different statements and structures within that code. If a relationship between complexity and the number of faults in general were demonstrated in the future, then this would further undermine the proposition that there is a general upper limit upon the number of faults in software. This is the case because complexity itself is a factor that does not have a general upper limit.[98]

4.3.2.4 The impact of software testing and fault removal

The matters that were considered in section 4.3.2.3 were confined to factors that might be supposed to have an association with the introduction of faults into a program during the course of its creation. No account was taken of the possibility that prior to use, software would be tested for the purpose of fault detection. In fact, there is widespread recognition that faults are usually introduced when a program is created. Testing as a means of fault detection is regarded as a major strategy for producing reliable software.[99]

---

96 Zhang X and Pham H, "An Analysis of Factors Affecting Software Reliability" (2000) 50 *Journal of Systems and Software* 43-56.

97 Zhang and Pham, Note 96 at 48.

98 The McCabe complexity measure increases with the number of different possible paths of execution of a program, which is in turn related to the number of conditional program statements or 'branches' in a program, see McCabe, Note 90 at 308. Like program size, this is not subject to a general upper limit.

99 Lyu, Note 15 at 20. See also Hall P, "Defect Detection and Correction" in Rook, Note 21, 111-136 at 111; Horgan J R and Mathur A P, "Software Testing and Reliability" in Lyu, Note 15, 531-566, 531.

Fault detection is synonymous with testing. It involves the operation of software in a test environment in which failures can be detected immediately because the expected result or outcome of the operation is known in advance. The premise for testing is that the discovery of faults during development provides an opportunity for the removal of those faults before the software is released for actual use.[100]

Fault detection is relevant in the present context when it is combined with fault removal. Those two activities could have the effect of reducing or eliminating the presence of software faults. If, for instance, testing and fault removal eliminated all faults in all software, then the matters considered in section 4.3.2.3 would be of no consequence. Even if such activities were partially successful in removing faults, they might give rise to an upper limit on residual fault numbers that could apply to all software. Such an outcome would support the condition that is presently being explored.

Three matters suggest that, in fact, fault detection and fault removal efforts can be expected only to lead to variable and unconstrained outcomes with respect to residual fault numbers in different items of software. These matters relate to selectivity in the test effort, selectivity in the fault removal effort and ineffectiveness of the fault removal effort.

*Selectivity in the test effort*

To detect every fault in a given program, every possible item of input information must be supplied to the program. A corresponding examination of the output in each case must then be made. In other words, the entirety of the possible input domain has to be tested.[101] This is generally not feasible since for many kinds of software the potential input domain is vast.[102] Recognition of the impracticality of exhaustive testing has led to practices that aim to test only that part of the possible input domain

---

100 See the discussion at Hall, Note 99 at 117-118.

101 This is a consequence of the fact that faults are only detectable through the exposure of the software to inputs for which the fault is sensitive: see section 4.3.1.

102 Horgan and Mathur, Note 99 at 535; Singpurwalla and Wilson, Note 19 at 3.

that is likely to be provided to the software when it is actually in use. That part is referred to as the 'operational profile' for the software. The rationale is that

> [u]sing an operational profile to guide system testing ensures that if testing is terminated and the software is shipped because of imperative schedule constraints, the most-used operations will have received the most testing and the reliability will be the maximum that is practically achievable for a given test time.[103]

The problem is that faults that are sensitive to inputs but which are not covered by the operational profile that is selected for a testing effort will not be detected nor removed. The extent to which this happens is dependent upon the appropriateness of the chosen operational profile. There is nothing inherent in these factors that suggests that this limited kind of testing will be subject to any fixed upper or (more relevantly) lower limits of efficacy.

*Selectivity in the fault removal effort*

The second reason why fault detection and fault removal efforts might lead to variable outcomes in terms of fault reduction relates to the possibility that a decision might be made to release software despite the presence of faults that were detected during testing but not removed.[104] The desire to complete a development project within predetermined constraints of time and cost is an obvious motivation for adopting such a course of action.

The extent to which these considerations will outweigh the desire to produce a reliable product will plainly vary from case to case and between different decision makers. It may be, however, that potential for variation is subject to some crude limits. A decision to release a program with an obviously overwhelming number of faults would seem to be incapable of justification on any basis. However, even this consideration is itself subject to the reality that what is 'obvious' also depends upon the perspective of the individual.

---

[103] Musa J, Fuoco G, Irving N, Kropfl D and Juhlin, B, "The Operational Profile" in Lyu, Note 15, 167-216 at 167. The implementation of testing based upon an operation profile was endorsed as 'best current practice' for the AT&T Bell Laboratories in 1991: Musa, et al at 215.

[104] See for example Hall, Note 99 at 118; Singpurwalla and Wilson, Note 19 at 191-193.

*Ineffectiveness of the fault removal effort*

The third matter that influences the efficacy of fault detection and fault removal efforts involves the possibility that an attempt to remove a detected fault could be unsuccessful. The removal of a fault in software requires the alteration of program code in some way. If the creation of that code was a process that was susceptible to human error in the first place, there is no reason why a repair effort would be immune to the same kind of error. Apart from the possibility that human errors in the fault removal process will fail to eradicate known faults, it has also been recognised that such errors can actually lead to the introduction of new faults.[105]

It is certainly the case that techniques for fault detection and removal have a potential to reduce the number of faults that are included in software during the production process. In some instances, they may be highly effective. There is, however, no basis to suppose that they will have any minimum general level of effectiveness. Their impact in ameliorating fault content in software must be regarded as variable from case to case. The relevant variability is driven by the extent to which a choice is made to use the techniques, and the effectiveness with which they are deployed.

The matters that have been considered in this and the previous section point not to a demonstrable limit upon the number of faults that are introduced into (and remain in) software in general. The situation is rather one of unconstrained variability. The most that can be said is that variable rather than uniform, or even constrained, results can be expected. Such a variability of outcomes accords entirely with Littlewood's observations about the uniqueness of individual software products.[106]

---

105 See for example Littlewood, Note 38 at 143. An attempt to account for this possibility in reliability modelling is described in Zeephongsekul P, Xia G and Kuma S, "Software-Reliability Growth Model: Primary-Failures Generate Secondary-Faults Under Imperfect Debugging" (1994) 43 *IEEE Transactions on Reliability* 408-413.

106 See Note 42.

4.3.2.5 Comparability of contribution to the input sensitivity region

An additional matter that was raised in section 4.3.2.1 was the requirement that there be comparability in the extent to which individual faults added to the size of the input sensitivity region for given software. This consideration was to have been satisfied if inferences were to be drawn from a finding that there is comparability in fault volumes for software generally. Exploration of this condition is, however, unnecessary in light of the fact that a finding of comparability in fault volumes has not been made.

What has been shown is that an inherent and generally applicable upper limit upon the number of faults in software is not demonstrable. In the result, no answer to the question of whether faults tend to contribute to the input sensitivity region comparably could assist in demonstrating the existence of some general limit on the size of that region.

### 4.3.3 The reliability of software on a general basis: an overview

It is convenient at this point to review the implications of the discussion in sections 4.3.1 and 4.3.2. What has been examined is the question of the reliability of software generally. The results of this examination do not support the contention that software generally meets or exceeds some high threshold of reliability because of characteristics that are inherent to the nature of software. All that has been indicated is an absence of 'inherent' factors that would constrain the reliability of software to any particular level, whether high, low or intermediate. The process of production of software appears to be one that involves much scope for variability in the final product. The question of how reliable software will be seems ultimately to be governed by factors for which there are many degrees of freedom.

It is noteworthy then that the software reliability literature is largely devoted to the development, measurement and expression of metrics that are intended to apply to individual items of software. In particular the development of 'models' of software reliability, a field in which there has been much activity, is devoted almost

exclusively to mathematical models that assume that each development effort will be unique.[107] Consequently, such models have application only on a case by case basis.

This is entirely consistent with the conclusions that have been developed here. Software is a variable quantity and dealing with items of software on a collective basis involves considerable difficulties. Halstead's work in *Elements of Software Science* might have been the most significant attempt to do this. What Halstead attempted to propound was a theory of homogeneity in the relationship between programmer and product for all items of software. It depended heavily upon assumptions about the comparability of programmer skills and competence and of the demands of discrete programming tasks.

Even this high water point—which must be revisited in light of the criticisms of Halstead's work—did not render a landscape that is uniform enough for the purposes that are relevant here. It is unsurprising then that the few attempts to draw general conclusions about a 'state' of software reliability that have been made are in truth founded on grounds that are less than rigorous. Consider, for example, Hatton's use of data from an analysis of software fault data for a fourteen-year period from just one centre of a single United States government agency to assert that

> [t]he simple conclusion is that the average across many languages and development efforts for "good" software is around six faults per [thousand lines of code], and that with our best techniques, we can achieve 0.5-1 fault per [thousand lines of code].[108]

This examination of the factors that may inherently govern the reliability of software further strengthens the doubts that were expressed in section 4.2.4 about the availability of suitable empirical data to found inferences about the reliability of computers. The many degrees of freedom that have been encountered must fairly be

---

[107] For a survey of the prominent models, see Farr W, "Software Reliability Modeling Survey" in Lyu, Note 15 at 71-117. For reviews of some of the earlier models see Goel A L, "Software Reliability Models: Assumptions, Limitations, and Applicability" (1985) 11 *IEEE Transactions on Software Engineering* 1411-1423 and Jelinski and Moranda, Note 39 (published in 1972).

[108] Hatton, Note 39 at 91. Other instances include Goel's assertion that software "is *often* imperfect" Goel, Note 107 at 1411 and Peled's claim that "a programmer is *highly* likely to introduce mistakes into his code": Peled D A, *Software Reliability Methods* (New York: Springer, 2001) 317. (Emphasis added in both quotations.)

represented in any data set from which conclusions might be drawn. The task of gathering suitably representative data is as formidable as it first appeared.

The most important implication of the examination of software reliability lies in its application to the reliability of computers generally. It was observed in section 4.2.3 that hardware and software are components that are in series from a reliability standpoint. The proposition that the reliability of software generally meets or exceeds some threshold is not demonstrable on an inherent or empirical basis with information and data that are presently available. With it, the proposition that the reliability of computers generally meets or exceeds some high threshold must also fall.

Because the reliability of a computer is no better than the reliability of each component, the lack of demonstrability of a general level of software reliability produces a corresponding lack of demonstrability of a general level of computer reliability. In the quantitative terms that were used in section 4.2.3, it is not possible to compute overall system reliability (the product of the reliability of each component) on a general basis. This is because the general reliability of one of the components is unknown.

This result makes any examination of the reliability of hardware unnecessary on a general basis. Even if the reliability of hardware could be shown generally to meet or exceed some high threshold, or even if it could be shown always to be perfect, the outcome would be the same. All that might result from an examination of hardware reliability on a general basis would be a discovery of similar variability. Whilst this would serve further to undermine the assumptions about the reliability of computers that have been explored here, it is unnecessary in light of the findings that have already been made.

### 4.3.4 Determining software reliability in a specific case

#### 4.3.4.1 Background and rationale

Sections 4.3.2 and 4.3.3 have produced findings that undermine a 'generalist' view of software (and therefore computer) reliability. For this reason, it is necessary to explore reliability on a specific basis. The object of this exploration is to ascertain whether the reliability of a specific computer might be assessed and, if so, what is required in order to do this. Once again, this question can be considered in terms of the reliability of each element of the computer. The reliability of software in a specific case is considered in sections 4.3.4.2 and 4.3.4.3. The reliability of hardware in a specific case is considered in conjunction with the overall treatment of the reliability of hardware in section 4.4.

#### 4.3.4.2 Software reliability as a probabilistic measure

*The application of probability theory to software reliability*

Software reliability is, by definition, a probabilistic measure that describes the occurrence of failure. The application of probability theory and statistical techniques to express this measure is common in software reliability engineering.[109] This application is an instance of the use of statistical analyses to fit some collection of observed data to a recognised pattern and then to use the known mathematical properties of that pattern to associate particular values of probability with unobserved (and possibly future) events of interest.[110]

The pattern that this kind of analysis seeks to identify is a 'probability distribution'. The use of probability distributions presupposes the existence of a random variable. A random variable is an observable thing or quantity that can assume different values

---

[109] The application of probability theory and statistical techniques is also foundational to reliability engineering in general: Ramakumar, Note 25 at 12.

[110] For a discussion of the approach to software reliability, see Littlewood, Note 38 at 143-144. For a discussion of the underlying statistical premises, see for example: Kenkel, Note 38 at 151.

in an experiment, such that the value that is assumed for any given experiment cannot be predicted with certainty.[111] For example, the toss of a coin involves a random variable, which is the side of the coin that will be shown when the coin comes to rest. The values that the variable may assume are either 'head' or 'tail'. It may be supposed that the likelihood of either outcome is the same, but nothing more can be said with certainty about the specific result that will in fact be achieved for a particular toss. Random variables may be discrete, meaning that they can assume only one of a fixed number of values, or they may be continuous, meaning they can assume any of an infinite number of values within some interval.[112]

A probability distribution is a measure, expressed mathematically and depicted graphically, that relates each possible value of a given random variable with the probability that in some experiment of interest, the variable will have that value.[113] For example, the probability distribution that describes the toss of a coin can be expressed as

$$P(X = \text{head}) = P(X = \text{tail}) = 0.5 \qquad (4\text{-}1)$$

because the probability that any toss of a coin will result in a 'head' is generally accepted to be equal to the probability of a 'tail'. In this equation, 'X' refers to the value of the random variable for a single toss.

When software is executed with particular input information, it will either operate without failure or it will fail. As discussed in section 4.3.1, the occurrence of failure depends upon the relationship between the sensitivity of faults in the software to particular input information and the input information that is supplied on a given occasion. Because these matters are generally unknown,[114] there is a random variable that is observable in the outcome of a given execution of the software. It can have the values 'failure' or 'no failure'.

---

111 Kenkel, Note 110 at 208-209.

112 Kenkel, Note 110 at 209.

113 See for example, Kenkel, Note 110 at 208-211 and 277-281.

114 Except in the rare (if not unattainable) case in which it is known with certainty that the number of faults in the software is zero. In that case, no exploration of reliability is required since, by definition, the software will have perfect reliability.

The probability that this variable will take either possible value changes each time new input information is provided to the software.[115] Usually, this will occur each time the software is executed. Although the outcome (in terms of 'failure' or 'no failure') is a random variable that is of considerable interest in the present context, the variability that is introduced by fresh input information for each execution of the software makes it unsuitable for study over a period of time. Further, the use of this variable in the absence of information about the underlying fault sensitivities permits only the assumption that, for any given execution, the outcomes of 'failure' or 'no failure' are equally likely. As a predictive tool, this assumption is of almost no usefulness. What is required is some other random variable that is connected with the execution process and for which more precise probabilities might be expressed.

By definition, software reliability refers not just to the probability of failure free performance, but the probability of failure free performance of software reliability over an interval of time. A variable that is relevant in the context of software reliability is one that indicates the points at which given software experiences failure within a time interval of interest. The postulation of this variable assumes the existence of a scenario in which there are many executions of the software with different input data over a period of time.[116] In this scenario, there might be several successive executions, each with different input information, that produce no failure in the sense that the software executes to the point at which the expected output is produced. For some executions of the software, the particular input information that is supplied will trigger a fault, which will result in a failure.[117]

The relevant underlying random variable that is observable in this scenario is the time between successive failures in the software, or the 'inter-failure' period. If the inter-failure periods are measured to any degree of fine precision[118] then it is unlikely that

---

115 It may also change over time for the same inputs if relevant changes occur in the 'state' of the software environment: see section 4.3.1.

116 See for example the description in Littlewood, Note 38 at 140.

117 Littlewood, Note 38 at 140.

118 As would be the case if for example, time is measured in seconds over a possible range of many hours or days.

exactly the same period would be observed more than once. In such a scenario, the inter-failure period is best viewed as a continuous random variable.[119]

Inter-failure periods may be measured by reference to actual calendar (or 'real world') time. Under this measure, the periods in which the software (and possibly the hardware) are not in actual operation are also counted as part of the inter-failure periods. If this measure is used during testing, it involves an assumption that during testing the real world rate of supply of input information for each execution will mirror the real world rate of supply of input information when the software is in actual operation.[120] The rate at which failures are induced under this measurement is dependant upon how much the software is used in a given interval of time.

Inter-failure periods may, in the alternative, be measured by reference to actual 'processing time'. This represents the time, for each execution of the software, during which the central processing unit is actually engaged.[121] Notably, this measurement distinguishes between individual executions of the software on the basis of the intensity of the processing that particular inputs induce. Due to the presence of program statements that invoke the conditional execution of particular program segments under particular conditions, some input values will lead to the processing of a greater number of statements than will others.[122] As a result, not every execution of the software will have the same duration. This factor is important when, as in the present context, interest lies in a retrospective assessment of the probability that a single execution resulted in failure that affected particular material. This matter is considered further below.

---

[119] For a discussion of the treatment of discrete variables as continuous, see Kenkel, Note 110 at 277.

[120] This might not be a valid assumption when the testing is automated, via a 'script' of input values that are continuously and rapidly applied to the software without human intervention. In order for the assumption to apply, the test environment must realise an acceleration of the operation of the software without altering the characteristics of the underlying failure process, except with respect to time. For a discussion of this state of 'true acceleration' see Ramakumar, Note 25 at 402-404. The possible use of automated testing is noted in Hall, Note 99 at 117.

[121] See Farr, Note 107 at 73 and 87, discussing the distinction between the two measurements.

[122] See Farr, Note 107 at 87, noting the view of Musa that processing time is a preferable measure because it is "more reflective of the actual stress induced on the software system."

*Data collection and data fitting*

The collection of data about inter-failure periods involves the observation of a suitably large number of executions of the software, each initiated by unique input information. In a test environment, the input information will be chosen after the development of an operational profile[123] for the software.[124] For each execution, any failure that occurs may be obvious, as in the case where the software simply ceases to execute, causes the operating system to signal an error condition or produces some obviously incorrect result. More importantly for present purposes, a failure may instead be non-obvious. Such a failure will be associated with the production of material that is apparently regular, but which contains inaccurate information in the sense that the information does not represent what should be contained in the material, given the functionality that is specified for the software and the particular input information that was supplied.

The ability to measure inter-failure periods in the test environment is dependent upon the ability to detect every single failure, irrespective of its obviousness. This may require considerable effort and expense if complicated independent verification of the outcome of each execution of the software is required. These constraints may limit the availability of sufficient data, which is a consideration that may in turn limit the availability of the relevant assessment techniques in a legal fact finding environment. If, however, data can be collected, then those data may be analysed with a view to identifying a pattern to the time intervals[125] within which inter-failure periods are

---

123 See section 4.3.2.4.

124 In any event, the tests that are carried out will represent only a sample of the population of possible executions of the software. The validity of inferences that are drawn from the data that is observed for the test inputs depends upon the extent to which the sample is one that is representative of the entire population of possible inputs. For a discussion of the general statistical considerations associated with sampling, see Kenkel, Note 110 at 311-314.

125 If the inter-failure period is treated as a continuous variable, then the frequency with which particular inter-failure periods occurs is expressed as the frequency with which inter-failure periods are observed to fall within a given time interval or range, rather than the frequency with which individual inter-failure periods are observed, see Kenkel, Note 110 at 277-278. The distinction can be illustrated with an example. The following inter-failure periods for given software are observed, namely t = 10, 17, 42, 63, 72, 103. When the observations are expressed in this form, the inter-failure variable (t) is treated as discrete. The same inter-failure variable can be expressed as continuous if the frequency of occurrence is related to some segment of the range of values that the variable may take. For the present example, inter-failure variable is a measurement of time so it can take any value that is greater than or equal to zero (that is, $t \geq 0$). The expression of the inter-failure period as a continuous variable for the present example is as follows.

observed to occur. This analysis may disclose the characteristics of the underlying probability distribution of the inter-failure periods as a whole.

If a probability distribution that fits with the observed failure data is identified, then it may be used to predict the probability of failures in the future. For a continuous probability distribution, this will involve the identification of a mathematical expression that associates any arbitrarily selected interval with the probability of occurrence of inter-failure durations that fall within that interval. Such a formula is referred to as a 'probability density function'.[126] A probability density function can be used to compute probabilities for any given time interval, not merely the intervals for which data have been observed.

4.3.4.3 Software reliability 'models'

*Description*

Studies of software reliability advance the process one step further by attempting to nominate, in advance of the collection of data, the kind of probability distributions that are believed to apply to the software failure process. The results of these attempts are known generally as 'software reliability models'.[127] A common assumption that underlies the models is that the software failures follow a Poisson process.[128] A Poisson process is a collection of events that has the following three characteristics.[129]

| Interval | $0 \leq t < 50$ | $50 \leq t < 100$ | $t \geq 100$ |
|---|---|---|---|
| Frequency | 3 | 2 | 1 |

126 Kenkel, Note 110 at 279.
127 See Note 107.
128 See Farr, Note 107 at 77; Singpurwalla and Wilson, Note 19 at 71; Littlewood, Note 38 at 145.
129 Kenkel, Note 110 at 279.

- The events are independent, in the sense that the occurrence of one event does not affect the probability of another event occurring in the same or any other time interval.[130]
- The probability of an event occurring is approximately proportional to the period of observation.
- There is a negligible probability that there will be more that one occurrence in any infinitesimally small interval.

The probability distribution for a Poisson process is a discrete one that is used to study the number of occurrences of an event in a specified time interval.[131] When the amount of time between occurrences of an event is of interest, as in the present case, the continuous distribution that corresponds with a Poisson process is an exponential distribution.[132] The probability density function for the exponential distribution is of the form

$$f(x) = (1/\mu)e^{-x/\mu} \qquad \text{for } x \geq 0 \qquad (4\text{-}2)$$

In this function, x represents time and must be greater than or equal to zero because time cannot be a negative amount. The parameter e is the Euler number 2.7182818284... and $\mu$ is the mean period between each occurrence of the event of interest.[133] For a software failure process, $\mu$ is the mean inter-failure period. Figure 4-3 is a graph that shows the characteristic shape of the exponential distribution for different values of the parameter $\mu$. Larger values of $\mu$ produce 'flatter' distributions, while smaller values of $\mu$ produce distributions that are 'gathered' toward the left edge of the graph.

---

130 In the context of the execution of software, observation continues until a single failure, at which time the observation period is 'reset' to zero. There is no attempt to observe multiple successive failures during a single execution of the software. This means that when the time measurement involves calendar or real world time (as opposed to processing time) the first condition may not be met, since the subjective severity of the error (in the view of the user) may vary. In some cases, the use of the software will continue because the failure was not noticed, or was regarded as trivial. In other cases, a failure may be regarded as serious, leading to cessation of the use of the software to allow investigations of the output information or related matters. In the latter cases, the use of a calendar or real world measure of time would cease to be compatible with the assumption that the failures were following a Poisson process.

131 Kenkel, Note 110 at 303.

132 Kenkel, Note 110 at 303.

133 The function and its parameters are described in Kenkel, Note 110 at 301-302.

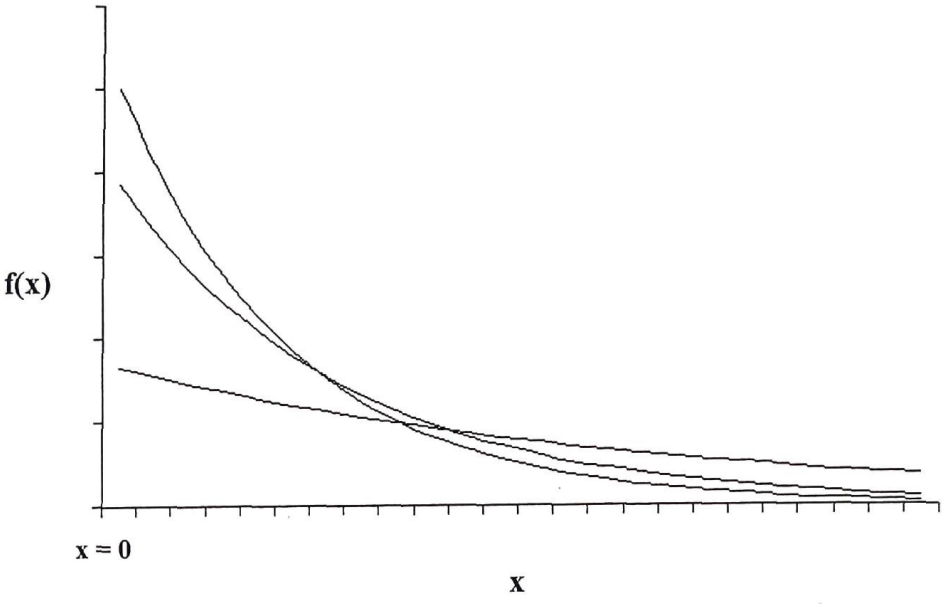



**Figure 4-3 Exponential distributions ($f(x) = (1/\mu)e^{-x/\mu}$) for different values of $\mu$**

The conjecture that inter-failure periods will be exponentially distributed implies that 'small' inter-failure periods (those that are less than the mean inter-failure period) occur much more frequently than do 'large' inter-failure periods.[134] Progressively smaller values of $\mu$ yield graphs for which the 'gathering' toward the left edge is increasingly exaggerated.

*Applicability of software reliability models in the present context*

The use to which software reliability models is most often put is the assessment of the effectiveness of fault detection and removal efforts within a development environment.[135] The articulation of a particular model of software reliability usually involves modification of the basic exponential equation through the insertion of

---

[134] This is a consequence of the fact that there is an absolute lower limit for any inter-failure period, namely zero. There can be no negative inter-failure periods, because time periods are always measured in positive numbers.

[135] Lyu, Note 15 at 18.

additional variables.[136] These variables change the shape of the probability distribution for the failure process[137] to take account (it is said) of the effects upon reliability of a reduction in the fault volume of the software over time.[138] For this reason, the models are sometimes referred to as models of software reliability 'growth'. The basic assumption that attaches to their use is that they are being applied to software that is being improved through testing and progressive fault removal.

This approach and analysis is inapplicable to software that has passed the development phase and is in actual use. Software in this phase is no longer subject to remediation to remove faults. A reasonable assumption is that software that has produced material that is offered for use in legal fact finding is in actual use rather than under development. It is more likely to constitute a completed and delivered product that is being used in a real world environment, for example in a private corporation, a government agency or perhaps even on a home personal computer. That environment will be connected with a particular legal dispute because information that is relevant to that dispute has been accumulated, processed or communicated in or from that environment.

In such a scenario, the relevant task will be to determine the past and present (but not the future) reliability of the software. Lyu refers to this activity as 'reliability estimation' as opposed to 'reliability prediction'. Its purpose is to determine

> *current* software reliability by applying statistical inference techniques to failure data obtained during system test or during system operation. This is a measure regarding the achieved reliability from the past until the current point.[139]

---

136 In some cases, a different probability distribution function is chosen as the initial descriptor of the underlying failure process. For a discussion of the details see Farr, Note 107 at 93-98.

137 Some models that are used in development environments concentrate primarily upon estimating fault volume (as opposed to reliability itself). These models employ techniques such as 'seeding' the program during testing with intentional faults that will produce a failure that can be recognised as having been caused by a 'seeded' rather than 'non-seeded' (or naturally occurring) fault. This process allows the number of non-seeded faults to be estimated by comparing the ratio of seeded to non-seeded failures over some test space. For a brief description, see Goel, Note 107 at 1416. This class of model is not relevant to the present discussion because the quantity of interest here is the reliability that is experienced, not the fault volume of the software.

138 Lyu, Note 15 at 18. For a discussion of the details see Farr, Note 107 at 77-93.

139 Lyu, Note 15 at 17.

It may nevertheless be the case that the task of finding an appropriate probability distribution for observed failure data could involve selection of a pre-existing model.

The aim in that case would be to select the model that appears best to suit the available data.[140] An attempt would then be made to extract from the data the parameters that the model requires.[141] For example, if a simple exponential distribution is selected as the 'model' that is to be used, it is still necessary to obtain the value of $\mu$ (the mean of the distribution) before the model can be used to assess the reliability of the software under review. As has been mentioned, most models use some additional parameters.[142] In each case these must be obtained or estimated from the available data.

An alternative approach is to disregard the preexisting models and to attempt to identify from the observed data the appropriate probability distribution and any necessary parameters directly from the available data.[143] This will involve an initial, largely intuitive, guess as to the probability distribution that may be involved. Statistical techniques such as regression or 'goodness-of-fit' analyses may then be used to provide a quantitative expression of the extent to which the observed data fits the hypothesised distribution.[144]

4.3.4.4 What information may be obtained?

It is pertinent at this point to reiterate the limits upon the information that might be obtained after a probability distribution (and any necessary parameters) has been obtained for the inter-failure periods for given software. What the probability density function for the distribution will indicate is the probability that an inter-failure period

---

140 Farr, Note 107 at 115.

141 Singpurwalla and Wilson, Note 19 at 102. See also Lyu, Note 15 at 17.

142 For examples, see Farr, Note 107 at 77-98.

143 For a description see Iyer K R and Lee I, "Measurement Based Analysis of Software Reliability" in Lyu, Note 15, 303-358 at 320-321.

144 For a description of the general techniques see for example: Kenkel, Note 110 at 604-605, 688-690, 715-719 and 738-743. For applications to software failure data see "Appendix B: Review of Reliability Theory, Analytical Techniques and Basis Statistics" in Lyu, Note 15, 747-779 at 772-776.

of a given duration has occurred.[145] The probability density function will not indicate directly the probability that a failure will (or did) occur at a given time.

This notwithstanding, it is possible to use the probability density function to calculate the probability that a failure occurred within a specified interval, when the location of that interval relative to the last failure is known. This may be done because, taken from the point of last failure, the time at which the next failure occurs is the same as the next inter-failure period. This is more than an indicator of reliability, it is a specific measure that is derived from the information about the reliability of specific software. It is the measure that has the greatest potential for direct application to issues that relate to evidentiary treatment. It could, for instance, be used to determine the likelihood that a software failure occurred during a specific period of interest, namely during the course of production of specific material.

A more general indicator of reliability, the mean time to failure, may also be available in some cases. This indicator is specific for given software, but 'general' in the sense that it does not vary for particular uses of the software. Unlike the first measure, it cannot be related directly to specific material of interest. The availability and usefulness of each of these two measures is considered in turn.

*Probability of failure within a particular time interval*

As was observed in section 4.3.4.2, inter-failure periods can be measured in terms of real world time or in terms of the actual processing time that is involved when the software is executed. Use of the latter measure gives rise to the opportunity to calculate the probability that a software failure occurred during a specific interval of processing time. The execution of software to produce particular material will involve a period of processing time that is small, but finite.[146] Ascertaining the size of this period might be possible, at least in theory, since an operating system can

---

[145] When the relevant perspective is retrospective, this will be the probability that an inter-failure period did occur. For instances of legal fact finding, this is the perspective of interest. What is sought in legal fact finding is information about what is likely to have taken place. The more orthodox focus in reliability studies is upon future events. In the latter sense, obtaining a probability distribution (and parameters) allows an assessment of the probability that inter-failure periods of particular durations will be experienced during future operation of the software.

facilitate the precise measurement of the computational time that is (or has been) dedicated to each application program.[147]

This information will indicate the length of the time interval of interest, but not its location relative to the last failure.[148] In general, however, the time of the last (or any) failure must be assumed to be unknown. If this were not the case, there would have to be a means of detecting each failure as and when it occurred. This means of detection would have to be effective whether or not the failures were obvious.[149] Were it available, such detection ability would render pointless any attempt to express the *likelihood* of failure at a particular time. In a legal fact finding environment, evidence of the detection of failure from this source would be a preferable means of scrutiny of given material.

The general assumption that must therefore be made is that the time of the last failure, and therefore the location of the interval of interest relative to it, are unknown. The important consequence of this assumption is that it excludes the possibility of probability calculations of the kind that have been foreshadowed when the relevant probability density function exhibits 'memory'. A probability density function exhibits memory when probability of failure in any given time interval that the function describes depends upon the period of time that has elapsed since the last failure. An example of this situation arises when a component 'ages' and experiences failure on an increasingly frequent basis.

A 'memoryless' function, by contrast, is one for which the probability of failure in any given time interval is constant. The underlying process does not 'remember' how much time has elapsed since the last failure.[150] A component whose performance is

---

146 Consider the measures of processing speed that were discussed in chapter three.

147 This is an incident of multitasking capabilities, such as those discussed in chapter three. Implementation of multitasking requires accurate monitoring of the processor time that is allocated to each program in order that a time sharing scheme can be implemented and managed. See for example the description given in Clements A, *The Principles of Computer Hardware* (Oxford: Oxford University Press, 3rd ed, 2000) 541-544.

148 That is, the period of time that elapses between the last failure and the commencement of the interval of interest.

149 In each case, the required capability could be met by the use of some method of independent verification of the results of each execution of the software in question for which there is a suitably high degree of confidence.

150 For a discussion see Ramakumar, Note 25 at 89-92.

governed by such a process will not wear out; it will only be subject to random failures.[151] Such a component has been said not to "degrade in quality with the time of operation."[152] This kind of behaviour appears to be consistent with the operation and failure of software, as those phenomena have been described in this chapter. It happens to be the case that the exponential distribution, which is commonly used as a foundation for modeling the reliability of software,[153] is memoryless.[154]

This means that the probability of failure for any given period of execution of software is dependent only upon the length of that period; it is independent of the location of that interval relative to the last failure. When the variable of interest (in this case, time) is a continuous random variable, the only memoryless distribution is the exponential distribution. This distribution will be applicable only where the underlying failure process is Poisson. In any other case, calculation of the probability of failure in a given time period of interest will not be possible if the location of that period relative to the last failure is unknown.

If a calculation of the probability of failure can be made, it will yield a numerical result which is between zero and one.[155] Were such a quantitative measure available for application in the legal fact finding environment, it would be necessary to highlight that it is subject to the accuracy of the chosen probability density function as a true descriptor of the underlying failure process. Although the measure seems relatively straightforward to obtain (in terms of the mathematical operations that are involved), its availability is limited by the extent to which an appropriate probability density function[156] and parameters can be identified. It is this limitation which largely excludes the prospect that this measure might be used in connection with a regime of

---

151 Components which do wear out, but for which a period of 'useful life' can be identified can be treated as falling within this category until the commencement of their wear out period: Ramakumar, Note 25 at 89.

152 Ramakumar, Note 25 at 92.

153 See section 4.3.4.3.

154 For a formal proof of this property see, for example, Ramakumar, Note 25 at 92.

155 In the case of an exponential distribution the relevant equation to calculate probability is $1 - e^{-t/\mu}$ (see for example Ramakumar, Note 25 at 92). In this equation, the parameter e is the Euler number, the parameter t is the length of the period of interest and the parameter $\mu$ is the mean period between each occurrence of the event of interest. The limiting cases for this equation are 0 and 1 (at $t = 0$, $1 - e^{-0/\mu} = 0$ and as $t \rightarrow \infty$, $1 - e^{-t/\mu} \rightarrow 1$ since $e^{-t/\mu}$ will approach 0 as $t \rightarrow \infty$).

156 Bearing in mind that the function must be 'memoryless' in light of the assumption that individual failures cannot be detected.

evidentiary treatment. The limitation also applies to the availability of the mean time to failure measure and is therefore discussed after that measure is considered.

*Mean time to failure*

When the inter-failure periods are exponentially distributed (that is, when the failure process is Poisson), the mean inter-failure period for that software is the parameter $\mu$. This parameter is constant for a single distribution; it does not vary with time. The mean inter-failure period is also known as the 'mean time to failure' and in engineering environments, this is the information that is usually of greatest interest.[157] It is a measure that says something about the software as a whole and, as such, it has a potential for importance in development contexts in which targets of quality or reliability exist. The mean time to failure for other distributions also does not vary with time,[158] so long as the failure process continues to be governed by the same distribution.

The mean time to failure is a more general indicator of the reliability of given software than a computation that is tied to a particular instance of execution, but it is still an unique value for each piece of software. Because the parameter is not time dependant, it can be ascertained and applied without knowledge of the location of any time interval of interest relative to the preceding failure. Knowledge of the width of that interval is also not required. Once sufficient test data have been obtained, there is no ongoing need to know the times at which the software fails. There is also no need to ascertain the processing time that should be attributed to the production of material of interest.

The drawback is that these dispensations come at the cost of precision. The mean time to failure is only a broad indication of reliability.[159] A larger mean time to failure signifies a lower frequency of failure over time and therefore a greater

---

157 "Appendix B: Review of Reliability Theory, Analytical Techniques and Basis Statistics" in Lyu, Note 15, 747-779 at 756; Singpurwalla and Wilson, Note 19 at 41.

158 See for example Appendix B: Review of Reliability Theory, Analytical Techniques and Basis Statistics" in Lyu, Note 15, 747-779 at 756-757.

159 For a criticism of the use of this information to quantify software reliability as potentially misleading, see Dale, Note 53 at 4.

reliability, but it does not alter the fact that failures will still occur, or the fact that failure is still associated with an underlying random element.

Quotation of a large mean time to failure period may instill false confidence and in the legal fact finding environment it may be misleading. It obscures the fact that failure can occur at any time, that failures are almost certainly not uniformly distributed for any given time domain and that more than one failure could have occurred within the quoted 'mean' period. It is highly questionable that a legal fact finding environment could meaningfully interpret and use this parameter to inform itself about the probability of failure in connection with individual items of computer-produced material. The more fundamental limitation, however, is one that applies to the measure that entails an actual calculation of the probability of failure. This limitation is considered in the following section.

4.3.4.5 The impact of the need to have suitable prior failure data

As has been shown, a precondition to the use of either of the two measures that were considered in section 4.3.4.4 is the identification of the probability density function (and any necessary parameters) that accurately describes the underlying failure process for the specific software of interest. As was discussed in section 4.3.4.2, this requires the collection and analysis of suitable test data. It may, depending upon the nature of a regime of evidentiary treatment that seeks to make use of such measures, require the fact finder to make an assessment as to the validity and integrity of the data collection and analysis techniques that have been used.

The need to have sufficient failure data is one of the most difficult to address in the present context. Data that may have been collected during the development of the software cannot be used for this purpose if those data were collected during any period in which ongoing modification to the software was taking place. In addition, the 'state'[160] of the environment in which the software operates cannot change if the data that are collected are to be taken to be representative of the reliability that will be

[160] See section 4.3.1.

experienced after their collection. For the observed data to have validity, the software environment must remain unaltered during both the collection of data and the ongoing use of the software of interest.

The principal problem is that collection of test data requires the methodical observation and verification of each execution of the software. It is reasonable to assume that this will necessarily involve the expenditure of significant time and cost. It is also reasonable to assume that in a given real world context, this expenditure would not be incurred simply for the benefit of developing test data in isolation from any intent to remove the faults that the testing may reveal. Any such testing effort is more likely to be carried out in order to facilitate the removal of faults from the software and, consequently, to improve its reliability.[161]

As has been said, it is, however, only data that are collected *without* subsequent fault removal that may be used to identify a probability distribution that will have validity for the software on any ongoing basis. To require that such data be collected without permitting the rectification of faults in tandem with the test effort will impose a substantial additional cost in any given environment. It seems highly unrealistic to seek to impose such expense solely to facilitate the admission of material that is produced by the software for the purposes of an isolated instance of legal fact finding. The more likely scenario is that availability of data of the kind that are required will be purely a matter of chance. Chance is, however, plainly an unsatisfactory basis for the application and operation of a regime of evidentiary treatment.

---

[161] It might be argued that test data produced during development could indicate a 'worst case' picture for reliability and that it could be used on the assumption that subsequent fault removal efforts would only alter the reliability of the software in a favourable way. This might be said to support the use of probability measures and calculation that are based upon such worst case test results. The difficulty with this argument is that it ignores the possibility of 'imperfect debugging', in which attempts to remove identified faults actually introduce new faults, see section 4.3.2.4 and Zeephongsekul, Xia and Kuma, Note 105.

## 4.4 The reliability of hardware

### 4.4.1 Concepts of reliability, faults and failures in hardware

The concepts of reliability, faults and failures that were introduced and discussed in section 4.3.1 in connection with software apply also to hardware.[162] Faults in hardware have, however, origins both in design errors and also in defective, damaged or worn out physical components.[163] Faults in hardware can also be classified according to their effect as either 'logical' or 'non-logical'.[164]

> A *logical* fault causes the logic value at a point in a circuit to become the opposite of the specified value. *Non-logical* faults include the rest of the faults such as the malfunction of the clock signal, power failure. etc.[165]

More significantly for present purposes, failures in hardware can be regarded as having a randomness that is, for the 'useful' life of the component, similar to the randomness that is associated with failures in software.[166] Consistently, the assumption that inter-failure periods are exponentially distributed during the term of the useful life—as they are for the entire 'life' of software—appears to be a common one.[167] The period for which a hardware component is considered useful is preceded by an initial period in which there is a high but decreasing failure rate due to material or manufacturing defects. This is followed by a final period in which the failure rate increases due to wearing out and associated damage to the physical components.[168]

---

[162] See for example Lala, Note 18 at 1-7, 12.

[163] See Lala, Note 18 at 12; Singpurwalla and Wilson, Note 19 at 68.

[164] Lala, Note 18 at 12.

[165] Lala, Note 18 at 12.

[166] Lala, Note 18 at 2-3; "Appendix B: Review of Reliability Theory, Analytical Techniques and Basis Statistics" in Lyu, Note 15, 747-779 at 753-754.

[167] See for example, Lala, Note 18 at 3-4; "Appendix B: Review of Reliability Theory, Analytical Techniques and Basis Statistics" in Lyu, Note 15, 747-779 at 753.

[168] Lala, Note 18 at 2-3; "Appendix B: Review of Reliability Theory, Analytical Techniques and Basis Statistics" in Lyu, Note 15, 747-779 at 753-754; Ramakumar, Note 25 at 59.

### 4.4.2 Application of hardware reliability measures

The reliability of hardware can be associated with, and expressed in terms of, particular quantitative measures. The potential for the application of these measures to legal fact finding is also capable of exploration in both general and specific contexts. The need to undertake such an investigation is, however, qualified in the present context by the results that were obtained in section 4.3. The qualification arises because, as was described in section 4.2.3, hardware and software contribute to the overall reliability of a computer system as elements in 'series'.

This means that the reliability of the system, and the ways in which measures of that reliability can be applied, depend upon the 'weaker' element. If software is, in either a general or a specific context, the weaker element in this context, then the relevance of the reliability of hardware is nominal. The investigation of the potential application of measures of hardware reliability to legal fact finding that is undertaken here is abbreviated because of this consideration.

*A minimum general threshold of hardware reliability and its relevance*

It was shown in section 4.2.3 that there is no support for the conjecture that the reliability of software generally meets or exceeds some minimum threshold. It is possible that a similar investigation of hardware reliability might reach different conclusions. Hardware and software are, after all, fundamentally different. Hardware is regarded, for example, as assuming a less complex role than software.[169] This is potentially significant because complexity is a factor that might be associated with errors and associated faults in software.[170]

Considerations such as these do not, however, support particular conclusions about the reliability of hardware as it may be compared to the reliability of software. They merely raise the possibility that the results of a detailed investigation, or suitably

---

169 See section 4.2.3.

170 See the discussion in section 4.3.2.3. Against this, there is the consideration that hardware is subject to a number of intermittent faults that are induced by unpredictable elements in the physical

representative data, might reveal that a minimum general threshold of hardware reliability does exist. Yet even if this were the case, the absence of support for, and the inability to quantify, a minimum general threshold of software reliability renders any information about hardware reliability redundant in a general context. Because hardware and software contribute to the reliability of a system in series, the inability to place a lower limit upon the reliability of software in general produces a commensurate inability to place a lower limit upon the reliability of computers[171] in general.

*Hardware and system reliability in a specific case*

The potential application of hardware reliability measures in a specific case is a matter that may hold more promise. In light of the fact that hardware and software are components in series, this application will be required in any case in which specific measures of software reliability are to be used. Sections 4.3.4.2 and 4.3.4.3 described the fitting of observed data to a particular probability distribution to derive information about the occurrence of failures in software. The techniques involved are also applicable to considerations of the reliability of hardware.[172]

It may, at least in theory, be possible to adopt a 'combinatorial' strategy at this point and view the assessment of reliability from the perspective of the computer as a single unit. Under this strategy, the computer would be regarded as a suitable subject for which reliability data could be obtained and about which reliability assessments could be made. This would validate the collection and use of failure data without regard to the underlying contribution of each element to individual failures.

This would be possible because the concern of evidentiary treatment does not extend to removal of the underlying faults. This being the case, it does not matter in the present context whether a particular failure was caused by a software fault or a hardware fault. All that is required is the correct identification of a probability

---

environment, including matters as subtle as levels of background radiation: see Lala, Note 18 at 20. These aspects of the physical environment do not affect software.

171 Or, more accurately, upon the reliability of combinations of hardware and software.

172 See Lala, Note 18 at 1-6.

distribution for the failure process for the overall system. It is noteworthy, however, that the failures that are observed from this combinatorial perspective are all failures that the system experiences, namely the aggregate of all hardware failures and all software failures. In this view, a computer system is—despite the fact that it is made up of heterogeneous elements—a single functional entity that can experience failure in some defined time period.

This view is in fact consistent with the rationale that appears to underlie some approaches to the evidentiary treatment of computer-produced material that are implemented in specific statutory provisions.[173] These approaches focus upon the computer as a whole, referring, for instance, to the admissibility of statements that are "produced by [a] computer" during a period for which "the computer was operating properly...".[174] A problem with such approaches is that they fail to account for the random nature of failures in software. In doing this they make no mention of the need to identify a probability distribution for the underlying failure process, and they do not consider the need to obtain suitable test data. These and related matters are examined in detail in chapter five.

The present (and unresolved) difficulty that attends the use of measures of hardware (or overall) reliability in a specific case involves the need to obtain suitable prior test data. As was discussed in section 4.3.5.4, this difficulty emerges as one that overshadows most, if not all others. Except where the required data are fortuitously available, it represents an insurmountable obstacle to the formulation and application of approaches to evidentiary treatment that seek to deal with the issue of reliability on a specific basis.

## 4.5 Conclusions

Reliability is important in the context of the present thesis because it has a direct bearing upon the likelihood that given computer output will be accurate. The

---

[173] For example, *Evidence Act* 1958 (Vic), s 55B(1); *Evidence Act* 1977 (Qld), s 95(1); *Civil Evidence Act* 1968 (UK), s 5 (now repealed).

reliability of a computer is governed by the reliability of its elements, namely hardware and software. Hardware and software are distinct, heterogeneous elements and they each exhibit a particular (though not always measurable) level of reliability. These elements are, from a reliability standpoint, arranged in 'series' with the result that imperfections in each element have a cumulative negative effect on the overall reliability of the computer as a system. The reliability of a given computer, or of computers generally, is governed by the 'weakest' of these two elements.

The level of reliability that is experienced for software is associated with two matters. They are the existence of faults in the software that are sensitive to particular information, and the frequency with which that information (as opposed to information for which there is no sensitivity) is supplied to the software over a given period of time. When information for which a fault is sensitive is supplied to the software, a failure will occur. Reliability is an expression of the likelihood of failure such that a greater number of failures over a given period is considered to represent lower reliability.

This conceptualisation of the reliability of software and of the failure process for software appears initially to be quite simple. That simplicity is eroded by a number of complicating factors. First, faults are manifestations of accidental human errors; their location and quantity is inherently variable. Second, the selection by a user of inputs that fall within regions of fault sensitivity within the input domain (and which therefore induce failure in the software) is a random process. That is, the particular way in which the software is used has no *a priori* connection with the regions of sensitivities for faults that reside in the software. Third, when software is in use, faults can only be detected when they induce failure.

The reliability of software is, because of such factors, governed by phenomena that are both variable and incapable of precise quantification. This suggests that difficulty will be encountered in attempting to characterise the reliability of software in a general way. A reasonable initial intuition may be that it will also be difficult to assess the reliability of specific items of software. For this thesis, there is, however, a

---

[174] See *Evidence Act* 1958 (Vic), s 55B(2)(a) and (c).

need to consider these two possibilities in detail. That need is dictated by the manner in which existing approaches to the evidentiary treatment of computer-produced material have attempted to deal with the question of reliability. Those attempts, which were alluded to in this chapter, are examined in chapter five.

As to the first possibility, an extensive examination of the factors that contribute to the reliability of software has shown that there is no support for the conjecture that software generally exhibits reliability that equals or exceeds some minimum threshold. Standing back from the intricacies which have been explored in this chapter, it may seem that a belief that the reliability of computers in general meets or exceeds some high minimum level is neither incredible nor far-fetched. The point to be made is that such a belief must be regarded as wholly arbitrary. It is unsuited to an environment in which truth identification by *rational* means is an important goal. The implications of the adoption of such a belief in the absence of a foundation for doing so are made clear when the existing approaches to evidentiary treatment are examined.

The measurement of the reliability of particular software is possible, but only if certain conditions are met. These conditions are onerous and appear for the most part to exclude the viability of an approach to evidentiary treatment that is based upon the assumption that they will be met in any given case. The most significant condition relates to the availability of suitably comprehensive test data that have been obtained *after* the cessation of development of the software. These data must be sufficiently representative to enable the identification of a probability density function that accurately describes the failure process for the software of interest.

This condition is onerous because the collection of test data requires that the output produced during testing be verified by some independent mechanism so that each instance of failure, whether obvious or not, can be recorded. Further, this data collection must occur at a time at which no remedial work is being carried out. Finally, the data must reflect the range of possible changes in the 'state' of the environment in which the software might be operated. This may be a particular difficulty where software is tested on particular hardware but then sold to many users

who each operate the software on other hardware and in conjunction with other software.

If this (potentially insurmountable) difficulty can be overcome, it is possible that two indicators of reliability for specific software may be available to the legal fact finding environment. Only one of these, the mean time to failure, will be available when the relevant probability density function is not 'memoryless'. As has been discussed, it is far from clear that either indicator could be used effectively in connection with a regime of evidentiary treatment. In this respect the greatest concern is with the fact that what such probabilistic measures are ultimately describing is largely a random process.

The extent to which statistical techniques can reduce the uncertainty that is inherent in any random process is limited. This point is saliently reflected in the nature of the Poisson process, which is a common starting point for 'models' of software reliability. The times between events that follow a Poisson process are exponentially distributed, but the probability of occurrence of any single event is completely independent of the time that may have elapsed since the occurrence of the preceding event. A further basis for caution is the fact that the process of fitting a probability distribution to sample data is itself an imprecise activity. Difficulties in verifying the adequacy of the sample data that have been used and the validity of the way in which they have been used represent yet further areas of concern.

The reliability of hardware is capable of analysis on similar bases. Like software, hardware might be postulated to exhibit a minimum general threshold of reliability. Because of the discrete contributions to reliability of hardware and software, this postulate is necessarily implied in the commonly encountered contention that the reliability of *computers* generally meets or exceeds some threshold. The investigation of this postulate is, however, unnecessary. This is because it is the combined reliability of both elements that is important in the context of material that is produced by a computer. The lack of support for the postulate that all software exhibits reliability that equals or exceeds some minimum threshold means that the satisfaction of this postulate for hardware would not alter the conclusions that can be drawn from the investigation of the reliability of software alone.

For specific hardware, the use of a measure of reliability is possible. More importantly, the use of a specific measure that encompasses the combined effect of the operation of both hardware and software is also possible. This measure may be derived when failure data for a given computer are analysed without regard to the origin of recorded failures as either software or hardware based. Again, the limitations that the need for test data imposes are relevant in this context and seem to exclude the possibility that measures of the kind that were considered in connection with software might be available in a legal fact finding setting.

The results that have been presented in this chapter represent real and significant limitations for the formulation of regimes of evidentiary treatment. On the one hand there is a clear need to address the question of reliability because it has a direct connection to the issue of accuracy. On the other hand, the rational quantification or qualification of this property seems near impossible on either a general or specific basis. The next chapter highlights, among other things, that existing approaches to evidentiary treatment have almost entirely ignored the difficulties that have been identified here. It presents an evaluation of the principal approaches to evidentiary treatment by reference to the extent to which the assumptions upon which they rely are congruent with the findings that have been presented here.

# 5. Current approaches to evidentiary treatment

## 5.1 Introduction

The purpose of this chapter is to apply the findings of chapters three and four to the evaluation of existing approaches to the evidentiary treatment of computer-produced material. What is examined here is how each approach deals with the issue of the reliability of computers. As was demonstrated in chapter one, reliability is a matter that affects the accuracy of information that a computer produces. Yet, as the discussion in chapter four establishes, assessing reliability on either a general or a specific basis is an extremely difficult task. These two considerations confront any approach to evidentiary treatment and how they are recognised and addressed is the focus of this chapter. What is considered in particular is the extent to which these matters are dealt with in a rational manner by existing approaches to evidentiary treatment.[1]

Before undertaking an evaluation of the existing approaches to evidentiary treatment, this chapter first presents a scheme of classification for the approaches and a rationale

---

[1] As stated in chapter one, the evaluation that is undertaken in this thesis is limited in two important respects. First, it deals only with the how the principal goal of rational truth identification is promoted by approaches to evidentiary treatment. Second, it is only the manner in which the issue of reliability is addressed that is considered in the context of this goal.

for adopting that scheme. Each approach is then considered in turn. In evaluating the approaches to evidentiary treatment, the chapter considers a number of different 'expressions' of each approach. These expressions are drawn from English, Australian and American jurisdictions. In some cases the expressions are sufficiently congruent that the evaluation produces a common outcome. In other cases, a more discrete consideration of the different expressions of an approach is required.

In all cases, the principal question that is asked is whether there is a sustainable foundation for the way in which the issue of reliability is dealt with by a particular approach. In a number of cases, it is evident that the issue of reliability is addressed largely or solely by reference to particular assumptions. Here a further enquiry is necessary. It is whether those assumptions are adequately justified, having regard to the ideal of rationality that is integral to legal fact finding. In answering this question, considerable use is made of the findings about the reliability of computers that were presented in chapter four.

## 5.2 Evaluating the principal approaches to evidentiary treatment

### 5.2.1 Classifying the approaches

#### 5.2.1.1 A scheme for classifying the approaches

Evidentiary treatment is manifested in regimes that govern how and under what conditions material may be used in legal fact finding.[2] Different frameworks for the classification of approaches to evidentiary treatment are possible. Approaches may, for instance, be classified according to the particular rules of evidence that they modify, replace or introduce. Approaches might alternatively be classified according to the manner in which they characterise the material with which they deal, or to other foundations or motivations.

---

[2] See the definition of this term in chapter one, section 1.1.1.2.

An important distinction between these two frameworks is the different emphasis that is placed upon the underlying motivations for the approach on the one hand, and its form of expression on the other. In each case, a particular scheme of classification that might be adopted will not differentiate 'good' from 'bad' approaches automatically; it will be predominantly a taxonomic device. A scheme of classification will provide a platform from which an evaluative exercise can be carried out.

In answering the evaluative questions with which this thesis is concerned, there is greater utility in adopting a scheme of classification that focuses upon the foundations for a particular approach, rather than upon its forms of expression. In the context of the fundamental objectives of legal fact finding, particular rules of evidence are not primarily important. They are merely the means by which underlying goals might be realised. In this sense, the fact that a particular approach to evidentiary treatment creates, for instance, a new exception to the opinion rule or to the rule against hearsay is not as important as the premises upon which the approach is based.

In contrast, the foundation of the methodology that a particular approach to evidentiary treatment implements has a much closer connection with the objectives of legal fact finding. This connection is especially apparent in light of the ideal that evidentiary treatment ought to facilitate rational truth identification. The foundations and underlying assumptions of an approach to evidentiary treatment should be demonstrably rational, so it makes sense to use a scheme for classification that focuses upon these features. The classification of approaches to the evidentiary treatment of computer-produced material that is adopted in this thesis is such a scheme.

#### 5.2.1.2 The three principal approaches

Under the scheme for classification that is adopted in this thesis, there are three principal approaches to the evidentiary treatment of computer-produced material

which can be discerned as significant in common law jurisdictions. As is described later in this chapter, there are also a number of variations or 'expressions' of each approach.

*The substantial equivalence approach*

The first approach, which is referred to here as the 'substantial equivalence' approach, regards computer-produced material as largely equal to other documentary material. Accordingly, it is said, computer-produced material should be treated from an evidentiary standpoint in the same way as its counterparts. The rationale of the substantial equivalence approach is reflected in the following material, which is taken from three different (and geographically disparate) common law commentators.

> There is no intrinsic reason why different régimes should apply to different forms of record-keeping, and every reason why they should not when the different forms are not readily distinguishable on their face. There may be no obvious difference in appearance between a document produced by the use of a computerised word-processing system and one produced by the use of a manual typewriter, nor is there the slightest justification for subjecting them to different hearsay rules.[3]

> Writing notes on a piece of paper is no different to storing them in a computer and, except insofar as they are modified by statute, the rules relating to the admissibility of these different data storage devices should logically be the same.[4]

> By and large, computer-generated evidence in the form of business records is "like any other" such evidence. Because the strictures applied to other paper apply so well to computer printouts, there is little reason to be concerned that rules of evidence are inadequate to deal with their authentication for admission into evidence.[5]

---

3 Tapper C, *Computer Law* (London: Longman, 4th ed, 1989) 395. See also *Rosenberg v Collins*, 624 F.2d 659, 665 (5th Cir, 1980): "computer data compilations ... should be treated as any other record of regularly conducted activity."

4 Brown R A, *Documentary Evidence in Australia* (Sydney: Law Book, 2nd ed, 1996) 355.

5 Kurzban S A, "Authentication of computer-generated evidence in the United States Federal Courts" (1995) 35 *IDEA - The Journal of Law and Technology* 437-459, 452, citing *Rosenberg v Collins* 624 F.2d 659, 665 (5th Cir, 1980).

The substantial equivalence approach can be summarized as embodying the view that within certain environments (chiefly those that involve record keeping) computer-produced material should not be seen as special. Therefore it requires no special treatment. In the words of the Australian Law Reform Commission, the argument is that "there is no basis for distinguishing between computer-produced business records and others."[6]

*The presumptive approach*

The second approach, which is referred to here as the 'presumptive' approach, regards computer-produced material as suitable for the application of a presumption about the reliability of the device that created it. The crux of this approach is the contention that any given computer ought to have the benefit of a presumption of reliability with respect to its operation. The foundation for the approach is the application of the common law maxim: *omnia praesumuntur rite et solemniter esse acta* (all acts are presumed to be done rightly and regularly).[7] It is said that this maxim creates a common law presumption that "[i]n the absence of evidence to the contrary, a mechanical instrument was operating correctly at the material time."[8]

*The specific computer approach*

The third approach, which is referred to here as the 'specific computer' approach, emphasises that computer-produced material will always be the product of a particular computer (as opposed to an instance of some wider class of material, or the product of some wider class of device).[9] This approach seeks to use information about the specific computer in question to furnish a basis for decisions about the probative value of that material. In all cases in which it is implemented, it is expressed in special

---

[6] Australian Law Reform Commission, Report 26, *Evidence (Interim)* (Canberra: Australian Government Publishing Service, 1985) Volume 1, paragraph 344.

[7] Broom H, *A selection of Legal Maxims: Classified and Illustrated* (London: Sweet & Maxwell, 8th ed, 1911) 737.

[8] Brown, Note 4 at 321, referring to the judgment of Stephen Brown LJ in *Castle v Cross* [1984] 1 WLR 1372 at 1377.

[9] More specifically, it will be the product of a particular combination of hardware and software.

statutory provisions. These are in similar terms in a number of jurisdictions.[10] The scheme of the provisions is to require foundational evidence in the form of certain assurances as to the proper operation of the computer in question and to provide that if such foundation evidence is given, statements contained in documents produced by the computer will be admissible.

In contrast to the terminology that is used here, the approach is sometimes referred to as 'computer specific'.[11] In fact, the statutory provisions and the approach that they express go somewhat further than is indicated by this terminology. The approach does not focus merely upon computers as a discrete subject matter or a discrete source of evidence. It focuses also upon the individual, or 'specific', computer that produced the material that is of interest in a particular case. This distinction is of considerable importance in light of the findings of chapter four. Considering the question of the reliability of a single computer is a vastly different undertaking from considering the question of the reliability of computers as a class of device. This difference is the basis for selecting the terminology 'specific computer' in preference to the terminology that has been used elsewhere.

5.2.1.3 Interaction with the rules of evidence

The extent to which an approach to evidentiary treatment interacts with the rules of evidence is, as has been observed, a secondary consideration in the evaluation of that approach. This notwithstanding, there are common issues that are presented by the potential for interaction between the rules of evidence and computer-produced

---

[10] Examples of current legislation include: *Evidence Act* 1977 (Qld), s 95; *Evidence Act* 1929 (SA), s 59B; *Evidence Act* 1958 (Vic), s 55B; *Computer Evidence Act* 57 of 1983 (South Africa), ss 2-3; *Evidence Ordinance (New Version)* 5731-1971 (Israel), s 36. Examples of now repealed legislation include: *Civil Evidence Act* 1968 (UK), s 5; *Police and Criminal Evidence Act* 1984 (UK), s 69. For a discussion of the Australian and United Kingdom provisions see generally Brown, Note 4 at 363-7. For a review of the Australian provisions see McNiff F V, "Computer Documentation as Evidence: An Overview of Australian Legislation Facilitating Admissibility" (1981) 1 *Journal of Law and Information Science* 45-60.

[11] See for example, Brown, Note 4 at 363.

material in a forensic setting. A brief account is given here of the key issues and of the problems that the potential for interaction creates.

This account provides some insight into some of the existing approaches to evidentiary treatment because, unlike the present thesis, those approaches appear to have been heavily influenced by a perceived need to 'fit' computer produced material into a pre-existing framework of orthodox evidence law. The account deals with the impact of three 'traditional' rules of evidence: the relevance rule, the rule against hearsay and the 'best evidence' rule. It also deals with the consequences of a concept of the 'weight' of evidence.

*The relevance rule*

The relevance rule is the primary rule of evidence in common law jurisdictions.[12] In order to be used in legal fact finding, material must be relevant. More specifically, it must be shown to be relevant. Almost invariably, the question of whether material is or is not relevant will be bound up with questions about the nature and origin of that material. The demonstration of these matters is frequently referred to as the 'authentication' of the material.[13] The relationship between the relevance rule and authentication was explained in these terms in *United States v Hernandez-Herrera* 952 F.2d 342 (10th Cir, 1991).

> The rationale for the authentication requirement is that the evidence is viewed as irrelevant unless the proponent of the evidence can show that the evidence is what its proponent claims.[14]

Because the majority of the rules of evidence are exclusionary in nature, the proponent of particular material is usually not concerned to authenticate material except to show that it is relevant. In some cases, however, the proponent will also

---

12 For a further discussion see chapter two, section 2.3.1.1.

13 For a discussion in the context of computer-produced material, see generally: Kurzban, Note 5.

14 952 F.2d 342, 343. For a further discussion of the rationale for the requirement of authentication and its relationship to the relevance rule see Australian Law Reform Commission, Note 6 at Volume 1, paragraphs 979-981.

have to demonstrate that an exception to an otherwise applicable exclusionary rule applies to the material. To do this, the proponent has to show that the material has additional characteristics that attract the operation of an exception to an exclusionary rule. It is possible, and convenient, to regard such requirements as aspects of the process of authentication. Exceptions to the hearsay rule provide good examples in this context. Some of these exceptions have a particular bearing on computer-produced material, because this material is frequently conceptualised in terms of its relationship to the rule against hearsay. This matter is considered further in connection with the evaluation of the substantial equivalence approach.

When the material is oral testimony, authenticating information can be provided by the witness giving the testimony and may be elicited in a manner that is sufficiently seamless that it can be said that the witness testimony is 'self authenticating'.[15] When the material is not oral testimony, the need to demonstrate relevance and the question of authentication take on a greater significance. This is because, as Tapper observes

> [a] document or thing cannot authenticate itself at common law, but must be introduced to the court by a human being whose task it is to explain its identity, its nature, its provenance and its relevance.[16]

In the case of computer-produced material, the issue is properly seen as a significant hurdle to admissibility.[17] A common question that must be addressed in this context is what kind of evidence will be regarded as sufficient to demonstrate the relevance of particular computer-produced material.

---

[15] The testimony will only actually be self-authenticating if the witness is able to demonstrate that the information that is to be imparted is in fact relevant.

[16] Tapper, Note 3 at 369. It may be that, strictly speaking, matters such as 'identity, nature and provenance' are foundations from which a court makes a determination or finding of relevance (and of any other matter that is to be the subject of authentication). If so, only the former matters will be the subject of actual proof by a party. In the United States, Fed R. Evid. 901(a) specifies what is to be proved in this context. It stipulates that "[t]he requirement of authentication or identification as a condition precedent to admissibility is satisfied by evidence sufficient to support a finding that the matter in question is what its proponent claims."

[17] See for example: Tapper, Note 3 at 369; Kerr O S, "Computer Records and the Federal Rules of Evidence" (2001) 49 *United States Attorney's Bulletin* 25-32 at 31; Australian Law Reform Commission, Note 6 at Volume 1, paragraphs 494 and 499.

The process by which computer-produced material is created and the components that are engaged by that process are both complex and obscure.[18] Contemporary application software in particular poses a greater problem. Due to its dependence upon the collective functionalities of operating systems and libraries of previously prepared program code, this software is the realisation of the design and development efforts of many different individuals.[19] Those individuals are most likely not to be associated with each other. This produces an obvious problem, since the focus of the authentication of material that is not oral testimony is the provision of "an [introduction of the material] to the court by a human being".[20] This point having been made, it is not asserted here that adherence to strict standards of authentication is necessarily required or even desirable[21] in the evidentiary treatment of computer-produced material. Authentication is merely one respect in which the interaction between computer-produced material and the rules of evidence is potentially problematic. Even though the full extent of the problem is not always recognised explicitly, it is a matter that has commanded attention in some approaches to the evidentiary treatment of computer-produced material, most notably the substantial equivalence approach.

*The rule against hearsay*

> An assertion other than one made by a person while giving oral evidence in the proceedings is inadmissible as evidence of any fact asserted.[22]

This is the rule against hearsay. It is to be expected that such a rule, prohibiting as it does particular uses of out of court assertions, should have a close bearing upon

---

18 As is demonstrated by the account that was provided in chapters three and four.

19 See in particular the discussion at chapter three, section 3.4 of the use of 'libraries' of program code and of the dependence of application programs upon functionalities that are implemented via operating system software.

20 Tapper, Note 3 at 369.

21 Although not necessarily accepted in this thesis without qualification, strong arguments against insistence upon the requirement that items of evidence be authenticated are canvassed in Australian Law Reform Commission, Note 6 at Volume 1, paragraphs 491-497.

22 Heydon J D, *Cross on Evidence: Sixth Australian Edition* (Sydney: Butterworths, 2000) paragraph [1260].

material that comes into existence outside the confines[23] of legal proceedings. The rule against hearsay is perceived to have a major impact upon the use of computer-produced material in legal fact finding. Tapper asserts that it is "by far the most important rule of the law of evidence in this context",[24] while Brown argues that "the main hurdles facing the common law admissibility of computer-produced evidence are the rules governing the admissibility of hearsay."[25]

Computer-produced material is, as was argued in chapter one, the result of a process of information transformation to which certain initial or input information has been supplied. It is possible that either the material itself, or the input information upon which it is based, or both, will amount to an out of court assertion that attracts the operation of the rule against hearsay. This possibility has been the subject of considerable attention in the relevant literature. The primary focus of interest has been the question whether in a particular case the rule applies at all.[26] This interest appears to turn upon a distinction between input information that is supplied by a human being and input information that is supplied by some automatic recording device or mechanism. Computer-produced material that results from human input is said to attract the operation of the rule against hearsay, whereas other material is not.[27]

Commentary in this area has explained the distinction in terms of the concept of 'real evidence'. Items of real evidence are

---

[23] As stated in chapter one, the thesis excludes consideration of so called 'demonstrative' evidence that involves computerised recreations or simulations that are displayed during the proceedings.

[24] Tapper, Note 3 at 377.

[25] Brown, Note 4 at 347. See also Tapper, Note 3 at 377; Kerr, Note 17 at 31.

[26] See for example the discussion in Brown, Note 4 at 352-355; Tapper, Note 3 at 373-375; Smith J C, "The Admissibility of Statements by Computer" [1981] *Criminal Law Review* 387-391 at 389-390; Laryea E T, "The Evidential Status of Electronic Data" (1999) *National Law Review* 3 at paragraphs 13-23.

[27] See Brown, Note 4 at 352, describing a computer operating in the latter mode as an 'independent data recorder'. Although it is said to have been wrongly decided (see Smith, Note 26), the case most cited in this context is *R v Pettigrew* (1980) 71 Cr App R 39. The mechanism in *Pettigrew* was a device that automatically recorded the serial numbers of bank notes.

> [t]hings [that] are an independent species of evidence, as their production calls upon the court to reach conclusions on the basis of its own perception, and not on that of witnesses directly or indirectly reported to it.[28]

It is said that computer-produced material is capable of reception as real evidence—thereby avoiding the operation of the rule against hearsay—because it is possible to attach significance to the particular form that the material takes. Tapper argues that

> [e]vidence derived from a computer constitutes real evidence when it is used circumstantially rather than testimonially, that is to say that the fact that it takes one form rather than another is what makes it relevant, rather than the truth of some assertion which it contains.[29]

The suggestion is that the fact finder's perception of facts *from* real evidence is in some way distinct from the assertion of facts *to* the fact finder by a testimonial account. Tapper notes that use of material in this way is only permissible when any statements or assertions that the material contains "originate in some purely mechanical function of a machine."[30]

This analysis has a slight appeal in the present context because it extends a degree of recognition to the fact that the process that a computer realises is one of information transformation, in which the processing of information is an important feature. Beyond this, the result of the analysis is only to make a distinction between a fact finder applying its own perception of a thing, and that fact finder being subject to a narrative account because of the words and symbols that may appear on the thing. However, when computer-produced material in the form of printed pages is involved, the strength of the distinction will be very slight indeed.

---

28 Heydon, Note 22 at paragraph [1270]. Compare Heydon, subsequently at [1330]: "anything other than testimony, admissible hearsay or a document, the contents of which are offered as testimonial evidence, examined by the tribunal as a means of proof" and Howard M N(ed), *Phipson on Evidence* (London: Sweet & Maxwell, 15th ed, 2000) at paragraph 1-07: "[m]aterial objects[,] other than documents, produced for inspection of the court ..."

29 Tapper, Note 3 at 373.

30 Tapper, Note 3 at 373 and at 374, citing the example of *The Statue of Liberty* [1968] 2 All ER 195, which concerned an automated system for recording radar images of vessels in the Thames estuary without any human input or intervention. For another discussion of that case, see Smith, Note 26 at 390. Smith uses the term 'direct' rather than real evidence, but the distinction from hearsay is the same as is drawn elsewhere.

The analysis is limited because it owes its existence to the belief that the key to the treatment of at least some kinds of computer-produced material is to try to establish that the hearsay rule does not apply to it. This amounts, however, only to an exercise in the application of orthodox evidence law. It involves no attempt to address the more important question of whether or not it or some other rule of evidence *ought* to apply, having regard to the objectives of legal fact finding.

In particular, the application of such a distinction (real vs. testimonial evidence) cannot be a substitute for analysis of whether the objectives of legal fact finding would best be served by modifying the application of the rule against hearsay. This is especially the case in the contemporary context in which there is a perceivable trend toward the substantial dismantling of the rule in a number of jurisdictions.[31] For this reason, no further attention is given here to the question of whether (as a matter of interpretation of existing evidence law) the rule against hearsay applies to computer-produced material.

Apart from these threshold concerns about the applicability of the rule, the relevant literature also reveals interest in the subsidiary question of whether computer-produced material attracts the operation of any of the exceptions to the rule against hearsay. The exception to the rule against hearsay that has received most attention in this context is the 'business records' exception.[32] The application of this exception to computer-produced material is premised upon the contention that when such material is produced in a business environment for use as a record, it may properly be considered to be the equivalent of a 'paper based' or manual counterpart. This thinking is at the heart of the 'substantial equivalence' approach. It is reviewed in connection with the evaluation of that approach in section 5.2.2.

---

31 See the discussion in chapter two, section 2.3.1.1.

32 See Tapper, Note 3 at 373, arguing that the business records exception is the one "most commonly relied upon to admit the evidence of records held on a computer."

*The 'best evidence' rule*

The best evidence rule has been said to restrict a party to the use of the "best evidence that the nature of the case [would] allow."[33] Tapper expresses the view that computer-produced material might attract the operation of the rule when it is based upon input information that is not itself established by original evidence.[34]

As is the case with questions about the applicability of the rule against hearsay, the problem to which the best evidence rule may give rise is not one that engages directly with the concerns of the present thesis. It too relates to the application of orthodox evidence law without regard to the objectives of legal fact finding. An additional consideration in this context is that the best evidence rule has very little contemporary significance.[35] For these reasons, no further attention is given here to the question of whether the use of particular computer-produced material that is the result of the provision of particular input information may infringe the best evidence rule.

*The weight of evidence*

Rules of admissibility determine what material will become evidence, but, as was discussed in chapter two, legal fact finding also involves the evaluation of that evidence. This evaluation is said to involve the 'weighing' of evidence in order to reach a final decision about the facts that are in dispute. It is sometimes suggested in this context that particular matters relate only to the 'weight' of evidence; they do not affect its admissibility.

More relevantly to the questions that are considered in this chapter, it has been asserted that issues about the reliability of a computer are relevant only to the weight of evidence, not its admissibility. This assertion appears originally to have been made by Smith.

---

33 *Omychund v Barker* (1745) 1 Atk 21, 49; 26 ER 15, 33.
34 Tapper, Note 3 at 372.
35 See chapter two, section 2.3.1.1.

> The computer differs from... other instruments only in that it can perform a variety of functions instead of only one. For that reason, it is necessary to have evidence... to establish the nature of the operations which the computer has been programmed to perform. It performs those operations just as mechanically as the thermometer or the camera. Of course the programmer may make a mistake but so may the person who for example, devises a scale on the thermometer. *This consideration goes to weight rather than admissibility.*[36] (Emphasis added.)

Brown embraces this distinction, albeit in rather a more tentative way. His citation of Smith's views is directly preceded by the (slightly inconsistent) suggestion that in cases in which computer-produced material was tendered as real evidence, there would be "evidence that the computer was operating correctly."[37] The presence of evidence as to the correct operation of the computer would, however, be the result of a requirement that such evidence be provided as a condition of admissibility. If correct operation and reliability were treated as matters relating to 'weight', then evidence tending to establish them would *not* be required to be presented as a condition of the admission of computer-produced material.

Smith's view plainly focuses upon matters that affect the reliability of computers. If it is accepted, then it follows that approaches to the evidentiary treatment of computer-produced material need not be concerned with questions of reliability in any material respect. So long as an approach facilitates the admission of the material, the question of reliability would be dealt with through the mechanism of allocation of weight. The suggestion that matters that relate to reliability ought to be addressed as questions of weight rather than admissibility does not, however, resolve the questions with which this thesis is concerned. As is true of views that relate to the applicability of particular rules of admissibility, this view is one about the scope and present effect of orthodox evidence law. The expression of it makes no attempt to apply any appreciation of the theoretical objectives of legal fact finding to the question of what the content of the law of evidence in this area *ought* to be. Added to this, it is flawed in two significant respects.

---

36 Smith, Note 26 at 390.
37 Brown, Note 4 at 354.

First, its application to computer-produced material fails to engage the proper balance between rules of admissibility and the mechanism of allocation of weight to evidence. As was demonstrated in chapter two, the role of rules of admissibility is to provide an indication that particular material has some capacity to aid the identification of the truth.[38] Rules of admissibility need not produce an assurance that given material is accurate, but they must reflect the underlying principle that legal fact finding is not ambivalent about finding the truth. Unless their application provides some indication about the probative value of the material, then they have no purpose at all in the context of truth identification.

It is true in this respect that rules of admissibility may happen to have a less significant role in some cases than in others. For material in respect of which an assumption of likely probity cannot be made, the rules of admissibility have an important role to play in attempting to provide a level of 'quality assurance'. Is it then appropriate to exclude from the operation of these rules the resolution of questions about the reliability of a computer?

It is clear enough that the reliability of a computer influences the accuracy of its output. Yet, as was established in chapter four, no assumption can be made that the reliability of computers generally meets a high threshold. All that can be said is that in the absence of specific information to the contrary, the reliability of a given computer must be taken to be uncertain in any given case. It is precisely this uncertainty that enlivens a need for rules of admissibility to address the issue of reliability in some way. If this need is not met, then material that is admitted for use in legal fact finding will carry no indication whatsoever that it has any capacity to aid the identification of truth.

The second flaw in Smith's view is that there is no basis for supposing that the process of allocation of weight is equipped to assess the reliability of a computer. In

---

[38] Although some rules may have a purpose in promoting other goals of legal fact finding by, for instance, excluding material that is subject to a recognised privilege or pubic interest immunity.

cases in which there is no obvious error on the face of particular computer output, the only opportunity for doing so is indirect. It will involve the assessment of the accuracy of that output only by comparing it with other evidence. In such cases, there will be no opportunity to attempt to quantify reliability by reference to the matters which were shown by chapter four to affect it, such as fault volumes, sensitivity to particular inputs, or the patterns that may emerge from the review of prior failure data.

It is true that the ultimate question for the fact finder relates to truth and, to this extent it is only the accuracy or inaccuracy of the relevant output that needs to be considered. This viewpoint is, however, problematic because the only possibility for assessing accuracy in the absence of information about reliability is by reference to other items in the pool of evidence that is before the fact finder. It cannot be assumed, however, that the pool of evidence will otherwise contain a sufficient quantity of accurate material to enable given computer output that is inaccurate to be recognised for its shortcomings. More generally, the argument that possible deficiencies in one kind of evidence can be remedied by comparison with other evidence is flawed because it is a circular one. It presupposes that the 'other' evidence will have no such deficiencies of its own.[39]

A further possibility is that in some cases there will be evidence from two different computers which is contradictory. If both pieces of output have been admitted without any attempt to address the issue of reliability, then the comparison of one against the other logically produces a deadlock which in turn reveals the circularity of argument that such an approach produces. The situation is made no better by the consideration that computer output might be weighed against non-computer output. In some cases, the only information that might be available about a given issue will be contained in material produced by a computer.

For these reasons, it is not appropriate that an approach to evidentiary treatment deal with the issue of reliability only via the mechanism of allocation of weight. More particularly, it is not appropriate that any such approach 'abdicate' the task of

---

[39] See also in this regard the discussion in chapter two, section 2.3.

attempting to deal with the issue of reliability from the mechanism of admissibility to the mechanism of weight.

### 5.2.2 The 'substantial equivalence' approach

#### 5.2.2.1 The predominant expressions of the approach: business records

At the heart of the substantial equivalence approach is the proposition that computer-produced material and other kinds of material are substantially the same. Any expression of the approach requires a manual analogue with which computer-produced material can be identified. Business records produced by manual means are one such analogue. Such records, and the rules of evidence that deal with them, are the focus of the predominant expressions of the substantial equivalence approach. These expressions appear in a number of jurisdictions and a selection of them are considered in the following sections.

##### 5.2.2.1.1 United States: Federal Rules of Evidence

*Background*

Decisions of United States Courts of Appeals[40] that deal with the application of Fed. R. Evid. 803(6) to computer-produced material comprise a significant expression of the substantial equivalence approach in a business records context. The rationales that are adopted in these decisions are important not only because they dominate the United States context. They also represent a template from which s 147 of the *Evidence Act* 1995 (Cth) has been produced.

---

[40] That is, the Courts constituted by 28 U.S.C. §41.

Fed. R. Evid. 803(6) enacts a 'business records'[41] exception to the rule against hearsay for those jurisdictions that apply the Federal Rules of Evidence. It provides, in part, as follows.

> **Records of regularly conducted activity**. A memorandum, report, record, or data compilation, in any form, of acts, events, conditions, opinions, or diagnoses, made at or near the time by, or from information transmitted by, a person with knowledge, if kept in the course of a regularly conducted business activity, and if it was the regular practice of that business activity to make the memorandum, report, record or data compilation ... unless the source of information or the method or circumstances of preparation indicate lack of trustworthiness.
> ...

The exception is important in the present context because of the obvious contemporary connection between business undertakings and the use of computers.[42] This importance is amplified by the fact that the definition of 'business' for the purposes of the various statutory formulations of the exception is very broad. Rule 803(6) concludes with the following definition of the term.

> The term "business" as used in this paragraph includes business, institution, association, profession, occupation, and calling of every kind, whether or not conducted for profit.

Decisions that apply Fed. R. Evid. 803(6) comprise the bulk of the contemporary judicial application of what is referred to in this thesis as the substantial equivalence approach.[43] These decisions evince an enthusiasm to apply to computer-produced material the same rules that were applied to other documentary records. In fact this enthusiasm predates the Federal Rules of Evidence, as the views of a number of state

---

41 For examples of the use of this terminology see Kwestel S, "The Business Records Exception to the Hearsay Rule–New is Not Necessarily Better" (1999) 64 *Missouri Law Review* 595-660; Kerr, Note 17 at 25, referring to Fed. R. Evid. 803(6) as the "business records exception"; McNiff, Note 10; New South Wales Law Reform Commission, *Report on Evidence (Business Records)* (Sydney: NSW Government Printer, 1973) at paragraphs 22-25, which appear under the heading: "The need for a business records exception". Compare Wigmore J H, *Evidence in Trials at Common Law* (Chadbourn revision) (Boston: Little Brown and Company, 1974) §1517-§1561b, referring to a "regular entries" exception and the introductory words to Fed. R. Evid. 803(6) itself: "Records of regularly conducted activity."

42 Tapper also observes that the application of rules concerning the admission of business records to computer-produced material has been the subject of comparatively more judicial attention in the United States. He attributes this to, among other things, the early adoption of computers in business settings in the United States: Tapper, Note 3 at 380.

courts illustrate. In *Transport Indemnity Company v Seib* 132 N.W.2d 871 (Neb, 1965) the Supreme Court of Nebraska held that under its rules of evidence relating to business records

> [n]o particular mode or form of record is required. The statute was intended to bring the realities of business and professional practice into the courtroom and the statute should not be interpreted narrowly to destroy its obvious usefulness.[44]

In *King v State ex rel Murdock Acceptance Corporation* 222 So.2d 393 (Miss, 1969) the Supreme Court of Mississippi held that

> [t]he rules of evidence governing the admission of business records are of common law origin and have evolved case by case, and the Court should apply these rules consistent with the realities of modern business methods. The law always seeks the best evidence and adjusts its rules to accommodate itself to the needs of the age it serves.[45]
>
> ...
>
> We are not to be understood as indicating that computer evidence is infallible. *Its probative value is the same as conventional books*, and it is subject to refutation to the same extent.[46] (Emphasis added.)

In a later decision that was also decided under state rules of evidence, *Louisiana v Hodgeson* 305 So.2d 421 (La, 1974), the Supreme Court of Louisiana said that

> [i]n affirming the admissibility of computer records and print-outs, [it was] in accord with the jurisdictions which have confronted the problem of adjusting the tested rule of evidence to the realities of contemporary business life while insuring the reliability of the information offered at trial.[47]

---

43 Kurzban, Note 5 at 441; Tapper, Note 3 at 386.
44 132 N.W.2d 871, 875.
45 222 So.2d 393, 397.
46 222 So.2d 393, 399.
47 305 So.2d 421, 428.

An initial concern that arises out of the application of this attitude is that the Federal Rules of Evidence also provide for a requirement of 'authentication' in the sense referred to in section 5.2.1.3. It is described in Fed. R. Evid. 901(a) as

> a condition precedent to admissibility [that] is satisfied by evidence sufficient to support a finding that the matter in question is what its proponent claims.

This requirement plainly creates a need to consider and to address the question of reliability. Computer-produced material is the product of a process of information transformation. It is both the input information and the way in which it is processed that will determine the composition, characteristics and identity of this material. To demonstrate these things in conformity with a requirement of authentication necessitates two things: a specification of the expected manner of processing and some proof of the likelihood that this expectation will be realised. The authentication of computer-produced material therefore requires information that qualifies the reliability of the relevant computer system directly, or information from which its reliability can rationally be inferred.

Despite this, decisions under the Federal Rules of Evidence generally do not emphasise the issue of reliability as a matter that is critical to authentication. This is an incident of the fact that computer-produced material has been viewed in these decisions as a near literal equivalent of the product of manual processes.[48] Kerr describes the position under the Federal Rules in the following way.

> The standard for authenticating computer records is the same as for authenticating other records. The degree of authentication does not vary simply because a record happens to be (or has been at one point) in electronic form.[49]

---

[48] For a manual process there is no transformation analogue that produces an issue of reliability. So, for example, recording information in hand written notes involves only the commitment of input information to paper. The retrieval of that information involves reading what has been written. At no stage does any transformation of information take place within the confines of the 'pen and paper' mechanism.

[49] Kerr, Note 17 at 26.

It is in this respect that the manner in which the Federal Rules have been interpreted most clearly embodies a substantial equivalence approach to evidentiary treatment. This manner of interpretation has advanced in some instances almost to the point of disregarding the functioning of the computer entirely. The position adopted in *United States v Croft* 750 F.2d 1354 (7th Cir, 1985) exemplifies this. The *Croft* Court held, speaking of material that had been produced by a computerised payroll system, that

> the relevant payroll evidence was simply transferred from payroll data sheets to a computer disk for convenient storage in the computer and easy retrieval on computer printouts. ... the actual computer program was *of little if any importance in the present case.*[50] (Emphasis added.)

In *United States v Young Bros, Inc* 728 F.2d 682 (5th Cir, 1984) the Court was faced with an express argument by the appellant that

> computer-generated records are less reliable than other kinds of business records because they depend upon the accuracy of the software as well as the person feeding the raw data into the computer.[51]

Without ruling upon the validity of the argument, the Court was nevertheless prepared to hold that the admission of the material was not undermined by the absence of foundation evidence as to the programming of the computer. The argument of the appellant in *Young Bros* raises, however, precisely the issue of reliability that must be addressed in the context of the evidentiary treatment of computer-produced material.

*Reliability and the 'lack of trustworthiness' limitation under Fed. R. Evid. 803(6)*

The need to consider an issue of reliability in connection with computer-produced material that is offered for admission under Fed. R. Evid. 803(6) is not limited to the requirement of authentication. A second and distinct basis for considering reliability arises because of the words of limitation that appear in Fed. R. Evid. 803(6). Those words have the effect that the exception to the rule against hearsay that is offered by Fed. R. Evid. 803(6) will not apply when "the source of information or the *method or circumstances of preparation* indicate lack of trustworthiness." (Emphasis added.)

---

50 750 F.2d 1354, 1365.
51 728 F.2d 682, 693.

This requirement has, however, been treated as indistinct from the separate issue of reliability that arises under Fed. R. Evid. 901(a). What has been argued in this connection is that addressing a question of trustworthiness for the purposes of one rule satisfies, or ought to satisfy, any need to address the question for the purposes of the other rule.[52] Kerr notes in this respect a "conceptual overlap between establishing the authenticity of a computer-generated record and establishing [its] trustworthiness ... for the business record exception to the hearsay rule."[53]

Whether analyses of this kind are correct is not a matter that needs to be resolved here. They encompass only questions about the interpretation of existing rules of evidence. They make little or no reference to the underlying goals of legal fact finding and, as such, their correctness is not directly material for present purposes. The question that is of interest here is not whether a particular rule of evidence or procedure can fairly be interpreted in a particular way. Rather it is whether a rule, combined with the way in which it is interpreted, takes proper account of the need to address the issue of reliability when dealing with computer-produced material. Viewed in the light of the goals of legal fact finding, the issue of reliability arises irrespective of the content of particular, existing rules of evidence and irrespective of the way in which they might be interpreted.

*Reliability and the 'use and reliance' criterion*

It happens to be the case that an issue of reliability has been recognised as arising under Fed. R. Evid. 803(6) and/or 901(a), albeit in a less than explicit way. The relevant question is whether, having been recognised, the issue is addressed in a way that can be said to have a rational foundation. The answer is that it is not. Rather, the issue is addressed by the adoption of an 'indicator' of reliability that lacks a rational foundation.

---

52 For a detailed discussion see Peritz RJ, "Computer Data and Reliability: A Call For Authentication of Business Records Under the Federal Rules of Evidence" (1986) 80 *Northwestern University Law Review* 956-1002, 984-986.

53 Kerr, Note 17 at 28.

The indicator that is adopted is no more than the fact that a computer system has been used and relied upon in a business environment.[54] In *United States v Moore* 923 F.2d 910 (1st Cir, 1991) it was held that "the ordinary business circumstances described [in that case] suggest[ed] trustworthiness ... at least where absolutely nothing in the record in any way implies the lack thereof."[55] In *Rosenberg v Collins*, 624 F.2d 659 (5th Cir, 1980) the Court relied upon the fact that the records "were sufficiently trustworthy in the eyes of [a] disinterested company to be relied upon by [that] company in conducting its day to day business affairs" as evidence of trustworthiness in the rule 803(6) sense.[56]

Kurzban explains the criterion of 'use and reliance' in terms of substantial equivalence.

> Records are admitted into evidence because of the trust that disinterested businesses place in them *and that trust is independent of the record's medium*, be it paper or computer-storage device.[57] (Emphasis added.)

This criterion plainly depends upon the proposition that an entity that conducts a business will be motivated to keep accurate records and that, consequently, it will only use computer systems that are sufficiently reliable for this purpose. This proposition coincides with the rationale for the original 'business records' exception to the rule against hearsay. That rationale has been explained in the following terms.

> [S]ince businesses must keep reasonably accurate records if they are to stay in business, those records are likely to be sources of sufficiently accurate information to be acceptable as evidence.[58]

---

54 Kurzban, Note 5 at 445. For a similar view see: Australian Law Reform Commission, Note 6 at Volume 1, paragraph 989.

55 923 F.2d 910, 915. This passage is referred to in Kerr, Note 17 at 28 and for a further discussion see: Kurzban, Note 5 at 445; Murray D R and Chorvot T J, "Stepping up to the Next Level: From the UETA to the URE and beyond" (2001) 37 *Idaho Law Review* 415-439 at n7. See also in this context *United States v Hayes*, 861 F.2d 1225, 1228-1229 (10th Cir, 1988), which was the principal authority relied upon by the *Moore* Court for its ruling.

56 624 F.2d 659, 665.

57 Kurzban, Note 5 at 459.

58 Brown, Note 4 at 221, and see related comments at 367. See also *Louisville and Nashville Railroad Co. v Knox Homes Corp.* 343 F.2d 887, 896 (5th Cir, 1965): "[i]t is the business record in the form regularly kept by the particular business and reliance thereon that gives the trustworthiness and hence legal admissibility to such records." For a more detailed discussion, see Wigmore, Note 41 at §1522.

Hope JA of the New South Wales Court of Appeal expressed a similar sentiment in a frequently cited passage from his Honour's judgment in *Albrighton v Royal Prince Alfred Hospital* [1980] 2 NSWLR 542.

> Any significant organisation in our society must depend for its efficient carrying on upon proper records made by persons who have no interest other than to record as accurately as possible matters relating to the business with which they are concerned ... When what is recorded is the activity of a business in relation to a particular person amongst thousands of persons, the records are likely to be a far more reliable source of truth than memory. They are often the only source of truth.[59]

The importance of the rationale is underscored by the significance of the outcome that the business record exception delivers. Material is "[rendered] admissible whether or not it can in fact be shown to be accurate."[60] The importance of this result requires that the rationale be revisited when, as in the case of computer-produced material, the exception is to be applied to a new class of material. This revisitation is undertaken in section 5.2.2.2.

5.2.2.1.2 Australia: The 'Uniform' Evidence Acts

A statutory version of the use and reliance criterion has been enacted as s 147 of the *Evidence Act* 1995 (Cth). The section relevantly provides as follows.

> **Documents produced by processes, machines and other devices in the course of business**
>
> (1) This section applies to a document:
>
> (a) that is produced wholly or partly by a device or process; and
> (b) that is tendered by a party who asserts that, in producing the document, the device or process has produced a particular outcome.
>
> (2) If:

---

59 [1980] 2 NSWLR 542 at 549. See also the discussion in Brown, Note 4 at 221.
60 Brown, Note 4 at 221.

(a) the document is, or was at the time it was produced, part of the records of, or kept for the purposes of, a business (whether or not the business is still in existence); and

(b) the device or process is or was at that time used for the purposes of the business;

it is presumed (unless evidence sufficient to raise doubt about the presumption is adduced) that, in producing the document on the occasion in question, the device or process produced that outcome

When it is combined with section 69 of the Act, which is an orthodox and 'medium neutral' formulation of the business records exception, section 147 can be used as an aid to the authentication of business records[61] that have been produced by a computerised process.[62]

Section 147 generates two issues that are relevant for present purposes. First, unlike the position under the Federal rules, use of the relevant device or process attracts only a presumption, which may be rebutted. The section therefore possesses a hybrid character. To the extent to which it is an expression of a 'presumptive' approach to evidentiary treatment, it is examined in section 5.2.3.

Second, the provision makes no explicit reference to a need to demonstrate reliance upon the device or process, as opposed to its mere use for the purposes of the business. This notwithstanding, it seems clear that the provision was meant to reflect an acceptance of the use and reliance criterion. This is confirmed in the report of the Australian Law Reform Commission in which enactment of the provision was recommended. The Commission argued that "[i]t is appropriate that it be assumed in the absence of contrary evidence that the devices relied upon in a business are

---

61 The Act provides a very expansive definition of the term 'business'. In addition to ordinary commercial activities, many other public and governmental activities are also deemed to be businesses for the purposes of the Act, see *Evidence Act* 1995 (Cth), Dictionary, part 2, clause 1(1).

62 Although is not stipulated in the Act, the reference in the section to 'devices' and 'processes' plainly encompasses computers: Brown, Note 4 at 320. This view is reinforced by the conceptualisation of the computer as realising a *process* of information transformation in the sense introduced in chapter one. By virtue of s 147(3), the section is not available in respect of documents produced: "for the purpose of conducting, or for or in contemplation of or in connection with, an Australian or overseas proceeding" or "in connection with an investigation relating or leading to a criminal proceeding."

trustworthy."[63] It remains to be seen whether an implied condition will be read into subsection 2(b) to the effect that the business also be shown to have relied upon the device or process.[64]

5.2.2.1.3 Australia: other jurisdictions

The regime that existed in New South Wales prior to the coming into force of the *Evidence Act* 1995 (NSW) also implemented a substantial equivalence approach.[65] This approach was copied with minor amendments in the other Australian jurisdictions.[66] Under Part IIC of the *Evidence Act* 1898 (NSW), which dealt with the admissibility of business records generally, no distinction was made between business records produced by a computer and other business records. That this result was intended is plain from the report of the New South Wales Law Reform Commission that recommended the enactment of Part IIC.

> We were led ... to consider the admissibility of statements in business records, whether the records are kept or produced by computers or by other reliable means.
>
> ...
>
> In the result, we recommend that the Evidence Act, 1898, be amended to provide a statutory exception to the rule against hearsay evidence: an exception which will facilitate the admission in legal proceedings of reliable statements in business records, *however kept or produced* ...[67] (Emphasis added.)

---

63 Australian Law Reform Commission, Note 6 at Volume 1, paragraph 989.

64 The matter was not addressed in the only decision that has considered the provision thus far, see: *R v Dudko* [2002] NSWCCA 336 at 53.

65 The regime was implemented by Part IIC of the *Evidence Act* 1898 (NSW). This Act was repealed in its entirety by s 3 of the *Evidence (Consequential and Other Provisions) Act* 1995 (NSW). The provisions of Part IIC of the 1898 Act arose out of the recommendations of the New South Wales Law Reform Commission's 1973 report: New South Wales Law Reform Commission, Note 41.

66 For a discussion see Brown Note 4 at 222. As Brown observes, the provisions were copied with minor changes in the Northern Territory and Tasmania and with more substantial changes in Western Australia. As noted in chapter one, Tasmania has now adopted the *Evidence Act* 1995 (Cth), as the *Evidence Act* 2001 (Tas). The Tasmanian provisions to which Brown refers have now been repealed by the s 199 of the 2001 Act.

67 New South Wales Law Reform Commission, Note 41 at paragraphs 4-5.

The provisions that referred specifically to computers or comparable devices (namely ss 14CE(6)(b), 14CD(1) and 14CN(1)(c)) all served to ensure that computer produced records would be treated in the same manner as other business records. The provisions were weaker than even the 'use' stipulation that appears in s 147(2)(b) of the *Evidence Act* 1995 (Cth). They merely required that the information that was to be admitted comprised a statement of a fact that was contained in a document[68] that formed part of the record of a business[69] and that the statement had "been made in the course of or for the purposes of the [that] business."[70]

5.2.2.1.4 United Kingdom (Civil Proceedings)

Like the position that obtained under Part IIC of the *Evidence Act*, 1898 (NSW), the United Kingdom now applies in civil proceedings a substantial equivalence approach without the constraint of a requirement of use or reliance. The legislative expression of the approach is implemented by the combined effect of ss 8, 9(1) and 13 of the *Civil Evidence Act* 1995 (UK), which provide as follows.

> 8. (1) Where a statement contained in a document is admissible as evidence in civil proceedings, it may be proved—
>
> (a) by the production of that document, or
>
> (b) whether or not that document is still in existence, by the production of a copy of that document or of the material part of it, authenticated in such manner as the court may approve.
>
> (2) It is immaterial for this purpose how many removes there are between a copy and the original.

---

68 *Evidence Act* 1898 (NSW), s 14CE(1).

69 *Evidence Act* 1898 (NSW), s 14CE(4).

70 *Evidence Act* 1898 (NSW), s 14CE(5).

9. (1) A document which is shown to form part of the records of a business or public authority may be received in evidence in civil proceedings without further proof.

13.

...

"document" means anything in which information of any description is recorded, and "copy", in relation to a document, means anything onto which information recorded in the document has been copied, by whatever means and whether directly or indirectly.

The enactment of these provisions coincided with the repeal of Part 1 of the *Civil Evidence Act* 1968 (UK), which had implemented a specific computer approach.[71] In the report in which it proposed the change of approach to evidentiary treatment in civil matters, the Law Commission said that it had "decided to recommend that no special provisions be made in respect of computerised records."[72] This is plainly the effect of the provisions. The key to admission of computer-produced material under the provisions is the very wide definition of 'document'. The Commission observed that this definition "will cover documents in any form and in particular will be wide enough to cover computer-generated information."[73]

5.2.2.2 Evaluation

What is being evaluated here is how approaches to evidentiary treatment address the issue of the reliability of computers. The focus of evaluation is how (and how well) an approach to evidentiary treatment facilitates the realisation of the goal of rational truth identification. At the broadest level, it is clear that the substantial equivalence approach depends upon a crucial assumption. The assumption is that the way in which a computer processes information will be such that it will not contribute to the possibility of error in output. This much is clear from the view that such output is not

---

71 See schedule 2 of the *Civil Evidence Act* 1995 (UK).

72 Law Commission, *The Hearsay Rule in Civil Proceedings* (London: H.M.S.O., 1993) paragraph 4.43.

73 Law Commission, Note 72 at paragraph 4.43.

materially different from other material, such as hand written records, for which no 'manner of processing' analogy is applicable. It is patently an assumption about the reliability of computers. More importantly it is an assumption that in general the reliability of computers will always be perfect or, at worst, that such reliability will be such that the risk of failure is nominal. Even when viewed conservatively, this amounts to an assumption that the reliability of computers will generally meet or exceed a high threshold in the sense discussed in chapter four.

Such an assumption is, however, quite problematic. The findings of chapter four, which were to the effect that such a condition could not be demonstrated with information presently available, make this assumption untenable. It is true that the Federal Rules expression of the substantial equivalence approach seeks to make use of environmental conditions—namely the fact of 'use and reliance'—as a basis for inferring reliability. Close scrutiny of this measure reveals, however, that it still entails an underlying need to resort to an assumption about high general levels of reliability. Elsewhere, the omission to impose even this measure necessitates a direct recourse to the assumption. This recourse is made explicit in the commentary that deals with the regimes that apply in the United Kingdom and Australia.

The problems associated with the 'use and reliance' criterion are considered in section 5.2.2.2.1, whilst the problems associated with recourse to a direct assumption about reliability are considered in section 5.2.2.2.2.

5.2.2.2.1 'Reliance', but not reliability

In section 5.2.2.1.1 it was observed that the 'use and reliance' criterion coincides with the rationale for the original 'business records' exception to the rule against hearsay. The attempt to 'borrow' the rationale of use and reliance from an older setting and apply it to computers[74] is not of itself objectionable, since the rationale involves a

---

[74] This was how the Supreme Court of Mississippi characterised the use of the business records exception, holding that such use did "not [depart] from the shop book rule, but only extend[ed] its

sound premise. The problem lies rather in the extent to which the rationale survives the attempt to apply it in a computerised context. Three issues arise in this regard. They relate to the tendency for accuracy that regular recording is said to engender, the ease of detecting inaccuracies in regularly kept records and the significance of the motive for accuracy.

*Tendency for accuracy*

The original rationale depends upon the premise that the business, and more specifically its employees, will have the *means* with which to keep accurate records and not merely the motive to do so. Assuming basic literacy, there is no reason to suppose that an employee could not make and maintain accurate records within a manual system. The record maker has, after all, near complete control over such a system. The critical factor is, as has been discussed, accuracy at the recording stage and it is only this factor that the original rationale addresses. Wigmore's discussion of the rationale, for instance, focuses only on this question.

> The habit and system of making such a record with regularity calls for accuracy through the interest and purpose of the entrant; and the influence of habit may be relied on, by very inertia, to prevent casual inaccuracies and to counteract the possible temptation to misstatements.[75]

McCormick was also concerned with accuracy at the recording stage. He too believed that habit played an important role in this respect, arguing that "the very regularity and continuity of the records is calculated to train the record-keeper in habits of precision."[76] The problem in the computerised setting is that these considerations disregard entirely the question of computer reliability. The rationale has no application to this question, however well it may provide a basis for confidence in the accuracy of input information. Application of the rationale to computers is viable

---

application to electronic record keeping": *King v State ex rel Murdock Acceptance Corporation* 222 So. 2d 393, 398 (Miss, 1969).
electronic record keeping.

[75] Wigmore, Note 41 at §1522.

[76] Cleary W (ed) McCormick's Handbook of the Law of Evidence (St Paul, Minnesota: West, 2nd ed, 1972) §306.

only if it is believed that a business entity will immediately cease using any computer that appears to it to be less than perfectly reliable.

This condition is, however, unrealistic since some proportion of errors or inaccuracies will be non-obvious and therefore not immediately detected. There is also the reality that it is difficult and uneconomical to discard and replace sophisticated—and presumably expensive—computer systems with any degree of frequency. Rather, it is at least possible that some organisations will tolerate a degree of imperfection for the sake of economy. The resulting scenario is altogether different from one in which an entity that conducts a business has complete control over a manual process of record keeping.

*Ease of error-detection*

The second issue that arises in the context of an attempt to apply the original rationale for the business records exception relates to the ease with which errors can be detected. The original rationale appears to have contemplated that errors in entries in a regular and systematic account of business transactions would be obvious. McCormick, for instance, suggested that "such books and records are customarily checked as to correctness by systematic balance-striking".[77] Moreover in a 'manual' setting

> misstatements cannot safely be made, if at all, except by a systematic and comprehensive plan of falsification. As a rule, this fact (if no motive of honesty obtained) would deter all but the most daring and unscrupulous from attempting the task.[78]

The difficulty here is that material that is produced by a computer is quite unlike the bound 'shop' or account books from which the exception arose.[79] The most material difference is that computer output can comprise loose sheets of paper, or even discrete images or sounds. Such things are no more than extracts which may present

---

77 Cleary, Note 76 at §306.
78 Wigmore, Note 41 at §1522.
79 For discussions see Wigmore, Note 41 at §1522; Brown, Note 4 at 326.

information in sequences and configurations that bear no resemblance to the state in which information was originally provided as input. What is presented may be a synthesis of original information and information that has been derived from it by some series of calculations or manipulations. As the discussion in chapter three reveals, the way in which a computer operates means that all output is created through the manipulation of input. This will be true even when the computer is used only to 'store and retrieve' information. These possibilities render the proposition that inaccuracies will necessarily be easy to detect, and hard to conceal, inapposite in the computerised setting.

*A motive for accuracy*

The third issue in this context involves the contention that an additional source of motivation to make accurate records is the risk to an employee of reproach if inaccuracies are detected. The New South Wales Law Reform Commission argues in this context that

> [t]hose who make, or supply information for, statements in business records have some corresponding motives for telling the truth. They have a duty to their employer to do so and, in general, no motive to misrepresent or conceal the truth. They fear the adverse consequences of inaccurate reporting being discovered by their superiors. They know that lying may result in dismissal.[80]

This argument is similar to the argument that systematic record keeping engenders a tendency for accuracy. Both focus upon the question of motive in isolation from the question of capacity or means. Yet it is not enough to consider only what a notional employee who is charged with record keeping will be motivated to do. In a computerised setting, the question of motive affects only the input information. It can have no impact upon the operation of the computer system and in particular it can have no impact upon its reliability. Employees who are engaged only to provide input information to computer systems cannot overcome, or compensate for, less than

---

[80] New South Wales Law Reform Commission, Note 41 at paragraph 18. See also Wigmore, Note 41 at §1522: "[i]f, in addition to this, the entrant makes the record under a duty to an employer or other superior, there is the additional risk of censure and disgrace from the superior, in the case of inaccuracies".

perfect reliability in those systems, no matter how strongly motivated they may be to do so.

This point further highlights the importance of the fact that computers realise processes of information transformation. To identify motives for accuracy that can only ever affect the integrity of input information is to offer no assurance about the accuracy of the output that the computer may produce. It is for this reason that a 'use and reliance' criterion does not demonstrate nor justify an inference of computer reliability.

The criticisms that have been made in connection with the expression of the substantial equivalence approach in the United States apply also to section 147 of the *Evidence Act* 1995 (Cth). They would apply even if an implied condition of reliance upon the device or process in question were read into the section. In the absence of such a condition, the provision would appear to depend only upon the bare fact of 'use'. In this scenario, the criticisms that have been made here are all the more compelling; the provision would provide no justification at all for the assumption that it necessarily makes about the reliability of the 'device or process' in question.

5.2.2.2.2 A need to make an assumption about reliability

As was demonstrated in section 5.2.2.2.1, the original rationale for the business records exception cannot be applied to the computerised setting to justify the 'use and reliance' criterion that was developed in the United States and which is reflected to a degree in the Australian legislation. The central impediment is the lack of control over the process that is adopted to maintain the records of the business.

Because the 'use and reliance' criterion alone cannot demonstrate reliability, an assumption about reliability must also be made. The necessary assumption is that even if computers do not exhibit perfect reliability, they nevertheless exhibit sufficiently high levels of reliability that their operation (as opposed to inaccuracies in

input) does not contribute significantly to inaccuracies in output. Under such an assumption, shortcomings in reliability are not a significant issue for the accuracy of computer output. The primary issue is instead the accuracy of input.

This position has been most explicitly articulated in the non-American jurisdictions. It is most frequently to be found in the reports of law reform bodies that have considered the question of 'computer output as evidence'. This may be because the non-American jurisdictions have had neither the rule 803(6) jurisprudence or a clear mechanism for its local transplantation as a reference point. In the absence of that influence, a more explicit engagement with the underlying issue of reliability was possible, and necessary. For this, no more satisfactory an outcome has been produced.

The recommendations of the comparatively early (1973) review of this subject by the New South Wales Law Reform Commission are an example of the way in which the position has been expressed. In its *Report on Evidence (Business Records)*,[81] the Commission argued that

> [t]here is no doubt that business records can be made and kept by a computer to a degree of accuracy which cannot, as a practical matter, be attained by a corresponding clerical system. Commonly they are so made, and kept. But errors in such records do occur. The cause of error is rarely a malfunctioning of the computer equipment.[82]

And that

> Generally speaking it may be said that with a well-prepared program the possibility of undetected error being caused by technical defects in the processing of information is very remote indeed. Apart from this, the reliability of the components now used in the electronic circuitry of computers and modern techniques of regular preventive maintenance make technical error a rare occurrence.[83]

---

81 New South Wales Law Reform Commission, Note 41.

82 New South Wales Law Reform Commission, Note 41 at paragraph 39.

83 New South Wales Law Reform Commission, Note 41 at Appendix D, paragraph 50. The Law Reform Commission of Western Australia also considered the subject of the admissibility of computer output and again adopted a similar view. It suggested, rather more succinctly but no less significantly, that "a computer itself does not normally make mistakes": Law Reform Commission of Western Australia, *Report on the Admissibility in Evidence of Computer Records and Other Documentary Statements* (Perth: The Commission, 1978) 57.

In formulating recommendations for the evidentiary treatment of computer-produced material under New South Wales law, the Commission was plainly concerned with the issue of the accuracy of input information. This was to the near exclusion of concern about what happened to that information within the computer systems of the record keeper. After asserting that "[s]tatements in business records produced by computers should be reliable if the information from which the statement was derived is reliable",[84] the Commission suggested that

> conditions should be imposed to safeguard the reliability of the source material ... [but] it [was] essential to relieve businesses using computers from the burden of proving strictly in legal proceedings the various steds [sic – steps] involved in the keeping of their records."[85]

The assumptions about reliability and the supervening importance of the accuracy of input that were made by the New South Wales Law Reform Commission were adopted by the Australian Law Reform Commission. In its 1981 *Hearsay Evidence Proposal* the Commission argued that

> [w]hile it is true that errors, accidental and deliberate, occur and can occur at every stage of the process of record keeping by computers the fact is, however, that they are the exception rather than the rule. They tend to occur at the stage when the information is fed into the system and there are techniques available which can be, and are, employed at each stage of the record keeping process to eliminate error.[86]

This passage was copied almost verbatim into the Commission's 1985 *Evidence (Interim) Report*.[87] It and other portions of the 1981 proposal were "fully endorsed" by the New Zealand Law Reform Commission in a report on the same subject.[88]

---

84 New South Wales Law Reform Commission, Note 41 at Appendix D, paragraph 73. The Commission's use of the term 'reliable' is imprecise, but plainly refers to accuracy rather than functional dependability.

85 New South Wales Law Reform Commission, Note 41 at Appendix D, paragraph 74.

86 Australian Law Reform Commission, *Evidence Reference, Research Paper No. 3, Hearsay Evidence Proposal* (Sydney: Australian Law Reform Commission, 1981) 127-128.

87 Australian Law Reform Commission, Note 6 at Volume 1, paragraph 705.

88 New Zealand Evidence Law Reform Committee, *Report on Business Records and Computer Output* (Wellington: The Committee, 1987) paragraph 127.

In the United Kingdom, the Law Commission arrived at a similar conclusion, albeit that it was concerned with the 'presumptive' rather than 'substantial equivalence' approach. In making its recent (1997) recommendations for repeal of section 69 of the *Police and Criminal Evidence Act* 1984 (UK), the Law Commission[89] relied upon the views of Colin Tapper, who had earlier argued that

> most computer error is either immediately detectable or results from error in the data entered into the machine. So widely has this been accepted that it has become institutionalized into the acronym "GIGO," or "garbage in, garbage out.[90]

Other views, whilst not referring directly to the issue of reliability, plainly embrace the assumption that it is not a major source of error. Reed, for instance, adopts this position by arguing that "it is certainly the case today that a computer record is likely to be at least as accurate as one maintained on paper."[91]

The significance of these assumptions is that they serve the substantial equivalence paradigm so well. They advocate emphasis upon the 'recording' of information to the exclusion, or at least at the expense of, attention to subsequent processing and/or retrieval. Such a distribution of emphasis is entirely appropriate for manual record keeping systems. When, for instance, information is recorded with pen and paper, it is the 'input' information that is critical. If a human recorder makes a mistake at this 'recording' stage, the accuracy of the record is indisputably undermined. If, however, the information is recorded accurately, there is little or no subsequent 'processing' of information that could lead to inaccuracies in the 'output' that is retrieved.[92] No 'device' is operative in any relevant sense. There is therefore no need to consider any issue of reliability.

---

89 Law Commission, *Evidence in Criminal Proceedings Hearsay and Related Topics* (London: H.M.S.O., 1997) at paragraph 13.7.

90 Tapper C, "Discovery in Modern Times: A Voyage around the Common Law World" (1991) 67 *Chicago-Kent Law Review* 217, 248 (and verbatim in Tapper, Note 3 at 396).

91 Reed C, "The Admissibility and Authentication of Computer Evidence – A Confusion of Issues" [1990-91] 2 *The Computer Law and Security Report* 13-16, 14-15.

92 The case of a complex 'index card' or similar system might be an exception. A disruption to the order of filing of index cards after the entry of information on them would be akin to a processing error for the system. It could lead to inaccuracies when information is subsequently retrieved, because a search that depended upon the card sequence might lead to an incomplete retrieval of information.

In the case of a system of record keeping that is computerised, the information transformation model that was introduced in chapter one indicates that the position is very different. The process of transformation and with it the reliability of the system are important contributors to the accuracy—or inaccuracy—of output. A computerised system of record keeping can only be the equivalent of a manual system when its reliability is perfect such that no errors in processing will ever occur.[93] This is why views of the kind referred to above are vital to the substantial equivalence rationale. Without the assumption that computer reliability is high, the premise for treating computer-produced material "as any other"[94] has no foundation.

The problem is that the findings of chapter four make it clear that such an assumption ought not—and cannot—be made. The disparity between the findings of chapter four and the position adopted in the literature that has been referred to here is explicable primarily on the basis that the literature has failed to undertake an adequate (or in some cases, any) investigation of the operation and reliability of computers before seeking to express conclusions about those topics.

More generally, the literature discloses a largely uniform and uncritical view of the reliability of computers. In the legal arena, it is a pervasive view. It appears to have had its genesis during the early 1970's, a period in which computers were "emerg[ing] from their years of infancy."[95] At that time the information that was available to answer questions about the reliability of computers was very limited. For instance, the 1973 technical monograph *Security, Accuracy, and Privacy in Computer Systems*, which may have been authoritative at the time of its publication, had only the following to say about the reliability of computer software.

---

93 This is merely the condition of equivalence. It is not necessarily a suitable prerequisite for the use of output from the system in legal fact finding, since the goal of legal fact finding (and evidentiary treatment) is not to restrict the use of sources of information to those that are accurate to a demonstrable certainty.

94 Kurzban, Note 5 at 452, citing *Rosenberg v Collins* 624 F.2d 659, 665 (5th Cir, 1980).

95 Martin J, *Security, Accuracy, and Privacy in Computer Systems* (Englewood Cliffs, New Jersey: Prentice-Hall, 1973) 3.

> Software, especially in its initial versions, sometimes contains subtle undetected errors. It is more likely to cause machine failures or cause a job to be rejected than cause errors in output. Techniques are now becoming better understood for controlling the reliability of software.[96]

In this environment it seems that legal commentary was free to offer wholly unsupported opinions about the reliability of computers and about the causes of error in computer output.[97] Within a short space of time, these views came to be cited as authoritative on the questions with which they dealt.[98] It may seem at first sight strange that opinions about the characteristics of physical processes could be given without foundation, and then subsequently embraced in this way. An explanation for this may lie in the fact that obtaining actual empirical data that related to these processes was (and is) far from trivial.[99] In these circumstances, the simpler—if less rigorous—course was to adopt the position that had initially been struck.

This dominant view defined a particular landscape of attitudes toward computers. It was within this landscape that the idea that computer output should be dealt with in the same way as other material was able to take root. The fundamental problem is that the view has not been supported on any rational basis. On the vital question of the reliability of computers, it is entirely uncritical. It is the product of a failure properly to examine this significant question at a suitable empirical level. This has had a distinct legacy: it leaves the substantial equivalence approach open to the criticism that it fails to address the issue of reliability in a rational manner.

---

96 Martin, Note 95 at 22.

97 See for example Roberts A, "A Practitioner's Primer on Computer-Generated Evidence" (1974) 41 *University of Chicago Law Review* 254-280, 264; Storm P M, "Admitting Computer Generated Records: A Presumption of Reliability" (1984) 18 *John Marshall Law Review* 115-154, 120-121. Many of the views expressed by the New South Wales Law Reform Commission fall into this category. See for example: New South Wales Law Reform Commission, Note 41 at Appendix D at paragraphs 56 and 73.

98 See for example: Anonymous, "A Reconsideration of the Admissibility of Computer-Generated Evidence" (1977) *University of Pennsylvania Law Review* 425-451, 442-443. The views of the New South Wales Law Reform Commission were also used in this way, most notably by the Australian Law Reform Commission in its 1981 research paper, see Australian Law Reform Commission, Note 86.

99 As was explained in chapter four, this remains a problem to the present.

The criticism is made stronger by the findings of chapter four, which show that an assumption that on a general level the reliability of computers meets or exceeds some high threshold is not one that can be supported. For the foregoing reasons, the substantial equivalence approach must be regarded as suffering from the fundamental flaw that the assumptions upon which it depends lack a rational basis. This flaw means that the approach fails to deal rationally with the possibility that less than perfect reliability will contribute to inaccuracies. It is true that rules of admissibility cannot be structured in a way that attempts to guarantee to an absolute certainty the accuracy of information that may be used in fact finding. It is, however, necessary for those rules to deal with questions of reliability in a rational way. Rules of admissibility that implement a substantial equivalence approach fail to do this.

It is the case that the approach focuses more attention on the question of the information that is supplied to a computer. Here, no criticism can be made because it is at this point that considerations of equivalence can properly be advanced. To subject such information to the same admissibility rules, without regard to the medium or device to which it is to be supplied or committed, is defensible because at this point no 'processing' of the information takes place. This is the case even though the skill and competence of the person who supplies the information may vary from case to case. Attempts to deal with such a variable via rules of admissibility would have very wide implications for the majority of the information that is presently used in legal fact finding, because individuals with varying skills, abilities and motivations supply such information.

On this point, Brown makes apposite reference to the judgment of Hutley JA in *Albrighton v Royal Prince Alfred Hospital*,[100] noting his Honour's explanation that the object of the relevant legislation was to "to admit records of relevant facts, without regard to the quality of the recorder." It is the attempt to extend this sentiment to the computerised setting that reveals the weakness in Brown's position. He appears to confuse the reliability (or competence) of the person who supplies the information with the reliability of the computer itself. He suggests that

---

100 [1980] 2 NSWLR 542.

> [t]here is no need here to consider questions of relative reliability. The issue of admissibility is quite distinct from that of weight and, as the Court of Appeal said in Albrighton's case, the legislation makes statements admissible, "without regard to the quality of the recorder."[101]

The crucial point is that computerised processes are processes of information transformation. What is done to the information after it has been supplied is as important a determinant of ultimate accuracy as is the initial input information. The distinction that must be made is between concerns about the quality of the recorder on the one hand, and the quality of the *processor* on the other.

5.2.2.3 Other expressions of the approach: public records

As the material that was considered in sections 5.2.2.1 and 5.2.2.2 indicates, the rule against hearsay has a significant role in the application of the substantial equivalence approach to evidentiary treatment. There are, however, a number of other exceptions to that rule. This fact gives rise to the possibility that computer produced material that is not a 'business record' might still be the subject of application of a substantial equivalence approach.

One of the other exceptions that has been considered in this context is the so-called 'public records' exception. Kerr cites *Hughes v United States* 953 F.2d 531 (9th Cir. 1992) as an example of the application of the public records exception in Fed. R. Evid. 803(8) to computer-produced material.[102] The language used by the *Hughes* Court plainly reveals that an 'equivalence' analysis was attractive to it. Indeed this analysis was thought sufficient to override concerns raised by the appellants concerning the lack of foundation for the introduction of the records in question as evidence.

---

101 Brown, Note 4 at 367.

102 Kerr, Note 17 at 31. For a discussion of the evidentiary treatment of public documents in Commonwealth jurisdictions see Brown, Note 4 at 62-71.

> The [appellants] next contend that the Forms were inadmissible because they were generated by a computer, and the government did not lay the foundation necessary for the admission of such computerized evidence. We reject this argument because this circuit as well as other circuits have held that official IRS documents, even if generated by a computer, are admissible as public records.[103]

The Court went on to find that the records were "self-authenticating domestic public documents under Fed. R. Evid. 902(1) because they were certified under seal."[104]

It is of interest that the *Hughes* Court appears to have regarded the provisions of that rule as effective to dispense with the need to deal with the particular authentication issues that arose because the material had been produced by a computer. In this respect the Court relied upon the decision in *United States v Farris* 517 F.2d 226 (7th Cir, 1975) which reached a similar conclusion about the capacity of public records to be self-authenticating. In reviewing the admission at trial of the material in question, the *Farris* Court held that "[t]here was no error in admitting the self-authenticated official computer print-outs."[105]

These decisions reflect a judicial comfort in applying in a medium neutral way provisions that provide for the self-authentication of material. Even when a computerised process produces the material in question, it is felt that there is no requirement for the provision of any assurance about the reliability of that process. Although no explicit reference to an assumption about reliability is made, the course that is taken in fact necessitates a more direct recourse to such an assumption.

The assumption is that the reliability of any computerised process that is involved in the production of the relevant record will be such that it will not contribute to inaccuracies in output. As was the case for the predominant (business records) expressions of the approach, the assumption must be made on a general basis. It is noteworthy that in the cases referred to, no attempt was made to impose any criterion of reliance. This may be congruent with the way in which computerised public

---

103 953 F.2d 531, 540.
104 953 F.2d 531, 540.
105 517 F.2d 226, 228-229.

records might be dealt with under the *Evidence Act* 1995 (Cth).[106] There, section 182(1) of the Act explicitly extends the operation of section 147 to

> documents that:
> (a) are, or form part of, Commonwealth records; or
> (b) at the time they were produced were, or formed part of, Commonwealth records.

Whether the effect of this provision will deviate from the United States position depends upon whether the true construction of section 147 is to impose a use and reliance condition upon the operation of the presumption, or merely the bare criterion of use. In the latter case, it seems that all that will be required is that the relevant device or process be 'used' to produce Commonwealth records in a given instance. This outcome would effectively mirror the application of the approach to public records in the United States.[107] In either case, the 'public records' expressions of the substantial equivalence approach that have been referred to in this section are susceptible to precisely the same criticisms that have been made of the 'business records' expressions of that approach.

### 5.2.3 The 'presumptive' approach

#### 5.2.3.1 The scope of the presumption

The essence of the presumptive approach is that it seeks to address the issue of the reliability of computers by applying a presumption as to their 'proper operation'.[108] It can be seen at the outset that, in terms of the model of the operation of a computer that was considered in chapters three and four, the presumption is one as to the reliability of the computer. It is not a presumption as to the accuracy of output. This is

---

106 This section is one of the few variations between the Commonwealth, New South Wales and Tasmanian versions of the Acts. It appears in only the Commonwealth version.

107 Although it is not mentioned explicitly in the relevant decisions, it is clear that the production of public records by a computer involves the 'use' of that computer.

108 Compare the Law Commission's reference to "proper functioning": Law Commission, Note 84 at paragraph 13.23.

illustrated by sections 146 and 147 of the *Evidence Act* 1995 (Cth), which are statutory expressions of the presumptive approach.[109]

These sections refer to the "production of outcome[s]" rather than to the characteristics of the product itself. In each case the party tendering the document or, in the case of s 146, the "document or thing", must assert that, in producing the document (or thing), the device or process produced a particular outcome. If the conditions prescribed by each section are met, then the presumption in each case is that

> (unless evidence sufficient to raise doubt about the presumption is adduced) ...in producing the document [or thing] on the occasion in question, the device or process produced that outcome.[110]

This presumption is altogether different from a presumption that information contained in or on the document (or thing) is accurate. As Brown observes, these sections restate the common law presumption,[111] which is one of proper operation. The presumption that is provided for by sections 146 and 147 cannot extend to the accuracy of the output of a computer because it has no operation in respect of the accuracy of input information. Yet both accuracy of input and correctness of operation are prerequisites to the accuracy of the output. Neither section 146 nor section 147 (nor the common law presumption) purport to deal with the accuracy of input information. For this reason, they cannot apply directly to the question of the accuracy of the information that is contained in any output.

In this respect the judgment of Spigelman CJ judgment in *R v Dudko* [2002] NSWCCA 336 plainly reads too much into section 147. His Honour said that

> [s]ection 147 of the Evidence Act provides that evidence of this character [i.e. a computer record], which forms part of the records of a business or is kept for

---

109 As was noted in section 5.2.2.1, section 147 has a hybrid character; it is relevant to two different approaches to evidentiary treatment under the classification scheme that is adopted in this thesis.

110 *Evidence Act* 1995 (Cth), s 146(2) and 147(2).

111 As to which see Note 8.

> the purposes of a business, is *presumed to have been accurately produced* by the device which makes the record.[112] (Emphasis added).

This misreading occurs elsewhere. For example, Crowley-Smith interprets section 147 as having the result that the "content of a document ... is presumed accurate pursuant to subsection (2)."[113]

5.2.3.2 Expressions of the approach

The three prominent expressions of the presumptive approach that are applied in common law jurisdictions are closely connected. They are section 146 and 147 of the *Evidence Act* 1995 (Cth) and the common law presumption. The effect of each is to give rise to a rebuttable[114] presumption that in producing particular output a computer has operated without failure. Although they are largely coincident, the scope of application of each differs slightly.

*Common Law*

The common law presumption is applied most directly in the United Kingdom.[115] Although it was displaced in that jurisdiction by statutory provisions that implemented a computer specific approach for both criminal[116] and civil[117] proceedings, it has recently been 'revived' in the criminal context. The relevant criminal provisions were repealed in 1999[118] without the enactment of a replacement

---

112 [2002] NSWCCA 336 at 53 per Spigelman CJ.

113 Crowley-Smith L, "The Evidence Act 1995 (Cth): Should Computer Data be Presumed Accurate?" (1996) 22 *Monash University Law Review* 166-173, 169 and see also at 172-173.

114 The common law presumption is a rebuttable presumption of law: Howard, Note 28 at paragraph 4-28. Sections 146 and 147 of the *Evidence Act* 1995 (Cth) expressly state the criterion for rebuttal, namely that "evidence sufficient to raise doubt about the presumption is adduced."

115 The common law presumption may apply in Australia even in the jurisdictions covered by the *Evidence Act* 1995 (Cth) because, as Brown correctly observes, both section 146 and 147 of that Act supplement, rather than modify or replace, any common law presumption that might apply. This is due to the saving effect of s 9(2)(b) of the Act, see Brown, Note 4 at 321-2.

116 *Police and Criminal Evidence Act* 1984 (UK), s 69.

117 *Civil Evidence Act* 1968 (UK), s 5. As discussed in section 5.2.2.1.4, this provision was repealed by the *Civil Evidence Act* 1995 (UK).

118 See *Youth Justice and Criminal Evidence Act* 1999 (UK), s 60(1).

provision of any kind, but on the express basis that the position would subsequently be covered by a common law presumption.[119] Of its recommendation for the repeal of the criminal provisions, the Law Commission made the following comments.

> We are satisfied that section 69 serves no useful purpose. We are not aware of any difficulties encountered in those jurisdictions that have no equivalent. We are satisfied that the presumption of proper functioning would apply to computers.[120]

Quinn supports the expectation of the Law Commission that the presumption would be applicable to computers. She contends that "the presumption regarding mechanical instruments applied to *all* types of computer evidence, regardless of the purpose for which it is adduced."[121] She notes, however, that

> the presumption of regularity developed initially in relation to the lawfulness of official actions and its suitability to machines, such as computers, has been questioned.[122]

Whether the presumption is truly applicable to the operation of a computer is a question of interpretation that need not be resolved for the purposes of the present discussion. The applicability of the presumption has been asserted at a sufficiently high level that evaluation of it as an expression of the presumptive approach is appropriate. The question that must instead be addressed is whether the use of this presumption is an adequate mechanism for dealing with the issue of reliability in a legal fact finding context. This consideration notwithstanding, it is noteworthy that if it is applicable to devices at all, the presumption is plainly limited to instruments that are "of a kind as to which it is common knowledge that they are more often than not in working order".[123]

---

119 Law Commission, Note 89 at paragraph 13.23.

120 Law Commission, Note 89 at paragraph 13.23.

121 Quinn K, "Computer Evidence in Criminal Proceedings: Farewell to the Ill-fated Section 69 of the Police and Criminal Evidence Act 1984" (2001) 5 *International Journal of Evidence and Proof* 174-187, 183.

122 Quinn, Note 121 at 183, referring to Zuckerman A A S, *The Principles of Criminal Evidence* (Oxford: Clarendon Press, 1989) 195 as an instance of such questioning.

123 Heydon, Note 22 at paragraph [1180], discussing the judgment of Stephen Brown LJ in *Castle v Cross* [1984] 1 WLR 1372 at 1377 and the reference in that judgment to the fifth United Kingdom edition of *Cross on Evidence*: Cross R, *Evidence* (London: Butterworths, 5th ed, 1979) at page 47.

This qualification presents substantial questions about the applicability of the presumption to computers. The presumption must be applied to the computer as a whole. It is not enough that, for example, it is applied to the hardware alone. For the reasons given in chapter four, the proper functioning of the hardware is insufficient to establish the reliability of the computer.[124] A presumption that hardware has operated correctly cannot serve to provide any assurance about the accuracy of information that is contained in the output that the computer may produce. The presumption must, if it is to be useful, somehow be applied to software as well.

The pertinent concern is that, for the reasons given in chapter four, each item of software is properly regarded as unique. More importantly, it cannot be assumed (or indeed known) that software in general exhibits a minimum threshold of reliability. The result is that software cannot be regarded as something[125] that is commonly known to be more often than not in working order. This means that the condition for application of the common law presumption cannot be satisfied for (at least) software. The same conclusion might be available in respect of hardware as well, but it is not necessary to consider this possibility here since the uncertainty surrounding a general state of software reliability is adequate to render the presumption inapplicable.

*Section 146 of the Evidence Act*

Section 146 of the *Evidence Act* 1995 (Cth) removes the constraint of 'prior common knowledge' as to proper functioning and instead imposes a condition of 'reasonableness' which is in the following terms.

> (1) This section applies to a document or thing:
>
> (a) that is produced wholly or partly by a device or process; and
>
> (b) that is tendered by a party who asserts that, in producing the document or thing, the device or process has produced a particular outcome.

---

[124] Because the reliability of a computer is determined by the reliability of the hardware *and* the reliability of the software: see chapter four, section 4.2.3.

[125] Software is not a mechanical device and arguably not a 'thing', but this is purely a formal point. It is not the basis upon which it is suggested that the common law presumption should not apply to software.

(2) If it is reasonably open to find that the device or process is one that, or is of a kind that, if properly used, ordinarily produces that outcome, it is presumed (unless evidence sufficient to raise doubt about the presumption is adduced) that, in producing the document or thing on the occasion in question, the device or process produced that outcome.

The condition that section 146(2) introduces is significant in three respects.

- First, what is required to be made is a finding about the reliability of the device or process in the sense in which that term was introduced in chapter one. The definition that was adopted is that of the Institute of Electrical and Electronics Engineers, namely "the ability of an item to perform a stated function under stated conditions for a stated period of time."[126] This definition is plainly congruent with the notion of the outcome that a device or process will 'ordinarily produce' if it is used properly. The concepts to which section 146(2) refers are the function that the device or process is designed (or expected) to realise, and the probability that it will realise that function as a matter of course.

- Second, despite the need to make a finding about what is plainly a complex characteristic, no guidance is given as to what circumstances will render a finding 'reasonably open'.

- Third, it is possible for the provision to apply upon a finding about the characteristics of a specific device or process, rather than the characteristics of the entire class or "kind" of device or process of which the one in question is a member. The discussion in chapter four indicated that there are very substantial difficulties in attempting to deal with questions about the reliability of computers on a general, as opposed to an individual, basis. In this respect the provision departs from the common law expression of the approach. The

---

[126] Institute of Electrical and Electronics Engineers, *The IEEE Standard Dictionary of Electrical and Electronic Terms* (New York: Institute of Electrical and Electronics Engineers, 6th ed, 1996) 904.

possibility of making a finding as to only the specific device or process in question may therefore render the provision of greater utility than would otherwise be the case.

*Section 147 of the Evidence Act*

Section 147 of the *Evidence Act* 1995 (Cth) was considered in section 5.2.2.1.2 in a 'business records' context. It is considered here for its effect as a presumption of the proper operation of a computer. To this extent it is largely coincident with section 146 of the Act. The distinction between the two provisions lies in the conditions under which they will be available. These conditions are considered in the following section.

5.2.3.3 Evaluation

*The conditions upon which the presumption depends*

The presumptive approach presents a difficult subject for evaluation in terms of the standards and criteria that have been adopted in this thesis. At the outset it can be said that the blanket application of a presumption as to proper operation cannot be sustained as rational. Such a presumption equates to a presumption of reliability. As the findings of chapter four indicate, there is no empirical basis for the making of such a presumption. Subject to the possibility of rebuttal,[127] such an application would be no improvement upon the shortcomings of the substantial equivalence approach.

The expressions of the presumptive approach that were described in section 5.2.3.2 do not, however, purport to apply this presumption in an unconditional way. Each attaches at least one criterion to the application of the underlying presumption.[128] In

---

127 See Note 114.

128 This statement assumes that in the case of the common law expression of the approach, the condition of common knowledge is to be applied: see section 5.2.3.2. If this is not the case, the unconditional application of a presumption of reliability to computers is unsustainable on the basis of the findings of chapter four.

this way, the approach is in each case considerably more restricted than would otherwise be the case. Except in the case of s 147 of the *Evidence Act* 1995 (Cth), it is difficult to assert that proper application of the relevant conditions produces an outcome that has no rational foundation.[129]

In the case of the common law presumption, the condition that computers (as a class of device) should be commonly known to be "more often than not in working order"[130] is closely congruent with a condition that there be empirical experience that indicates that computers exhibit reliability at or above some high threshold. Satisfaction of such a condition would permit rational inferences to be made about the reliability of computers generally. It would supply a sound statistical basis for the making of such inferences. This would in turn represent an adequate foundation for application of the presumption. The finding of 'reasonableness' that is required to be made in connection with section 146 may also provide such a foundation, albeit that the prerequisites for making such a finding are not stated as explicitly as might otherwise have been done.

The difficulty that arises in connection with both the common law and section 146 expressions of the presumptive approach is that there is a considerable potential for misapplication. In each case, the relevant conditions (common knowledge, reasonableness) must be applied to the combination of hardware and software. It is not enough to consider, for instance, the hardware alone.[131] It will be insufficient to conclude that computer hardware is a device that is commonly known to be 'more often than not' in working order or to find that it is a device that ordinarily operates without failure, even if there was a clear basis for doing so.

---

129 Section 147 suffers from the shortcomings that were discussed in conjunction with the 'use and reliance' criterion that is associated with the substantial equivalence approach: see section 5.2.2.2.1.

130 See Note 123.

131 Because the reliability of a computer is determined by the reliability of the hardware and the reliability of the software.

This point deserves emphasis because it may be tempting to consider (or to accept) that computer hardware is a largely standard item that is used so widely that it can properly be said to be a subject about which there is a fund of 'common knowledge'. It seems, however, much less likely that the same could be accepted for software, which is much less standardised. Yet, as was pointed out in section 5.2.3.2, the presumption that is to be applied must be applied to the computer system as a whole, not merely to the hardware. In his discussion of the common law presumption and its relationship with section 146 of the *Evidence Act* 1995 (Cth), Brown overlooks this point. He appears to distinguish the 'computer' from the 'program', arguing that

> [a] computer is a universal machine, capable of being programmed to perform any computable task and, therefore one which, if properly used, will produce the logical outcome of its programming.[132]

The problem is that if the 'device or process' or, in terms of the common law presumption the 'instrument', to which the presumption is to apply is the hardware alone, then the question of the reliability of the software is left unanswered. The argument here is not the purely formal one that software cannot be considered to be an instrument, device or process because it is intangible. It can and should be regarded as a functional unit that is capable of giving rise to the same issues of proper operation and reliability that arise in the case of hardware. That the hardware alone has operated correctly (or can be presumed to have done so) is a necessary, but not a sufficient, basis for drawing rational inferences as to the accuracy of the output that a computer might produce. It will be impossible to have any confidence about the accuracy of information that is contained in given computer output unless the operation of the software, and the possibility of its failure, are also considered.

Brown's analysis demonstrates one of the key respects in which the requirements of the condition that attaches to the common law presumption might be incorrectly applied by a fact finder. The condition of common knowledge must be satisfied for the hardware and software combination to which the presumption of reliability is to apply, not merely for one component. The condition might also be misapplied if the

---

132 Brown, Note 4 at 323.

basis for asserting the existence of the requisite common knowledge is inadequate. This is a problem that is closely related to the difficulties that were identified in connection with the substantial equivalence approach. Here the relevant question is this: what is an adequate foundation for asserting that it is common knowledge that computers[133] are more often than not in working order? In the case of the substantial equivalence approach the question was instead: what is an adequate basis for assuming that computers more often than not function correctly?[134]

The difficulties in making assessments about the reliability of computers in general that were identified in chapter four are apposite here. If it cannot be demonstrated that the reliability of computers generally meets or exceeds some high threshold, then it is difficult, if not impossible, to demonstrate that this (unverified) conjecture is the subject of common knowledge.

The condition that applies to the presumption under section 146 of the *Evidence Act* 1995 (Cth) presents difficulties of a slightly different nature. What is required under the section is that a finding of reliability be 'reasonably open'. It was observed in section 5.2.3.2 that no guidance is given as to what this may require. If the discussion in chapter four is taken into account, it is apparent that the assessment of the reliability of computers on a rational basis is a very difficult task. On a general level, it is probably impossible. In the case of a specific computer, it will be a very significant and difficult task to deal with the question of the reliability of the software alone, even before the question of the reliability of the hardware is considered.

Such critical matters may, however, be overlooked when the section is applied. The report of the Australian Law Reform Commission[135] which recommended the enactment of section 146 and the balance of the *Evidence Act* 1995 (Cth) certainly does not deal with them. Rather, as was discussed in section 5.2.2.2.2, the report

---

133 As combinations of hardware and software.

134 The question was previously couched in terms of the reliability of computers generally exceeding some high threshold (section 5.2.2.2), but both are directed to the same issue.

135 Australian Law Reform Commission, Note 6.

propounds the view that computers are generally reliable and the errors in output are most often caused by errors in input, rather than by failure of the computer itself.

A court seeking to apply section 146 may well have regard to these assertions[136] to conclude (incorrectly) that a finding as to reasonableness can be made in the case of computers without any further factual foundation. In this scenario, the reliability of computers might be considered a matter about which little or no enquiry is required. This is the course that Brown foresees.

> There will come a point at which the courts will treat all computers in their computational roles as proven reliable scientific instruments. The next step will be to accept the reliability of all machine processes, subject to evidence to the contrary, and to focus on the reliability of any data source as the most likely source of error in output.[137]

To proceed on such a footing would be, in effect, to make the application of a presumption unconditional. The question of reliability would cease to be an issue, apart from the possibility of rebuttal. In these circumstances, the application of a presumption of proper operation, which is a presumption of reliability, would lack an adequate foundation.

The problem that reliability is a matter that is difficult to establish is very significant, and the solution that the presumptive approach attempts to provide is unique. The expressions of it which have been considered must, however, face the criticism that they do not provide an adequate basis for the application of the solution that is offered. Whilst attempts have been made to limit the availability of the approach to cases in which particular conditions have been established, it seems unlikely that those conditions will be effective to ensure that there will be an adequate basis for

---

[136] Section 15AB(1) of the *Acts Interpretation Act* 1901 (Cth) provides that regard may be had to extrinsic material for purposes relating to the interpretation of a provision of an Act. Section 15AB(2)(b) provides that the material to which regard may be had includes "any relevant report of a Royal Commission, Law Reform Commission, committee of inquiry or other similar body that was laid before either House of the Parliament before the time when the provision was enacted." The relevant Australian Law Reform Commission report (Report 26, *Evidence (Interim)*) appears to meet this condition. It was 'presented' to the Commonwealth Parliament on 21 August 1985, see: Parliament of the Commonwealth of Australia, *Parliamentary Paper No. 302/1985*.

[137] Brown, Note 4 at 367.

application of the presumption. These conditions appear not to overcome the substantive difficulties that are presented by the findings of chapter four.

*The use of a presumption as a response to the problem of difficulty of proof*

A further criticism that must also be considered is whether the use of a presumption is justified as a response to a problem of difficulty of proof. Chapter four demonstrated that proof of the reliability of a computer system is a difficult task. Aspects of this difficulty of proof have been recognised in connection with the development of the presumptive approach. The issue was raised by the Australian Law Reform Commission in the following terms.

> To require extensive proof, on each occasion, of the reliability [sic – accuracy] of ... computer records is to place a costly burden upon the party seeking to tender the evidence, to give the opposing party a substantial tactical weapon and to add to the work of the courts. In many cases there will be no bona fide issue as to the accuracy of the record. It is more efficient to leave the party against whom the evidence is led to raise any queries and to make any challenges it may have.[138]

The use of a presumption does not, however, cure the difficulty. It merely transfers the problem to the opposing party, who is then obliged to raise a doubt about the issue of reliability. Failure to do this will lead to the admission of the material in question.[139]

The result in such a case will be the admission of material when no effective steps to investigate or to assure the reliability of the computer in question have been taken. This situation will come about in the case of material admitted pursuant to section 147 of the *Evidence Act* 1995 (Cth) because, as has been demonstrated, the condition imposed by that section does not provide an adequate assurance of reliability. In the

---

138 Australian Law Reform Commission, Note 6 at Volume 1, paragraph 705.

139 In the case of the *Evidence Act* 1995 (Cth), admission is subject to a general discretion under s 135 to exclude the evidence "if its probative value is substantially outweighed by the danger that [it] might:

(a) be unfairly prejudicial to a party; or
(b) be misleading or confusing; or
(c) cause or result in undue waste of time.

case of the common law presumption, or under section 146 of the Act, it will come about because the conditions that govern the availability of the presumption in those cases have not been adequately applied.

Consideration of the use of a presumption as a mechanism of evidentiary treatment therefore presents a fundamental choice. It is between restricting the use of computer output to the product of a computer which has been shown to be reliable (knowing that proof of reliability is difficult) on the one hand, or instead allowing the product of a computer to be used simply because that computer has *not* been shown to *lack* reliability.[140]

Each alternative has its own shortcomings. Requiring that reliability be demonstrated as a condition precedent to admission introduces the risk that the product of a highly reliable computer will be excluded merely for want of proof of reliability. Yet the position advocated by the Australian Law Reform Commission presents the problem that the opposing party may not be able to demonstrate a lack of reliability. The difficulties of proof of reliability that are indicated by the discussion in chapter four apply, even in a specific case, to attempts to prove that there is a basis for doubting proper operation or reliability.

It is true that in the case of sections 146 and 147 of the *Evidence Act* 1995 (Cth), such difficulties may be addressed via the 'request' mechanism in sections 166–169. That mechanism is designed to allow, among other things, access to documents and 'things' for the purposes of examination, testing and copying.[141] It may be used to obtain "documents" or "things" for production, examination or testing[142] in order to determine "a question that relates to the authenticity, identity or admissibility."[143] The apparent usefulness and wide availability of this mechanism must, however, be

---

140 In each case the standard of reliability would be some suitably high level; it need not be absolute.

141 *Evidence Act* 1995 (Cth), s 166.

142 *Evidence Act* 1995 (Cth), s 166.

143 *Evidence Act* 1995 (Cth), s 167(c).

viewed as limited in light of the restrictive judicial interpretation that has been given to it.[144]

It is difficult to justify a preference for one alternative over the other in a way that is truly compelling. The question at hand is not one that can be answered merely by reference to the ideal of rational truth identification. The potential for the admission of material for which there has been no scrutiny of the issue of reliability is not absolute, since it is ceases to apply once doubts about reliability are raised. It is therefore not possible to say that the approach lacks any rational foundation. As is the case with any approach to evidentiary treatment, it may in particular cases operate in a way that results in the admission of inaccurate material. Certainty of accuracy is not, however, the benchmark by which the operation of any such approach is to be reviewed.[145]

The answer may lie in the view that the use of a presumption (even one that may be rebutted) in place of a requirement of some demonstration of reliability seems to be better suited to processes in respect of which an initial assumption of high reliability can justifiably be made. This was arguably the view of the Australian Law Reform Commission, which expressed in very clear terms the belief that, even apart from any question of difficulty of proof, such an assumption could be made about computers.[146] As has been noted, the Commission advocated the use of a rebuttable presumption about reliability.[147] It was, however, in the same report that the proposition that errors in record keeping by computers "are the exception rather than the rule [and that] they

---

144 See for example *Australian Petroleum Pty Ltd v Parnell Transport Industries Pty Ltd & Ors* [1998] FCR 537, 540-542 (request mechanism need not be afforded as a prerequisite to admissibility under s 69 of the Act); *Commissioner of Taxation v Karageorge* (1996) 22 ACSR 199 (request mechanism does not apply to documents tendered under s 1274 of the *Corporations Law* (Cth)); *Telstra Corporation v Australis Media Holdings* (NSW Supreme Court, 18 March 1997, McLelland CJ in Equity, unreported) (request mechanism does not apply to interlocutory proceedings); *Pecar v National Australia Trustees Ltd* (NSW Supreme Court, 27 November 1996, Bryson J, unreported) (request mechanism not available where compliance with the request requires the participation of a non-party).

145 As was discussed in chapter two, section 2.3.1.2, the function of rules about admissibility, which in the present context also includes rules about presumptions, is to operate on the basis of likelihood of accuracy, rather than certainty.

146 Australian Law Reform Commission, Note 86.

147 Australian Law Reform Commission, Note 6 at Volume 1, paragraph 705.

tend to occur at the stage when the information is fed into the system" was presented as fact.[148] The Law Commission expressed similar views in conjunction with its recommendations to repeal section 69 of the *Police and Criminal Evidence Act* 1984 (UK).[149]

Why was it important to hold and to express such a belief in conjunction with proposals for the adoption of a presumptive approach, unless it advanced the case for that approach? When it is shown that this belief has no rational basis,[150] the use of a presumption appears less attractive. It cannot be said that the application of the presumption is entirely devoid of a rational foundation, since it admits the possibility of doubt about reliability and, therefore, rebuttal. At the same time, use of a presumption appears to have a considerably weaker basis when it is appreciated that no general assumption can rationally be made about the reliability of computers.

The result is that the optimal conditions for use of a rebuttable presumption do not exist. It is true that proof of reliability is a costly and difficult matter, as the Australian Law Reform Commission has suggested. There are, however, no means by which the reliability of computers can *in fact* be said generally to meet or exceed some acceptably high threshold. The situation is not one of mere difficulty or expense of proof. Information that is contained in computer output may or may not be affected by computer failure in any given case and may show no signs of inaccuracy on its face.[151] Whatever assumptions might be made on the basis of instinct or intuition, it cannot be *demonstrated* in a logically robust manner that this possibility is limited to an acceptably low margin.

There is a resulting uncertainty about reliability, and about the possibility of failure that will obtain for each instance in which computer output is considered. Whilst it is not the function of the law of evidence to eradicate this uncertainty entirely, attaching

---

148 Australian Law Reform Commission, Note 6 at Volume 1, paragraph 705.

149 Law Commission, Note 89.

150 As was done in chapter four.

151 This follows from the fact that not all failures will prevent the production of material. As was discussed in chapter four, a failure of software in particular might affect the accuracy of output without this being apparent on the face of the material.

a rebuttable presumption of reliability is not an appropriate response to it. If the position were otherwise, the mere fact of difficulty or expense in proving reliability would warrant the application of a presumption to dispense with proof not only in the case of computer-produced material, but in other cases as well.

These cases could involve a range of complex scientific and technical information for which proof of reliability is costly and difficult.[152] The present position for such information is, however, that proof of reliability remains a condition of admissibility.[153] Such a condition is imposed undoubtedly because there is no generally shared view that the complex underlying processes are so notoriously reliable that the issue of reliability can be ignored by the fact finder. For a given item of software (and possibly for some items of hardware too) the position is very close to this situation. As has plainly been demonstrated in chapter four, there can be no shared view about the reliability of software. For this reason, there is no sound basis upon which computer-produced material should have the benefit of a presumption of the kind that has been considered here.

---

[152] Such as reports of DNA analyses for example. For a brief account of the complexity underlying this kind of information, see chapter one, section 1.1.1.1. For early judicial consideration of this technology see: *People v. Wesley* 140 Misc.2d 306, 307-308 (Albany County Ct, 1988); *People v. Castro* 545 NYS.2d 985, 961-962 (1989).

[153] Almost invariably through the mechanism of 'expert' opinion. See for example, Freckleton I and Selby H, *The Law of Expert Evidence* (Sydney: LBC Information Services, 1999) 48. In the case of DNA technology, the reliability of the particular processes that are used in DNA technology must be proved as a condition of the admissibility of the results of tests performed with that technology, see for example: Flannery I M, "Frye or Frye Not: Should the reliability of DNA Evidence Be a Question Of Weight Or Admissibility?" (1992) 30 *American Criminal Law Review* 161-186, 181; Thompson WC and Ford S, "DNA Typing: Acceptance and Weight of the New Genetic Identification Tests" (1989) 75 *Virginia Law Review* 45-108. It is noteworthy that whilst there has been debate about whether proof of the reliability of testing methodologies, or *how* the relevant technology is used in a given setting, should be a condition of admissibility, that debate has not questioned the need to furnish proof of the reliability of the underlying technology itself: see for example Flannery at 181.

### 5.2.4 The 'specific computer' approach

5.2.4.1 The principal expressions of the approach

The specific computer approach is distinct from the other approaches to evidentiary treatment in two important respects. First, it recognises in a direct way the need to address the issue of whether a computer has operated correctly in producing material that is to be admitted.[154] Second, it focuses upon the reliability of only the computer that produced the material that is of interest; it does not attempt to deal with the reliability of computers generally. The approach imposes conditions upon admissibility that relate directly to the operation of the computer and from which, it seems, an inference of reliability is to be drawn.

The first formulation of the specific computer approach dates to approximately 1968.[155] This vintage places it at the genesis of consideration of the issues presented by the potential use of computer-produced material in a forensic context. From this beginning, the specific computer approach has had a troubled history. That history culminated in its complete abolition in the United Kingdom,[156] the jurisdiction in which it originated. Elsewhere the approach currently has only limited significance. Closely congruent expressions of it are in force in three Australian states: Queensland,[157] South Australia[158] and Victoria.[159] Expressions of the approach are also in force in South Africa[160] and Israel.[161]

---

154 In some cases, as critics of the approach have observed, it does this to the exclusion of any attempt to verify the accuracy of the input information, see for example Tapper, Note 3 at 396.

155 The formulation was enacted as section 5 of the *Civil Evidence Act* 1968 (UK). The origin of the formulation is unclear. Tapper suggests that it was the result of the recommendations of the Law Reform Committee: Tapper, Note 3 at 394, referring to Law Reform Committee, *Thirteenth Report: Hearsay Evidence in Civil Proceedings* (London: H.M.S.O., 1966), but this report contains no recommendation as to the enactment of provisions that deal with computers specifically, much less a recommendation in the terms that appeared in section 5 of the Act. The Law Commission confirms that the 1966 report contained no relevant recommendation and suggests that the section 5 "would appear to have been something of an afterthought": Law Commission, Note 72 at paragraph 3.14.

156 In the United Kingdom the approach formerly applied in civil cases by virtue of section 5 of the *Civil Evidence Act* 1968 (UK), which was repealed by schedule 2 of the *Civil Evidence Act* 1995 (UK). The approach formerly applied in criminal cases by virtue of section 69 of the *Police and Criminal Evidence Act* 1984 (UK), which was repealed by s 60(1) of the *Youth Justice and Criminal Evidence Act* 1999 (UK).

157 *Evidence Act* 1977 (Qld), s 95.

158 *Evidence Act* 1929 (SA), s 59B.

The specific computer approach is diminishing in significance. In only one of the instances just mentioned (Israel) has the introduction of the approach been a recent initiative.[162] In addition to this, virtually every review of or commentary upon legislation that implements the approach has criticised it.[163]

In light of these special circumstances, a detailed review of each expression of the approach is not undertaken here. What is instead considered is the operation and effect of the approach in Queensland and Victoria. These jurisdictions apply expressions of the approach that are identical in all material respects. More importantly the provisions in these states are identical to the original *Civil Evidence Act* 1968 (UK) provisions, and it is the latter provisions which have attracted the most commentary.[164] Some comparative observations about the expression of the approach in South Australia are included in section 5.2.4.2.

*Queensland and Victoria*

In these jurisdictions, four matters are required to be demonstrated. Upon their demonstration, a document produced by a computer[165] is admissible to the following extent.

---

159 *Evidence Act* 1958 (Vic), s 55B.

160 *Computer Evidence Act* 57 of 1983 (South Africa), ss 2-3.

161 The relevant provision is section 36 of the *Evidence Ordinance (New Version)* 5731-1971 (Israel). It was inserted by the *Computer Law* (5755-1995): Sage E R, *Israel - Excerpts from the Computer Law* 5755-1995 <http://www.rgr.co.il/English/Resources/COMPUTERr.pdf> (Visited 2 June 2003). For a discussion of the provisions see Deutch M, "Computer Legislation: Israel's New Codified Approach" (1996) 14 *John Marshall Journal of Computer & Information Law* 461-482, 477-480.

162 *The Computer Law* (5755-1995) was enacted in 1995.

163 Examples of such criticism are given in the following section. The principal exception appears to be the relatively early review by McNiff, which abstains from direct criticism of the relevant Australian legislation: McNiff, Note 10.

164 Subsequent references to criticisms of the *Civil Evidence Act* 1968 (UK) provisions therefore have direct application to the Queensland and Victorian provisions.

165 The term 'computer' is defined as "any device for storing and processing information": *Evidence Act* 1977 (Qld), s 95(7); *Evidence Act* 1958 (Vic), s 55B(8).

- The document must contain a statement that tends to establish a fact of which direct oral evidence would have been admissible.

- If so, the statement is admissible as evidence of that fact.[166]

The four matters that must be demonstrated are[167]

(a) that the document containing the statement was produced by the computer during a period over which the computer was used regularly to store or process information for the purposes of any activities regularly carried on over that period, whether for profit or not, by any person; and

(b) that over that period there was regularly supplied to the computer in the ordinary course of those activities information of the kind contained in the statement or of the kind from which the information so contained is derived; and

(c) that throughout the material part of that period the computer was operating properly or, if not, that any respect in which it was not operating properly or was out of operation during that part of that period was not such as to affect the production of the document or the accuracy of its contents; and

(d) that the information contained in the statement reproduces or is derived from information supplied to the computer in the ordinary course of those activities.

The provisions also include a mechanism for the certification of these conditions by a 'qualified person',[168] and for dealing with instances in which more than one computer is involved in the production of the document of interest.[169] The court is given a residual discretion to exclude a statement that would otherwise be admissible under

---

166 *Evidence Act* 1977 (Qld), s 95(1); *Evidence Act* 1958 (Vic), s 55B(1).

167 *Evidence Act* 1977 (Qld), s 95(2); *Evidence Act* 1958 (Vic), s 55B(2).

168 Being a person "occupying a responsible position in relation to the operation of the relevant device or the management of the relevant activities (whichever is appropriate)": *Evidence Act* 1977 (Qld), s 95(4); *Evidence Act* 1958 (Vic), s 55B(4).

169 Combinations of computers are, in essence, treated as a single computer for the purposes of the provision: *Evidence Act* 1977 (Qld), s 95(3); *Evidence Act* 1958 (Vic), s 55B(3).

the provision if it appears that it would be "inexpedient in the interests of justice that the statement should be admitted."[170]

5.2.4.2 Evaluation

5.2.4.2.1 Areas of prior criticism

The specific computer approach to evidentiary treatment has been the subject of considerable criticism. That criticism has touched upon four largely related areas. What is most significant in the context of the present thesis, however, is a criticism that can be made in light of the findings of chapter four, but which has been almost entirely overlooked in the literature that deals with this approach.

The four principal respects in which the approach has previously been criticised are as follows.

(a) The need to have a special approach in light of the fact that there are alternative approaches has been questioned.

(b) Reference has been made to the failure of some expressions of the approach to deal with the issue of the accuracy of input information.

(c) It has been said that the approach is ineffective to address the issue of reliability.

(d) It has been said that some expressions of the approach are complex and that compliance with them is difficult.

---

170 *Evidence Act* 1977 (Qld), s 98; *Evidence Act* 1958 (Vic), s 55B(7).

It is in connection with the third of these matters that, drawing upon the findings of chapter four, a new and more important criticism may be made. Each area of criticism is considered in turn below.

*The need for a special approach in light of the alternatives that are available*

It is frequently argued that there is no need for an approach that accords different evidentiary treatment to computer output. In its place the application of one of the other two approaches that have been identified here (substantial equivalence, presumptive) is suggested as a preferable alternative.[171] However, as has been demonstrated earlier in this chapter,[172] each of those approaches suffers from shortcomings in the way in which they deal with the issue of reliability. The argument that a specific computer approach is not needed because there are viable alternatives can therefore be discounted.

*The failure to address the issue of the accuracy of input*

It is clear from the information transformation model that was presented in chapter one that both the accuracy of input information and the reliability of the process to which that information is subjected will contribute to the accuracy of the output. Both matters demand attention, although in the context of approaches such as the substantial equivalence approach, the principal criticism is plainly that the issue of reliability is the one that has been dealt with inadequately.[173]

In the case of some (but not all) expressions of the specific computer approach, the reverse is true. As Tapper points out, the *Civil Evidence Act* 1968 (UK) provisions contained

---

171 For example (advocating a substantial equivalence approach): Tapper, Note 3 at 395; Australian Law Reform Commission, Note 6 at Volume 1, paragraph 344(d); New Zealand Evidence Law Reform Committee, Note 88 at paragraph 125; New South Wales Law Reform Commission, Note 41 at paragraph 4; Law Commission, Note 72 at paragraphs 3.14-3.21 and (advocating a presumptive approach) Brown, Note 4 at 366-367; Law Commission, Note 88 at paragraph 13.23.

172 See sections 5.2.2.2 and 5.2.3.3.

173 See generally section 5.2.2.2.

> no requirement that the originator of the information processed by the computer should have had, or even be reasonably capable of being supposed to have had, personal knowledge of the truth of that information.[174]

To this extent, Tapper is correct[175] and this omission represents a major shortcoming of the expressions of the approach that are affected by it.[176]

*The effectiveness of the approach in addressing the issue of reliability*

In reviewing the *Civil Evidence Act* 1968 (UK) provisions, the Law Commission made the following observations.

> [T]here is a heavy reliance on the need to prove that the document has been produced in the normal course of business and in an uninterrupted course of activity. It is at least questionable whether these requirements provide any real safeguards in relation to the reliability of the hardware or software concerned.[177]

The Law Commission touches upon an important matter that is worthy of further examination in its own right. It relates to the inferences that are available from the fact that the computer of interest has operated for some period of time without failure.

The existence of a period of failure free operation is made significant by one of the conditions that is imposed by the Queensland and Victorian provisions. The relevant condition is

---

174 Tapper, Note 3 at 396. The point is also made by the Law Commission: Law Commission, Note 72 at paragraph 3.15.

175 Tapper subsequently argues that had this omission not occurred, there would have been no need for adoption of the conditions that were prescribed by the provisions: Tapper, Note 3 at 396-7. This is not correct. Assurances about the accuracy of input are necessary to provide assurances about the accuracy of output, but they are not sufficient to do so. They do nothing to address the issue of reliability.

176 Some other expressions of the approach do address the issue of the accuracy of input information. See for example *Evidence Act* 1929 (SA), s 59B(2)(b). The position under (the now repealed) section 69 of the *Police and Criminal Evidence Act* 1984 (UK) was less certain because it depended upon whether the input information amounted to hearsay. For a discussion, see Quinn, Note 121 at 175-178.

177 Law Commission, Note 72 at paragraph 3.15.

(c) that throughout the material part of that period the computer was operating properly ...[178]

The 'period' in question is the period within which the relevant material was produced and during which "the computer was used regularly to store or process information for the purposes of any activities regularly carried on over that period ..."[179]

This is the only condition that refers to proper operation. The others[180] are directed to the regularity of use of the computer. This condition is important in its own right because its inclusion represents an attempt to draw upon the experience of failure as a basis for assessing reliability. In this respect, the use of this condition defines an approach to the evidentiary treatment of computer-produced material that is congruent with engineering methodologies. Such congruence is not, however, a basis for automatic endorsement of the approach. What must be demonstrated is that the use of such a condition is also effective to promote the objective of rational truth identification.

The efficacy of the condition can be evaluated in two contexts. First, it can be asked if the condition is effective as an indicator of the reliability of the specific computer of interest. Second, it can be asked if the condition is effective as an indicator of the likelihood that particular material of interest has been affected by a computer failure. In examining these possibilities here, the primary problem associated with the observation of failure that was identified in chapter four is put to one side. That problem related to the difficulty of detecting non-obvious error.

To certify to the court that there has been a period of failure free operation implies that any failure that lead to obvious *or* non-obvious errors in output would have been noticed. Yet this implies the availability of an accurate failure detection mechanism.

---

178 *Evidence Act* 1977 (Qld), s 95(2)(c); *Evidence Act* 1958 (Vic), s 55B(2)(c). The South Australian regime also attaches significance to a failure free period. Section 59B(2)(d) of the *Evidence Act* 1929 (SA) provides the following as a condition of admissibility. "[T]he computer has not, during a period extending from the time of the introduction of the data to that of the production of the output, been subject to a malfunction that might reasonably be expected to affect the accuracy of the output."

179 *Evidence Act* 1977 (Qld), s 95(2)(a); *Evidence Act* 1958 (Vic), s 55B(2)(a).

180 The text of each of the four conditions was set out in section 5.2.4.1.

Such a mechanism would render obsolete any attempt to infer the likelihood of failure for given output. The mechanism itself would be a preferable means of assuring the absence of failure in connection with the material in question. A mechanism that, for instance, involved cross checking with original 'paper based' records implies that such records are readily available for consultation. If so, the need to meet the criteria for admission of the computer produced counterpart could be avoided by recourse to the original records as evidence.

It is the case, however, that the problems of efficacy that are identified below are at least as significant as this issue of failure recognition. Even if that latter issue could be addressed in some way, the problems of efficacy would still represent serious shortcomings in the specific computer approach.

(a) Effectiveness to indicate the reliability of the specific computer of interest

The reliability of a specific computer may be capable of assessment by reference to observations of its operation over some time interval. Chapter four considered the possibility of such assessment for the case of specific software.[181] It was observed that the underlying failure process for software is characterised by the chance that unpredictable inputs from some input domain would intersect unknown regions of sensitivity within that domain. Those regions are created by unknown faults in the software. Because the relevant variables cannot be predicted, the failure process must be treated as random. This characterisation will be the best that can be made unless and until information is available that can qualify the failure process in some more precise way. In the absence of such information, the times at which individual failures will occur will be unpredictable.

---

[181] It was demonstrated in chapter four that the reliability of a computer is governed by the reliability of both the hardware and software. Both elements affect the overall reliability that will be experienced. Difficulties in assessing the reliability of just one of these components will impede or prevent the assessment of the reliability of the system as a whole. For this reason the findings of chapter four have direct application in the present context, even though they are limited to the question of the assessment of the reliability of software.

What may be predictable is a distribution of inter-failure periods and a 'mean time to failure' for that distribution. The latter was identified in chapter four as a common measure of software reliability. Such a limited descriptor of the software is, however, of little or no usefulness in the fact finding environment because it cannot aid in the prediction of individual failures. It is worse still that even this limited measure requires a set of data that record a suitable *number* of failures and the times at which they occur. Without information about the times at which failures have occurred, it is impossible to infer the relevant probability distribution for inter-failure periods.

The observation of a *single* period of failure free operation, which is the scenario envisaged by the Queensland and Victorian provisions[182] permits no inferences about the reliability of the relevant software—and therefore the relevant computer—to be made. The position is made no better by the imposition of other conditions,[183] such as continuity of use,[184] that deal with matters other than failure. At their highest, such conditions appear to be a manifestation of a 'use and reliance' type criterion of the kind that was encountered in connection with the substantial equivalence approach. For the reasons that were given in section 5.2.2.2.1, such a criterion cannot of itself be a basis for inferring reliability.

(b) The likelihood that specific material is affected by failure

It is arguable that the issue to which the condition of observation of a failure free period of operation is directed is not the reliability of the specific computer in question. It may instead be the more immediate question of the likelihood that the particular material of interest was affected by some failure of the computer. There is, after all, only one relevant use for reliability measures in the legal fact finding

---

182 And also by section 59B(2)(d) of the *Evidence Act* 1929 (SA).

183 Namely the conditions set out in section 5.2.4.1. In the case of the South Australian regime, there are six other conditions apart from the one imposed by section s 59B(2)(d) of the *Evidence Act* 1929 (SA), but none is referable to failure.

184 See *Evidence Act* 1977 (Qld), s 95(2)(a) and *Evidence Act* 1958 (Vic), s 55B(2)(a). Compare *Evidence Act* 1929 (SA), s 59B(2)(a), which requires that the court be satisfied that "the computer is correctly programmed *and regularly used to produce output of the same kind* as that tendered in evidence pursuant to this section." (Emphasis added.)

environment. That use is to provide information about the likelihood that particular material has been affected by computer failure.

Measures of reliability may have many applications in an engineering context,[185] but in a forensic setting such measures are but a means to an end. To focus directly upon the question of whether specific material has been affected by failure is therefore reasonable in the present context. The crucial question relates to the efficacy of the methodology that is adopted. What may have been envisaged is a possibility that a period of failure free operation surrounding[186] production of the material of interest would signify a reduced likelihood that the material itself was affected by failure.

Here the problem of recognising non-obvious error overshadows this consideration to a significant extent. To refer to an observed failure free period with any certainty is not consistent with the case in which there is uncertainty as to whether specific material has in fact been affected by failure. Apart from this problem, there is another difficulty. It relates to the random nature of the failure process for software. The randomness of individual failures renders the observation of a failure free period insignificant in the context of specific output. It is true that a large number of failures that have been observed for given software might be associated within a single identifiable probability distribution for inter-failure periods. Nothing can, however, be inferred from the observation of a single failure free period.

It is, in the result, appropriate to be concerned that the conditions imposed by the specific computer approach are ineffective to address the issue of reliability. As has been demonstrated, the reasons for this are subtle and the Law Commission touched only upon the periphery of the problem. Even so, the conjecture that it raised was well founded. Lack of efficacy in addressing the issue of reliability and the ancillary

---

185 For example, verification that a product meets a customer's needs, the promotion of efficiency in test efforts and reducing the cost of maintenance: Donnelly M, Everett B, Musa J and Wilson G, "Best Current Practice of SRE [Software Reliability Engineering]" in Lyu M R, *Handbook of Software Reliability Engineering* (Los Alamitos: IEEE Computer Society Press, 1996) 219-254 at 219-220.

186 Or in the case of the South Australian regime, preceding it. Section 59B(2)(d) of the *Evidence Act* 1929 (SA) refers to "a period extending from the time of the introduction of the data to that of the production of the output ..."

question of the likelihood of specific failure is a very significant shortcoming of the specific computer approach.

*Complexity and difficulty*

The contention that there is no need for a specific computer approach is sometimes accompanied by the argument that some expressions of the approach are complex, and/or that compliance with them is difficult.[187] Brown makes the point more clearly than elsewhere, arguing that the "unrealistic complexity of the conditions for admissibility in computer-specific legislation" have contributed to the failure of the relevant statutes.[188] He sees the complex conditions as giving rise to a regime that imposes upon computers "unreasonably high standards of reliability."[189] Tapper voices considerable trepidation about the complexity of the *Civil Evidence Act* 1968 (UK) provisions,[190] but is less concerned about the former s 69 of the *Police and Evidence Act* 1984 (UK).[191]

Criticisms of this kind are directly relevant to the primary goal of rational truth identification in legal fact finding. They question whether what is being asked for in the legislation is really necessary in order to provide an assurance as to the accuracy of the record. In other words, does requiring a proponent of evidence to incur the difficulty of compliance produce a 'better' outcome in terms of the goal of rational truth identification, or does it simply give rise to wasted effort? Considerations of economy and expedition are also relevant to this enquiry, since the complaint is—at least in part—one that compliance with the requirements of the legislation is a task that is in many cases too onerous.

---

187 See for example Brown, Note 4 at 366; Tapper, Note 3 at 395 and 397; Miller C, "Electronic Evidence—Can You Prove The Transaction Took Place?" (1992) 9 *Computer Lawyer* 21-33; Law Commission, Note 89 at paragraph 13.8.

188 Brown, Note 4 at 366.

189 Brown, Note 4 at 366.

190 Tapper, Note 3 at 395-398.

191 Tapper, Note 3 at 402, but compare at 405.

The criticism has two underlying aspects: necessity and efficacy. These are related to the first and third areas of criticism that have been considered here. Some attempt to address the issue of reliability is necessary to meet the requirements of rationality, yet neither of the other two approaches to evidentiary treatment meets these requirements. In other words, there is a need to address the issue of reliability that is not met by the other approaches. The specific computer approach is, however, ineffective to do this. To this extent, the criticism of complexity and difficulty has merit. Complying with the conditions imposed by the approach has little utility because it does not produce information that is useful in the context of evidentiary treatment.

5.2.4.2.2 Overview

A central argument that has been developed in this thesis is that it is not possible to deal with the issue of the reliability of computers on a general or 'collective' basis. The specific computer approach is distinguished from the other two principal approaches to evidentiary treatment because it does not attempt to do this. It may therefore seem strange that the review of the specific computer approach that has been presented here shows that this approach also has substantial shortcomings.

To recognise that reliability is a variable characteristic that cannot be dealt with collectively is a necessary, but not sufficient, prerequisite for the appropriate evidentiary treatment of computer-produced material. What is also required is that any attempt to deal with reliability on a specific basis will itself have an adequate foundation. If an approach imposes conditions upon admissibility that seek to assure reliability, then those conditions must be effective to do this. The conditions that are imposed by the expressions of the computer specific approach that have been reviewed here do not meet this requirement.

In contrast to the other two approaches to evidentiary treatment, a framework for criticism of the specific computer approach was already well established by prior literature. In most cases, the literature expressed criticisms of the approach in

conjunction with the presentation of one of the other two approaches as a preferable alternative. Except in one respect, the criticisms that were made were well founded.[192]

## 5.3 Conclusions

This chapter has examined the manner in which the principal existing approaches to the evidentiary treatment of computer produced material deal with the issue of reliability. It has provided a scheme for the classification for approaches to evidentiary treatment and has identified three principal approaches within that scheme. These have been called the 'substantial equivalence', 'presumptive' and 'specific computer' approaches.

The chapter has, by reference to specific examples or 'expressions', evaluated each approach according to the criteria that an approach to evidentiary treatment should deal with the issue of reliability in a rational manner. In this respect each approach has been found to exhibit shortcomings. The criticisms that have been made have shared a common focus. That focus has been the seemingly insurmountable problem that the reliability of computers cannot be the subject of a general assumption nor, within any practical limits, the object of assessment on a specific basis.

In some cases, expressions of an approach seem largely to disregard the need for evidentiary treatment to have a rational foundation. Expressions of the substantial equivalence approach fall into this category and they do so because the philosophy that underpins this approach exhibits a fundamental flaw. In attempting to characterise computer-produced material as the near equivalent of the product of other processes, the philosophy severely inhibits the prospect that the issue of reliability will be addressed at all, much less in a rational manner.

---

[192] The exception being the criticism that there is no need for a specific computer approach in light of existing, viable alternatives.

In other cases, the criticism draws more heavily upon the complexity and uncertainty that is inherent in the concepts of computer reliability and computer failure. For instance, the subtle nature of the failure process for software and its relationship to overall reliability have been used to illustrate the shortcomings of the specific computer approach. This represents a significant insight because in adopting a specific focus, the approach may otherwise have remedied the principal defects in the other two approaches.

In all cases, the evaluation that has been carried out has drawn heavily upon the findings of chapter four. It is for this reason, if for no other, that what has been concluded about each approach differs from the positions that have been adopted elsewhere. The greatest point of difference is that whilst the prior literature largely champions the substantial equivalence approach, the evaluation that has been undertaken here has found substantial flaws in it. These conclusions suggest that an alternative approach to evidentiary treatment is required. A proposal for such an alternative approach is presented in chapter six.

# 6. An alternative approach to evidentiary treatment

## 6.1 Introduction

Chapter five identified important shortcomings in the principal existing approaches to the evidentiary treatment of computer-produced material. Those shortcomings related to the manner in which the issue of reliability has been dealt with under those approaches, having regard to the objective of rational truth identification that is central to legal fact finding. The purpose of this chapter is to propose an alternative approach to evidentiary treatment that represents an improvement upon the existing approaches.

## 6.2 The problem to be addressed

### 6.2.1 The elements of the problem

The purpose of any approach to evidentiary treatment must be to promote the realisation of the goals of legal fact finding. This is clear from the definition of the term that was given in chapter one. In one sense, this statement defines all that an alternative approach to the evidentiary treatment of computer produced material

should set out to achieve. What the preceding four chapters of the thesis have shown, however, is that the use of computer-produced material as evidence presents three special, conflicting considerations which any approach to evidentiary treatment must address.

The first consideration relates to rationality. There is a need to have a rational foundation for asserting that a source of information will be able to aid the identification of truth. As was demonstrated in chapter two, this is an incident of the dominant rational paradigm under which legal fact finding is pursued in common law jurisdictions. The second consideration is that it is very difficult to quantify or qualify the reliability of computers in a way that would furnish the foundations about probative capacity that are required for the purposes of evidentiary treatment. The need to have a foundation that relates to the issue of reliability is clear from the role that reliability plays as a determinant of the accuracy of computer output. The scope and extent of the difficulties that are involved were described in chapter four.

The third consideration is manifested in the 'real world'. It arises out of the fact that despite the difficulties that are inherent in addressing the issue of reliability in both the engineering and the legal fact finding contexts, individuals, governments and organisations continue to use computers ubiquitously. The potential for computer-produced material to be required to aid the determination of issues in instances of legal fact finding is substantial. More importantly, it is likely to increase, rather than abate. The need to provide an alternative solution to the problems that have been identified in this thesis is not a temporary one.

These considerations generate a particular backdrop against which the goals of legal fact finding have to be pursued in the context of the evidentiary treatment of computer-produced material. More importantly, the considerations produce an obvious conflict. On the one hand there is an aspiration that legal fact finding processes will operate on rational premises. On the other hand, the reliability of computers is a quantity that is central to the accuracy of computer output but it is a quantity that cannot be characterised in a way that is suitable for the purposes of legal fact finding. At the same time the need for legal fact finding to be able to make use of

computer-produced material is a significant and, most likely, a permanent phenomenon.

This conflict is properly seen as tripartite in that it is due to the combined effect of the three elements that have been identified (rationality, uncertainty and ubiquity). A reasonable initial assumption is that the key to reducing or eliminating the conflict lies in the alteration of the conditions under which one or more of these elements are expressed. This may result in the attenuation of the effect of one or more of the conflicting elements. This course is not, however, suggested without recognising that substantial difficulties may be faced in this regard. For one or more elements, it may be neither possible nor palatable to alter the conditions under which they are manifested.

This notwithstanding, due consideration of a sufficient range of options may reveal a greater scope for change than initially appears to be the case. The exploration of this possibility for solving the problem that has been identified here is made more attractive by the tripartite nature of the problem under consideration. A sufficient change in the effect of just one of the elements might resolve the problem to a large degree.

Departure from the condition that legal fact finding pursue truth in a rational manner would reduce or remove the significance of the reality that it is difficult to quantify or qualify the reliability of computers. In these circumstances, the fact that there is an ongoing need for legal fact finding to be able to make use of computer of computer-produced material would present no special difficulties. Curtailing the use of computers would also greatly reduce the problem. If this were done, there would still be a need for the use of computer-produced material in legal fact finding and a tension between the other two factors, but the size of the problem would be much smaller. Similarly, the provision of a more robust foundation for dealing with the issue of reliability would itself reduce the impact of the problem.

These considerations give rise to the need to ask which (if any) of the three factors that have been identified as ingredients of the problem are most amenable to change? This question is addressed in the following section.

### 6.2.2 Altering the influence of the elements of the problem

*Rationality in truth finding*

The identification of truth by rational means is an 'institutional' value that is central to legal fact finding. It may be argued that departing from this goal would represent a fundamental shift in the goals and methodology of legal fact finding. On this view, departure from the ideal of rationality and the associated need to make rational decisions about evidentiary treatment would be regarded as unacceptable. A contrary view is that what is more important is the *extent* to which there is a departure from the ideal of rationality.

Under the latter view, the use of unsupported assumptions about the reliability of computers would be considered to be something quite distinct from a return to ancient superstitious practices that characterised the 'irrational' modes of trial.[1] Such a course might be regarded as a merely 'technical' or 'formal' departure from standards of rationality. It might further be argued in this view that the relevant motivation is merely to facilitate the use of material that has come into existence because computers are widely used and relied upon. This, it may be said, would amount to no more than recognising the "realities of modern business methods."[2] It might further be said that it is very difficult to characterise an approach which is motivated in this way as 'irrational'.

Whilst this latter argument is superficially attractive, it ultimately exhibits three important shortcomings. First, it attempts to appeal to a sentiment that there should be a preference for 'substance over form'. This would be appropriate if the underlying

---

1 See chapter two, section 2.2.1.

'truth' was that the risk of inaccuracy in computer output due to failure was in fact marginally low. Yet, as was demonstrated in chapter four, this assertion cannot be made. It is not the case that there is some supervening shared or 'common' knowledge about the state of the underlying physical processes in question that simply cannot be demonstrated in a rational way. All that can be demonstrated is that there is genuine uncertainty about reliability which is operative on both a general and a specific level. This is the only underlying 'truth'. In these circumstances an appeal to substance over form must fail.

Second, the argument provides no clear indication of the basis for decision making that ought to apply to the formulation of approaches to evidentiary treatment. What is the mechanism that would operate in place of a logically sustained foundation for a given approach? If a purely arbitrary position is to be adopted, then how is a particular position to be selected from the range of positions that might be adopted? It can be assumed arbitrarily that computers are generally reliable, but it can also as easily be assumed on the same basis that computers are generally *not* reliable. The departure from a rational framework renders this kind of decision making difficult or impossible to undertake.

The third reason why even a less than absolute departure from rationality is unappealing is that it exhibits a certain circularity. Departing from a rationalist ideal amounts to altering the goals of an undertaking merely to suit the exigencies of the methods that are available to attain that goal. The goal of legal fact finding would be altered from the identification of truth by rational means to the identification of truth by some other means. If this is a valid step to take, then it invites the possibility for dispensation with many aspects of legal fact finding that are difficult, time consuming and expensive.

It has been made clear in this thesis that computers are unique in terms of the manner in which they operate and the ubiquity of their use. This notwithstanding, the argument that evidentiary treatment of computer-produced material should be

---

2 *King v State ex rel Murdock Acceptance Corporation* 222 So. 2d 393, 397 (Miss, 1969).

undertaken according to different standards does not make clear if there are to be any boundaries beyond which those different standards should not be applied. As was observed in chapter five, there is a range of complex material in respect of which various dispensations of proof might be superficially attractive. If the requirements of rationality are to be relaxed or discarded for computer-produced material, why should they not be relaxed or discarded for a range of other material?

The considerations that have been canvassed make it clear that rationality in truth finding is not an element that is amenable to change for the purposes of resolving the problem that is of present interest.

*Ubiquity of computer use*

The extent to, and purposes for, which computers are used is a matter that is strictly outside the scope of the matters that might be controlled by the procedural law. In short, no change to procedural law can be expected to change the extent to which use is made of computers for the processing, handling and storage of information. All that may be controlled is the extent to which computer-produced material is *recognised* by the law of evidence.

One possible course would be to implement a general prohibition upon the use of computer-produced material for the purposes of legal fact finding. This would render the ubiquity of computer use irrelevant for the purposes of evidentiary treatment. Exclusion of the material would remove the need to find a solution to the problem that is of present interest. Despite the completeness of the 'solution' that is offered, this course is plainly unattractive.

It has the potential to exclude a very wide class of information which, in large part, may not exist in any other form. Whilst it may preserve or promote the *rationality* of truth identification by excluding from consideration material which is not amenable to appropriate scrutiny, it has the potential to detract substantially from the identification of truth as an end in itself. If a repository of information provides the only support for

a particular contention, then access to that repository is, subject to the need to consider the question of accuracy, consistent with the pursuit of truth.

A less drastic alternative could involve the restriction of admissibility to material that, say, has been produced by software which has been run on a given kind of hardware on a very widespread basis. The rationale for this approach would be that established patterns of use by a large number of entities ought to provide some assurance of the reliability of given software when executed on given hardware.[3] An approach similar to this was proposed when the issue of computer output as evidence first arose in the United States but did not emerge as a paradigm for admissibility.[4]

It is true that a 'widespread and established use' criterion of the kind proposed here carries with it no assurance of reliability. Yet it appears to go some way to remove some material from the province of general uncertainty about reliability that otherwise obtains for computer-produced material generally. Whilst this capacity to distinguish particular material may seem initially to be appealing, there are, however, two difficulties with the proposed strategy. First, it requires that quantitative standards of use be established and be capable of being applied. Not only must it be determined what will amount to 'widespread use', but information about the extent of use of particular software will have to be amassed before it can be determined that such software meets the criterion.

Second, it may provide a false sense of security. It might be thought that there is a relationship between ubiquity of use and reliability, but a simple example shows that this may not be the case. The example relates to the software product known as 'Microsoft Excel'. It is notoriously known that Microsoft Excel is a very widely used product. It performs a variety of mathematical calculations in a 'spreadsheet' format. Amongst its capabilities are a wide range of statistical operations. Despite the extent

---

[3] The reliability that is ultimately of interest is, as was observed in chapter four, the reliability of the combination of particular hardware and software.

[4] One of the conditions of admissibility stipulated in the early decision in *King v State ex rel Murdock Acceptance Corporation* 222 So. 2d 393 (Miss, 1969) was that "the electronic computer equipment is recognized as standard equipment": 222 So. 2d 393, 398. That condition was not incorporated into the later decisions under the Federal Rules of Evidence.

to which it is used, studies of various aspects of its statistical functionality have demonstrated that this software contains various faults.[5] Although this is an isolated case, it demonstrates that even very widespread use is not an infallible indicator of reliability.

Based upon the considerations of the prerequisites for implementation and questions about efficacy, it can be said that even a partial prohibition upon the use of computer-produced material that operates on the basis of a criterion of 'widespread use' does not present an attractive solution to the problem of interest.

*Uncertainty about reliability*

The causes of the uncertainty that attaches to the reliability of computers were identified in chapter four. They arose out of the variability of the factors that are at play in the design and operation of computers. In particular, much is due to factors that govern the process of creating software. Altering such factors directly must be considered to be outside the realm of possibility for present purposes.[6] What may instead be possible is to reduce the uncertainty that arises when a computer is used to produce information bearing material.

The relevant focus in this context is not upon the operation of the computer as a system of unknown reliability, but rather upon the *environment* where the computer is used to accomplish a given task or function. What can be asked about such an environment is whether the computer is 'trusted' to produce accurate information, or

---

5 Inaccuracies in the calculations presented by Microsoft Excel are reported in Knüsel L, "On the Accuracy of Statistical Distributions in Microsoft Excel 97" (1998) 26 *Computational Statistics & Data Analysis* 375-377; McCullough B D and Wilson Berry, "On the Accuracy of Statistical Procedures in Microsoft Excel 97" (1999) 31 *Computational Statistics & Data Analysis* 27-37. It is of note that the inaccuracies reported by McCullough and Berry were detected by them for a later version of Excel: McCullough B D and Wilson Berry, "On the Accuracy of Statistical Procedures in Microsoft Excel 2000 And Excel XP" (2002) 40 *Computational Statistics & Data Analysis* 713-721. McCullough and Berry observed that "some users might assume that Microsoft would not release a new version without fixing known errors" but went on to report that "[a]ll the Excel 97 errors reported in [their 1999 study] exist in Excel 2000." (at 714).

6 It well may be a matter that is in fact not feasible from an engineering standpoint. As was discussed in chapter four, research in the field of software reliability has been ongoing for just over thirty years. Despite this, the problems that were outlined in that chapter still exist today.

whether instead there is some degree of verification or 'cross checking' of the accuracy of its output. There may, for instance, be a degree of human verification of the relevant information. There may be a system that is truly independent of the computer which can verify the accuracy of the output in some way. Alternatively, the computer may be trusted completely in a given environment, such that users within that environment will have no means for the detection of inaccuracy in the output of the computer.[7]

These considerations relate ultimately to the level of redundancy[8] that is implemented in the environment in which the computer is used. Redundancy is achieved by providing mechanisms or components that do not increase the functional capacity of the primary system of interest but operate merely to prevent or to mitigate failure in that system.[9] A redundant mechanism may involve, for instance, manual verification of output by a person with knowledge of, or at least familiarity with, the expected output. Other methods of verification are also possible, such as comparison of the output of interest with the output from a parallel computer system.[10]

In such environments, reliance upon the computer to perform a given expected function with given input is not absolute. It may in fact be quite minimal. This is because the correct operation of the computer is not left to chance. It is instead verified by the redundant mechanism. This being the case, the *effect* of uncertainty about the reliability of the computer in question will be significantly reduced. As a

---

7 Except in the case of obvious error.

8 The application of redundancy in the present context was inspired by the work of Claude Shannon (1916-2001) on information theory. See generally: Shannon C E "A Mathematical Theory of Communication" (1948) 27 *Bell System Technical Journal* 379-423 and 623-656 reprinted with minor revisions in Weaver W and Shannon C E, *The Mathematical Theory of Communication* (Urbana, Illinois: University of Illinois Press, 1949).

9 For Shannon's description of redundancy in a communication system, see Weaver and Shannon, Note 8 at 75

10 That is, another computer to which the identical input is provided. The output of this computer is compared with the output of the computer of interest to verify correctness of operation. This kind of mechanism would normally require that software of a different design be used to facilitate verification, since an identical copy of the software will have the same fault sensitivities for the same inputs. For a further discussion see McAllister D F and Vouk M A, "Fault-Tolerant Software Reliability Engineering" in Lyu M R, *Handbook of Software Reliability Engineering* (Los Alamitos: IEEE Computer Society Press, 1996) 567-614 at 574-575.

strategy of uncertainty reduction, the use of information about redundancy appears to hold promise.

In particular, the only additional information that this strategy requires is an account of the environment in which the material of interest was produced. This requirement can be contrasted with the (onerous) need to obtain representative failure data, which would be required in order to make a direct assessment about the reliability of software or hardware. It can also be contrasted with the difficulties that would apply if a criterion of 'widespread and established use' were to be employed.

The strategy is also intuitively appealing. For given computer output, concerns about accuracy must, as was established in chapter one, be resolved by reference to the circumstances in which the output was created. This involves an enquiry about, among other things, the reliability of the process of production. Given that ascertaining reliability is difficult, the next most pertinent question is: if particular computer output contains inaccuracies, what indication of those inaccuracies would have been given in the environment in which the output was produced? The strategy of reducing uncertainty about reliability by examining the presence and effect of redundancy measures asks precisely this question. This strategy is the foundation for the alternative approach to evidentiary treatment that is proposed in this chapter.

## 6.3 An alternative approach

Of the possibilities that were considered in section 6.2, the most promising appears to lie in an attempt to reduce the uncertainty that attaches to the issue of computer reliability. The focus of the approach that is presented here is the extent to which inaccuracy in computer output would be indicated in the environment in which the output is created. In this way, the approach emphasises the extent to which there has been *reliance* upon the computer alone to, in effect, verify the accuracy of its own product. As was mentioned in section 6.2.2, there is a potential for significant variation in the extent of such reliance. The approach that is proposed here uses this

potential as a basis for determining the particular treatment that should be given to particular material.

### 6.3.1 The extent of reliance

The potential for variability in reliance is a loose analogue of the variability in reliability that must be assumed to apply for computers generally. The critical difference is that whereas the reliability that is exhibited by a given computer is largely unplanned,[11] the extent to which that computer is relied upon in the environment in which it is used is the direct result of conscious action. The choice to incorporate redundant mechanisms into the environment in which a computer is used is a deliberate one.[12] It is therefore far easier to establish the extent to which a computer has been *relied upon*, than it is to establish the extent to which a computer is *reliable.*

Establishing the extent of reliance would likely be limited to the provision of qualitative information about the presence of redundant mechanisms. There is of course scope for variability in the extent of the information that can be provided. The underlying question to which such information would relate will, however, always be the same. It is an enquiry as to the indication that would be given if, in producing given material, the computer failed to operate in the manner expected. In other words, to what extent is the computer relied upon to the exclusion of any other indicator of accuracy?

### 6.3.2 A composite 'system'

The approach that has been described to this point might be thought to present difficulties of recursion. Not every measure that might be implemented in order to

---

[11] In the sense that perfect reliability is presumably hoped for, but, due to unintended errors, faults and other mishaps, may not actually be achieved.

[12] Albeit that it may be subject to limiting factors of effort and cost.

achieve redundancy will itself be perfectly reliable. A reasonable inference, however, is that the success of what has been proposed is contingent upon the effectiveness of any redundant mechanisms that exist in the environment of interest. Yet the question of assessing the reliability of the redundant mechanism may be considerably more complex than the question of assessing the reliability of the computer itself. This will be the case particularly when the redundant mechanism involves a process that is not rigorously defined or which depends heavily upon human involvement.[13] An example of such a situation is one where the redundant mechanism involves the review of output on an informal basis by a person who is thought to be capable of detecting error in that output because they have some degree of familiarity with what is expected. In other words, where the person in question is merely checking the output to ascertain if it 'looks right'.

The problem of assuring the reliability of the redundant mechanism(s) does not discredit the proposal because failure of the redundant mechanism is only relevant when it coincides with a failure of the primary system (the computer). If there are multiple redundant mechanisms that are each capable of detecting error independently, then all must fail with the primary system before a problem will arise. More formally, the computer and each independent redundant mechanism are arranged in 'parallel' from a reliability perspective. Unlike hardware and software, which are components arranged in 'series', no 'weak link' scenario arises.[14] Instead, the reliability of the entire collection is enhanced by the addition of each extra component.[15]

More importantly, it is possible to demonstrate this to a fact finder in a way that permits evaluation of the level of reliance. A fact finder can be easily instructed that a greater number of independent redundant mechanisms equates to lower reliance upon the primary system or indeed upon any single redundant mechanism. This is due to

---

[13] Although it may be observed from the discussion in chapter three that human involvement in software design and production is largely responsible for the underlying issue of reliability that arises in the first place.

[14] The reliability of components in series and parallel was discussed in chapter three.

the fact that, apart from its theoretical soundness, the proposition is intuitively appealing on a general level. The presentation of this kind of information would seem to be far less complicated than an attempt to quantify the reliability of a computer directly. It is possible only because the computer and the redundant mechanisms are being treated as components that make up a composite 'system'.

### 6.3.3 A basis for distinguishing between material

An account of the extent of reliance upon a computer alone can be used as a mechanism for distinguishing material for which there ought to be reservations about accuracy, from material for which no such reservations are necessary. Where there is a very high degree of redundancy in a given environment, there is a much greater prospect that failures which would otherwise produce non-obvious inaccuracies will be detected. Where there is no redundancy at all, there is no prospect that such failures will be detected. Although no guarantee of accuracy is provided, at some point between these two extremes the risk of undetected failure might be seen to be sufficiently low that use of the material in question is warranted. This proposition provides a basis upon which conditions for the use of computer-produced material might be formulated and, in turn, implemented via rules about admissibility. This in turn forms a foundation for expressing the proposed alternative approach as an approach to the evidentiary treatment of computer-produced material.

### 6.3.4 An expression of the alternative approach

#### 6.3.4.1 Reliability

The essence of the proposed alternative approach to evidentiary treatment is the use of information about the extent to which the computer was relied upon to the exclusion

---

[15] Ramakumar R, *Engineering Reliability: Fundamentals and Applications* (Englewood Cliffs, New Jersey: Prentice-Hall, 1993) 150. Even the addition of a component of zero reliability would merely leave the overall reliability of the collection unchanged.

of any other indicator of accuracy or error when given material was produced. Two matters must be determined in connection with any expression of the proposed approach. First, how will this information be used to establish conditions for the admissibility of the material in question? Second, which party will be responsible for furnishing this information?

*Using information about reliance to establish conditions of admissibility*

What is proposed is that information about the extent of reliance upon a computer will be used as a mechanism to determine admissibility; it will not be used merely as an aid to the assessment of weight. This course is proposed because of the limitations that are inherent in the process of allocating weight. In particular, as was demonstrated in chapter five,[16] this process is unsuited to the resolution of questions about the reliability of a given computer.

A standard for determining admissibility that is based upon information about reliance would involve the following question. Is the extent of reliance upon the proper functioning of the computer alone so great that the material should be excluded? This standard of admissibility clearly departs substantially from prevailing approaches to the evidentiary treatment of computer produced material. It is, however, a necessary departure. If material is excluded under the proposed approach, it will only be because the risk of failure cannot be quantified within acceptable limits. These will not be cases in which the risk is 'unknown, but probably small'. They will be cases in which there is no means by which the risk can be characterised in any way, other than to stipulate that it is entirely unknown.

The importance of this point would be diminished if the goals of legal fact finding were ambivalent about the identification of truth, but they are not. Under the adversarial model, legal fact finding is sometimes characterised as an impartial and largely passive process.[17] *Inter parties* this is undoubtedly the case,[18] but such

---

16 See chapter five, section 5.2.1.3.

17 See for example Damaska M R, *Evidence Law Adrift* (New Haven: Yale University Press, 1997) 74.

equanimity is not a universally applicable feature of legal fact finding. When it comes to a choice between outcomes that reflect the truth about the matters in dispute and outcomes that are inconsistent with truth or those that merely coincide with it by chance, legal fact finding is decidedly partisan and it is decidedly biased. The underlying concern "to get at the truth"[19] is still important, even in the contemporary context. Yet it is only when this important aspect of the 'character' of legal fact finding is reiterated that the need to impose an exclusionary mechanism of the kind mentioned here can be appreciated.

At the same time, the argument that there should be flexibility in the mechanisms of evidence law must also be given due consideration. Although it may lead in some cases to a degree of inconsistency of application, such consideration requires that a standard for admissibility in the present context must be couched in terms of what is reasonable. The standard for admissibility that is proposed here is as follows.

It should be demonstrated that:

(a) some mechanism(s) of redundancy (however formulated and implemented) was or were utilised in connection with the production of particular material in the setting in which it was produced; and that

(b) it is reasonably likely that any error(s) in the operation of that computer that affected the accuracy of information contained in that material would have been detected by such mechanism(s).

In addition to providing a degree of flexibility, the conditions that have been proposed also stop short of insisting upon the positive demonstration of accuracy. This reduces the risk that probative material will be excluded because it cannot be shown to be accurate and it is consistent with the role that rules about admissibility should fulfill.

---

[18] Subject to a recent 'trend' toward greater emphasis on judicial involvement in the 'case management' of litigation.

[19] Wigmore J H, *A Students' Textbook of the Law of Evidence* (Brooklyn: Foundation Press, 1935) 5.

What results is not a guarantee of accuracy, but a basis for inferring that computer-produced material that is admitted for use in a given instance of legal fact finding will be more likely than not to aid the identification of truth. Of greater importance still, the basis of inference is one that has a rational foundation.

*Responsibility for furnishing information about reliance*

It is implicit in the formulation of the standard for admissibility that has been proposed that the party offering the material in question will bear responsibility for providing information about the extent of reliance. However, this need not be the position. It would for instance be a simple matter to reverse the relevant burden of proof. This may be done (using the conditions formulated above) by stipulating that given material is to be admitted unless it is shown that:

> (a) it is not reasonably likely that any error(s) in the operation of a computer that affected the accuracy of information contained in the material in question would have been detected by any mechanism of redundancy (however formulated and implemented) that was utilised in connection with the production of that material, in the setting in which it was produced.

It is therefore possible to articulate the relevant standard of admissibility so as to place the burden upon either the proponent of material, or the opposite party. The principal argument against placing the burden upon the proponent of computer-produced material which has been advanced is that of the Australian Law Reform Commission. This argument was considered in chapter five, but it is appropriately reiterated here. The Commission argued that

> [t]o require extensive proof, on each occasion, of the reliability [sic – accuracy] of ... computer records is to place a costly burden upon the party seeking to tender the evidence, to give the opposing party a substantial tactical weapon and to add to the work of the courts. In many cases there will be no bona fide issue as to the accuracy of the record. It is more efficient to leave

> the party against whom the evidence is led to raise any queries and to make any challenges it may have.[20]

This argument was advanced in respect of proof of the 'reliability' of material by the proponent of evidence. It focused upon the possibility of insistence upon proof as an illegitimate tactic in the absence of a bona fide dispute about the accuracy of the relevant material. It is, however, unpersuasive in the context of the standard that has been proposed here. It is more apposite to the case in which the proponent faces the task of demonstrating the reliability of the computer itself. For the reasons given in chapter four, such a task is undoubtedly difficult and in many cases will be impossible. The task of providing an account of the existence of redundant mechanisms in the environment in which the computer has been operated in fact avoids the complexities that are associated with the proof of reliability. All that is required under the approach proposed here is to describe the measures that have been put in place in a particular setting.

More generally, the Commission's argument is an argument against the imposition of *any* condition of admissibility being made the responsibility of the proponent of material for fear that there will be insistence that it be complied with. It hardly solves the general problem that an adversarial mode of trial presents, namely that the selection of evidence is necessarily a partisan process in which the cooperation of the parties cannot (and should not) be assumed. If a solution to this problem is desired, then it must be pursued at a more fundamental level at which the possibility of structural change to the adversary system can be explored.

6.3.4.2 Input information

The conditions that were proposed in section 6.3.4.1 are capable only of addressing the issue of reliability, and the associated concern of the possibility of failure that is

---

[20] Australian Law Reform Commission, Report 26, *Evidence (Interim)* (Canberra: Australian Government Publishing Service, 1985) Volume 1, paragraph 705.

not obvious on the face of the relevant computer output. They were not designed to address the issue of the possibility of inaccuracy of input information.

Although the question of evaluation of the treatment of input information is outside the scope of this thesis, it is useful to make brief reference to it in the context of a proposal for an alternative approach to the evidentiary treatment of computer-produced material. This is because the proposed approach involves the imposition of additional standards of admissibility, yet the conditions nominated in section 6.3.4.1 do not by themselves describe an optimal alternative approach to evidentiary treatment.

An answer to the question of the treatment of input information may in fact lie in the substantial equivalence approach. This may initially appear to be paradoxical, since that approach was heavily criticised in chapter five for its failure properly to appreciate and to deal with the issue of reliability. An important limitation upon those criticisms is that they said nothing about the way in which the substantial equivalence approach deals with the possibility of inaccuracy of input information. It is in this respect that the approach imposes conditions that may represent an appropriate basis for dealing with such information.

The point at which input information is provided to a computer is the point up to which the notion of equivalence is appropriate. Whilst it was incorrect for Brown to assert that "[w]riting notes on a piece of paper is no different to storing them in a computer",[21] it can be said that writing notes is no different to providing information to a computer *as input*. The relevant point of distinction, which has been the focus of attention in this thesis, is in what occurs subsequently. It may therefore be appropriate that input (as opposed to output) be treated in the same manner as it would have been treated had it been offered directly as evidence.

An additional condition that may appropriately form part of the alternative approach that has been proposed here is that any input information that was utilised in the

---

[21] Brown R A, *Documentary Evidence in Australia* (Sydney: Law Book, 2nd ed, 1996) 355.

production of material should be admissible in its own right. If, for instance, the information amounts to hearsay, then it would be subject to the rule against hearsay (and the exceptions to that rule). This additional condition (upon the admissibility of the output) can therefore be formulated as follows: the information upon which the relevant output is based, or from which it has been derived, would have been admissible if, instead of having been provided to a computer, it had been committed to writing or communicated orally.[22] What would be effected by this condition is a treatment of input information in a manner that reflects the form that such information would have taken, had it not been supplied to a computer.

## 6.4 Conclusions

This chapter has presented an alternative approach to the evidentiary treatment of computer-produced material that attempts to address the problems that were identified in chapter five. The proposed approach confronts the problem of reliability not by seeking to assess reliability directly, but rather by attempting to qualify the risk that that uncertainty presents.

The device used for this qualification is the extent to which the computer is relied upon to the exclusion of redundant or verifying mechanisms. Such mechanisms may be mechanical or human in nature, but will operate to provide some level of verification that a failure in the operation of the computer has not occurred. The proposed approach also deals with the question of accuracy of input. It recognises that this is a respect in which notions of equivalence have a sound basis. Information that has been supplied to a computer should be treated in exactly the same way as it would be treated if offered directly as evidence.

Although the approach cannot be said to represent a 'magic' cure to the problem that has been considered in this thesis, the result that it achieves is significant. Unlike other approaches to evidentiary treatment, the proposed approach is able to address

---

[22] Other than as witness testimony in the proceedings.

the issue of reliability in a rational manner. What the approach offers is a means for dealing with a difficult problem, namely how to deal with an uncertainty that cannot directly be resolved. Whilst other approaches overlook the problem, or otherwise fail adequately to deal with it, the proposed approach confronts it directly.

The approach undoubtedly owes much to the specialised analysis of computer-produced material that was presented in chapters three and four. The need to have reference to such information is of course reflected in the nature of the material in question. However desirable it may be to meet the problems of evidentiary treatment with solutions that are medium neutral and generic, it is not—as has plainly been demonstrated—possible to do this in the instant case. Despite this, it is noteworthy that the aim of the proposed approach, and the outcome that it produces, are not special, nor are they technology specific. In the ultimate analysis, the proposed approach will merely function as any approach to the evidentiary treatment of any material *should* function. It will operate to aid and to promote the primary goal of legal fact finding.

# 7. Conclusions

## 7.1 Overview

This thesis commenced with the proposition that legal fact finding is an important process because it is fundamental to the application and enforcement of law. What has also been demonstrated to be important are the goals that are prescribed for this process, and the manner in which those goals are pursued. The evidentiary treatment of computer-produced material is but one focus for the pursuit of goals of legal fact finding. Computer-produced material presents to legal fact finding unique characteristics and unique challenges. These are due to the properties of the process by which this material is produced and the nature of the elements that are at play within that process.

What this thesis has established is that the pursuit of the goals of legal fact finding within this particular context requires an approach that responds to the unique characteristics of computer-produced material. It may be very convenient to overlook or to ignore the factors that distinguish computer-produced material and to treat it as the equivalent of other kinds of material. As has been described, this is the central premise for a predominant approach to the evidentiary treatment of computer-produced material that is applied in the United States, the United Kingdom and in

Australia. Yet to do this is inconsistent with the central goal of legal fact finding, the identification of truth by rational means.

An important characteristic of computer-produced material that has been almost entirely overlooked by existing approaches to the evidentiary treatment of computer-produced material is the fact that this material is the artefact of a process of information transformation. The particular properties that computer-produced material possesses are those that the process imparts to it. Those properties are, in turn, the result of the function that the process is designed to implement. The possibility that the process will not implement the function that is intended gives rise to the concept of reliability. More than this, it gives rise to a need to address the issue that is presented by the possibility that reliability may be less than perfect. This is particularly the case when the property that is of interest is the accuracy of the information that is contained in given material.

Despite the existence of a clear need to address an issue of reliability, prevailing attitudes to the evidentiary treatment of computer-produced material do very little to meet this need. When attempts have been made to address this issue, they have failed to do so in a rational manner. The reasons for this are varied, but a common theme relates to a failure to locate and to articulate sustainable foundations for assumptions that are made about the reliability of computers, and about how that reliability might be assessed. These are, of course, matters that are attended by complexity, difficulty and uncertainty. Yet the adoption of a rational methodology requires that there be a sustainable basis for the choices that are made about how legal fact finding is to be undertaken.

Legal fact finding values the identification of a physical, empirical truth and this is what necessitates the need for evidentiary treatment to identify and to operate on the basis of physical realities that underlie subjects such as the reliability of computers. There is no warrant here for the substitution of a separate 'legal' conception of how—and how well—computers operate. This is why the articulation of unqualified views to the effect that the reliability of computers in general meets or exceeds some high minimum level have no place in the context of evidentiary treatment. Standing back

from the intricacies which have been explored in this thesis, it may seem that such views are neither incredible nor far-fetched. They may in fact appear superficially to be quite attractive. Their adoption would lead ultimately to a much simpler solution to the problem that is presented by the potential for the use of computer-produced material in legal fact finding. Yet they are in form, and in substance, strictly arbitrary. As such, their application is wholly unsuited to an environment, such as legal fact finding, in which truth identification by *rational* means is a principal goal.

The dictates of a rational paradigm do not, however, extend merely to the exclusion of arbitrary choices about how legal fact finding is to be undertaken and how particular material is to be received. They require also that the attempt that is made to engage with the underlying physical reality be effective. The issue of efficacy is an additional respect in which the reliability of computers presents difficulties. This is demonstrated by shortcomings in the expressions of the 'specific computer' approach that were reviewed in the thesis. To require that a computer of interest exhibit a failure free period of operation is of course a considerable departure from the assumptions that other approaches to evidentiary treatment have adopted. Unfortunately, it is not a measure which can be said to be *effective* to deal with the issue of computer reliability.

The failure of existing approaches to the evidentiary treatment of computer-produced material suggests the need for an alternative. The search for such an alternative requires in turn that there be a more explicit recognition of the factors which contribute to the problem that is presented by the potential for the use of computer-produced material in legal fact finding. Indeed it is plainly a failure fully to appreciate the extent of the problem that has bolstered support for measures such as the substantial equivalence approach. This approach can be portrayed as attractive, appropriate and reasonable only under conditions that exclude any thorough examination of the problem that it is seeking to address.

The examination that has been conducted in this thesis shows that this problem is substantial. It involves knowledge from fields of research that offer no easy solutions to the dilemmas that they describe and characterise. In many respects, the

contribution that the field of engineering can make in the legal environment is to do no more than indicate the magnitude of the problem that the issue of computer reliability in fact presents. It is unsurprising that the search for an alternative approach to the evidentiary treatment of computer-produced material yields no 'magic' cures. Despite this, it is possible to improve upon the existing approaches in a manner that is consistent with the goal of rational truth identification. The result offers no guarantees of accuracy, either in the information that may be admitted for use under such approach or in the indirect assessment of reliability that is implemented by the approach. The alternative that has been presented nevertheless realises significant improvements of efficacy and rationality over the existing approaches.

What then is the essence of the contribution that this thesis makes? The central argument that was identified at the outset was that the possibility of inaccuracies in computer output that are due to operational deficiencies is a matter that has to be addressed in the context of evidentiary treatment. It was also asserted that principal existing approaches to evidentiary treatment fail to do this. These arguments have clearly been established. In the course of addressing these arguments, the thesis has gone further. It has revealed that the problem that the use of computer-produced material in legal fact finding presents is not a simple one. It possesses considerable depth and complexity. It is not, as the advocates of the substantial equivalence approach may contend, one that is easily addressed. Indeed, to this problem the thesis presents no simple answers.

Yet the accurate characterisation of this central problem as one that is difficult rather than straightforward serves an important purpose in its own right. It requires those who are concerned with evidentiary treatment to pose the difficult questions, and it prevents reliance by them upon enquiries and methods that are less demanding. This should not be regarded as particularly extraordinary since the ascertainment of facts for the purposes of applying law and attaching legal obligation and legal liability is itself a difficult undertaking.

The ideal that the truth should be identified and that this should be done rationally amplifies the difficulties that will inevitably be encountered in the execution of any fact finding endeavour. Asking questions, irrespective of the difficulty that this entails, is fundamental to rational enquiry. The essence of the contribution which this thesis makes is to identify the questions which must, of necessity, be asked when the evidentiary treatment of computer-produced material is being considered. In doing this it provides to the process of legal fact finding a fundamentally important capability in connection with this material: the capacity to identify and to pursue the kinds of enquiries that are essential to the identification of truth by *rational* means.

## 7.2 Applications of the research

A direct application of the findings of this research would involve changes to the law of evidence to remedy the shortcomings which are exhibited by the existing approaches to the evidentiary treatment of computer-produced material. The alternative approach that has been presented is a template for such change.

Beyond this, the model for the evaluation of evidence law that has been presented also has scope for application. It emphasises the evaluation of the effectiveness of the 'evidentiary treatment' of material against the goals of legal fact finding. This is in contrast to the orthodox methodology, which relies upon the evaluation of rules of evidence in a more isolated context. Computer-produced material is but one example of a kind of material that has the potential to engage a number of different rules of evidence.

Many kinds of 'technical' material, for instance, have the capacity to infringe the rule against opinions. As such, they may be admitted as evidence under the 'expertise' exception to that rule. In some cases this will be a proper course, but in others it may represent the unnecessary expenditure of time and resources. In short, the fact that material happens to be an opinion may not be the best determinant of whether it ought to be dealt with under the rule against opinion and the exceptions to that rule.

In most jurisdictions, rules of evidence are not constitutionally entrenched and may be altered as easily as any other statutory provision or rule of common law. At a time at which the efficacy and desirability of long standing rules such as the rule against hearsay is the subject of sustained questioning there is little justification to see any rule of evidence as an end in itself. Ultimately, there is greater appeal in seeking to establish a more direct connection between the goals of legal fact finding and the material which is to be used, or excluded from use, in the pursuit of those goals. The model of evaluation that has been employed here may be a means by which this outcome might be realised.

A further application of this research may lie in the analysis, for the purposes of evidentiary treatment, of specific information technologies. The thesis dealt with computers via a single information transformation model and made only limited references to specific technologies and programming languages. It is conceivable that in certain cases a more detailed analysis of a particular information technology or application may be required.

It has been shown that legal and law reform literature is yet to embrace a consistent, well defined and useful system of terminology and conceptualisation that can deal with information technologies. Yet the empiricist concerns of legal fact finding demand that any analysis that is undertaken should conform to reality. This ultimately requires that there be a degree of conformity with the principles and concepts that are employed in the engineering fields. The extent to which those principles and concepts are accessible can, however, be limited. This is especially true of orthodox legal environments for which an existing appreciation of these matters cannot be assumed.

The thesis presented an account of a number of concepts that may have application to further technology specific analyses. These included, in addition to the information transformation conceptualisation, the notions of faults, failures and reliability and the roles, design and operation of hardware and software. These matters have the potential to be useful as a template for analysis in contexts in which the subject of interest is the accuracy of information that has been produced by processes that have more specific characteristics than the general process that has been considered here.

A more general application of this research may be to shape attitudes to the way in which new technology is characterised for the purposes of evidence law. It is an often expressed view that law should keep up with and adapt to new technology. This may be true in a broad sense. Yet as this research clearly demonstrates, the adaptation of evidence law to technological change requires much more than an uncritical acceptance of modern innovation. It requires a considered appreciation of any given technology not just for its strengths, but for its shortcomings as well.

# BIBLIOGRAPHY

## BOOKS AND CHAPTERS IN BOOKS


Abbate J, *Inventing the Internet* (Cambridge, Massachusetts: MIT Press, 1999).

Ayer A J, *The Problem of Knowledge* (London: Penguin, 1956).

Baer J L, *Computer Systems Architecture* (Rockville, Maryland: Computer Science Press, 1980).

Biermann A W, *Great Ideas in Computer Science: A Gentle Introduction* (Cambridge, Massachusetts: MIT Press, 2nd ed, 1997).

Biggerstaff T J and Perlis A J (eds), *Software Reusability Volume 1: Concepts and Models* (Reading, Massachusetts: ACM Press, 1989).

Bobrow L S, *Fundamentals of Electrical Engineering* (New York: Oxford University Press, 2nd ed, 1996).

Boehm B W, *Software Engineering Economics* (Englewood Cliffs, New Jersey: Prentice-Hall, 1981).

Boyce J C, *Digital Logic and Switching Circuits: Operation and Analysis* (Englewood Cliffs, New Jersey: Prentice-Hall, 1975).

Broom H, *A selection of Legal Maxims: Classified and Illustrated* (London: Sweet & Maxwell, 8th ed, 1911).

Brown R A, *Documentary Evidence in Australia* (Sydney: Law Book Company, 2nd ed, 1996).

Brown T A, *Genetics: A Molecular Approach* (London: Chapman & Hall, 3rd ed, 1998).

Cleary W (ed), *McCormick's Handbook of the Law of Evidence* (St Paul, Minnesota: West, 2nd ed, 1972).

Clements A, *The Principles of Computer Hardware* (Oxford: Oxford University Press, 3rd ed, 2000).

Cohen J L, *An Introduction to the Philosophy of Induction and Probability* (Oxford: Clarendon, 1989).

Cohen J L, *The Probable and the Provable* (Oxford: Clarendon, 1977).

Colson, F H (ed) *Cicero: Pro Milone* (Bristol: Bristol Classical Press, 1980).

Copi I M, *Introduction to Logic* (New York : Macmillan, 5th ed, 1978).

Cromwell T, "Dispute Resolution in the Twenty-First Century" in *Canadian Bar Association Systems of Civil Justice Task Force* (Toronto: Canadian Bar Association 1996).

Cross R, *Evidence* (London: Butterworths, 5th ed, 1979).

Damaska M R, *Evidence Law Adrift* (New Haven: Yale University Press, 1997).

Damaska M R, *The Faces of Justice and State Authority* (New Haven: Yale University Press, 1986).

Dawd P "Probability and Proof", Appendix to Anderson T and Twining W, *Analysis of Evidence: How to Do Things with Facts, Based on Wigmore's Science of Judicial Proof* (Boston: Little Brown & Co, 1991).

Delisle R J, *Evidence: Principles and Problems* (Toronto: Carswell, 1984).

Denning A, *Freedom Under the Law* (London: Stevens & Sons, 1949).

Dunham W, *Euler: The Master of Us All* (Washington, D.C: The Mathematical Association of America, 1999).

Flowers S, *Software Failure, Management Failure: Amazing Stories and Cautionary Tales* (Chichester: Wiley, 1996).

Freckleton I and Selby H, *The Law of Expert Evidence* (Sydney: LBC Information Services, 1999).

Goldschlager L and Lister A, *Computer Science: A Modern Introduction* (Englewood Cliffs, New Jersey: Prentice Hall, 2nd ed, 1988).

Hahn H, *The Internet Complete Reference* (Berkeley, California: Osborne McGraw-Hill, 1994).

Halstead M H, *Elements of Software Science* (New York: Elsevier, 1977).

Harris J W, *Legal Philosophies* (London: Butterworths, 2nd ed, 1997).

Heydon J D, *Cross on Evidence: Sixth Australian Edition* (Sydney: Butterworths, 2000).

Howard M N(ed), *Phipson on Evidence* (London : Sweet & Maxwell, 15th ed, 2000).

Inman K and Rudin N, *An Introduction to Forensic DNA Analysis* (New York: CRC Press, 1997).

Institute of Electrical and Electronics Engineers, *IEEE Guide for Use of IEEE Standard Dictionary of Measures to Produce Reliable Software* (New York: Institute of Electrical and Electronics Engineers, 1988).

Institute of Electrical and Electronics Engineers, *The IEEE Standard Dictionary of Electrical and Electronic Terms* (New York: Institute of Electrical and Electronics Engineers, 6th ed, 1996).

Jacob J I, *The Fabric of English Civil Justice* (London: Stevens & Sons, 1987).

Jelinski Z and Moranda P, "Software Reliability Research" in Freiberger W (ed), *Statistical Computer Performance Evaluation* (New York: Academic Press, 1972) 465–484.

Jolowicz J A, *On Civil Procedure* (Cambridge: Cambridge University Press, 2000).

Kaye D H "What is Bayesianism?" in Tiller P and Green E D (eds) *Probability and Inference in the Law of Evidence : The Uses and Limits of Bayesianism* (Boston: Kluwer Academic Publishers, 1988) 1-19.

Kenkel J L, *Introductory Statistics for Management and Economics* (Boston: PWS Kent, 3rd ed, 1989).

Kerans R P, *Standards of Review Employed by Appellate Courts* (Edmonton, Alberta: Juriliber, 1994).

Khoshafian S, *Object Orientation: Concepts, Analysis & Design, Languages, Databases, Graphical User Interfaces, Standards* (New York: Wiley, 1995).

Krawczak M and Scmidtke J, *DNA Fingerprinting* (Oxford: BIOS Scientific Publishers, 1994).

Lala P K, *Fault Tolerant and Fault Testable Hardware Design* (Englewood Cliffs, New Jersey: Prentice-Hall International, 1985).

Lee G, *From Hardware to Software* (London: Macmillan, 1982).

Lucas H C, *Introduction to Computers and Information Systems* (New York: Macmillan, 1986).

Lyu M R (ed) *Handbook of Software Reliability Engineering* (Los Alamitos: IEEE Computer Society Press: 1996).

Martin J, *Security, Accuracy, and Privacy in Computer Systems* (Englewood Cliffs, New Jersey: Prentice-Hall, 1973).

McCann D "Range of Findings Open to the Coroner" in Selby H (ed) *The Aftermath of Death* (Sydney: Federation Press, 1992) 11-21.

McEwan J, *Evidence and the Adversarial Process: The Modern Law* (Oxford: Hart Publishing, 1998).

Miller C J, *Contempt of Court* (Oxford: Oxford University Press, 3rd ed, 2000).

Moenssens A A, Inbau F E and Starrs J E, *Scientific Evidence in Criminal Cases* (Mineola, New York: Foundation Press, 3rd ed, 1986).

Moore B (ed), *The Australian Oxford Dictionary* (Melbourne: Oxford University Press, 1999).

Moore C C, *A Treatise on Facts or the Weight and Value of Evidence* (Northport, New York: Edward Thompson, 1908).

Nagylaki T, *Introduction to Theoretical Population Genetics* (New York: Springer-Verlag, 1992).

O'Connor D J, *Correspondence Theory of Truth* (London: Hutchinson, 1975).

Odgers S, *Uniform Evidence Law* (Sydney: LBC Information Services, 5th ed, 2002).

Palmer A, *Principles of Evidence* (Sydney: Cavendish, 1998).

Parsons T W, *Introduction to Compiler Construction* (New York: Computer Science Press, 1992).

Peled D A, *Software Reliability Methods* (New York: Springer, 2001).

Ramakumar R, *Engineering Reliability: Fundamentals and Applications* (Englewood Cliffs, New Jersey: Prentice-Hall, 1993).

Reed C (ed), *Computer Law* (London: Blackstone Press, 3rd ed, 1996).

Roberts G, *Evidence: Proof and Practice* (Sydney: Law Book Company, 1998).

Rook P (ed), *Software Reliability Handbook* (London: Elsevier, 1990).

Schneier B, *Applied Cryptography: Protocols, Algorithms, and Source Code in C* (New York: John Wiley & Sons, 2nd ed, 1996).

Sigfried S, *Understanding Object-Oriented Software Engineering* (Piscataway, New Jersey: IEEE Press, 1996).

Simpson J A and Weiner E S (eds), *The Oxford English Dictionary* (Oxford: Clarendon Press, 2nd ed, 1989).

Singpurwalla N D and Wilson S P, *Statistical Methods in Software Engineering: Reliability and Risk* (New York: Springer-Verlag, 1999).

Slater R, *Portraits in Silicon* (Cambridge, Massachusetts: MIT Press, 1987).

Stephen J F, *A Digest of the Law of Evidence* (London: Macmillan, 1893)

Stone J, Wells W A (ed) *Evidence: Its History and Policies* (North Ryde, New South Wales: Butterworths, 1991).

Stork D G (ed), *HAL's Legacy: 2001's Computer as Dream and Reality* (Cambridge, Massachusetts: MIT Press, 1997).

Tapper C, *Computer Law* (London: Longman, 4th ed, 1989).

Thayer J B, *A Preliminary Treatise on Evidence at the Common Law* (reprint: South Hackensack, New Jersey: Rothman, 1969).

Trankell A, *Reliability of Evidence. Methods for Analyzing and Assessing Witness Statements.* (Stockholm: Beckman, 1972).

Twining W L and Stein A, 'Evidence and Proof' in *The International Library of Essays in Law and Legal Theory* (Aldershot: Dartmouth, 1992).

Twining W L, *Rethinking Evidence: Exploratory Essays* (Oxford: Blackwell, 1990).

Twining W L, *Theories of Evidence: Bentham and Wigmore* (London: Weidenfeld and Nicolson, 1985).

Weaver W and Shannon C E, *The Mathematical Theory of Communication* (Urbana, Illinois: University of Illinois Press, 1949).

Wegner P, "Capital-Intensive Software Technology" in Biggerstaff T J and Perlis A J (eds), *Software Reusability Volume 1: Concepts and Models* (Reading, Massachusetts: ACM Press, 1989) 43-97.

Wells W A, *Evidence and Advocacy* (Sydney: Butterworths, 1988).

Wigmore J H, *A Students' Textbook of the Law of Evidence* (Brooklyn: Foundation Press, 1935).

Wigmore J H, *Evidence in Trials at Common Law* (Chadbourn revision) (Boston: Little Brown and Company, 1974).

Wigmore J H, *Evidence in Trials at Common Law* (Tillers revision) (Boston: Little Brown and Company, 1983).

Wigmore J H, *The Principles of Judicial Proof as Given by Logic, Psychology, and General Experience and Illustrated in Judicial Trials* (Boston: Little Brown & Co., 1st ed, 1913, 2nd ed, 1931, sub nom: *The Science of Judicial Proof* 3rd ed, 1937).

Zuckerman A A S, *The Principles of Criminal Evidence* (Oxford: Clarendon Press, 1989).

**ARTICLES**

Anastaplo G, "On Crime Lawyers and O.J Simpson: Plato's Gorgias Revisited" (1995) 26 *Loyola University Chicago Law Journal* 455-471.

Anonymous, "A Reconsideration of the Admissibility of Computer-Generated Evidence" (1977) *University of Pennsylvania Law Review* 425-451.

Atchison B, "DNA Statistics May be Misleading" (2003) 41 *New South Wales Law Society Journal* 68-70.

Basili V R and Perricone B T, "Software Errors and Complexity: An Empirical Investigation" (1984) 27 *Communications of the ACM* 42-52.

Bennett R B, Leibman J H and Fetter R E, "Seeing is Believing; Or is it? An Empirical Study of Computer Simulations as Evidence" (1999) 34 *Wake Forest Law Review* 257-294.

Bertolino A and Strigini L, "Assessing the Risk Due to Software Faults: Estimates of Failure Rate Versus Evidence of Perfection" (1998) 8 *Journal of Software Testing Verification and Reliability* 155-166.

Brooks F P, "No Silver Bullet: Essence and Accidents of Software Engineering" (1987) 20 *IEEE Computer* 10-19.

C E Powell, Computer Generated Visual Evidence: Does Daubert Make a Difference? (1996) 12 *Georgia State University Law Review* 577-599.

Carbine JE and McLain L, "Proposed Model Rules Governing the Admissibility of Computer-Generated Evidence" (1999) 15 *Santa Clara Computer and High-Technology Law Journal* 1-72.

Chatterjee S, Misra R B and Alam S S, "A Generalised Shock Model for Software Reliability" (1998) 24 *Computers & Electrical Engineering* 363-368.

Cohen J, "The Logic of Proof" [1980] *Criminal Law Review* 91-103.

Crowley-Smith L, "The Evidence Act 1995 (Cth): Should Computer Data be Presumed Accurate?" (1996) 22 *Monash University Law Review* 166-173.

Damaska M R, "Truth in Adjudication" (1998) 49 *Hastings Law Journal* 289-308.

Deutch M, "Computer Legislation: Israel's New Codified Approach" (1996) 14 *John Marshall Journal of Computer & Information Law* 461-482.

Eggleston R, "Beyond Reasonable Doubt" (1977) 4 *Monash Law Review* 1-22.

Eggleston R, "Probabilities and Proof" (1963) 4 *Melbourne University Law Review* 180-211.

Everett W, Keene S and Nikora A, "Applying Software Reliability Engineering in the 1990s" (1998) 47 *IEEE Transactions on Reliability* 372-378.

Fenton N E and Neil M, "A Critique of Software Defect Prediction Models" (1999) 25 *IEEE Transactions on Software Engineering* 675-689.

Flannery I M, "Frye or Frye Not: Should the Reliability of DNA Evidence be a Question of Weight or Admissibility?" (1992) 30 *American Criminal Law Review* 161-186.

Gappmair W "Claude E. Shannon: the 50th Anniversary of Information Theory" (1999) 37 *IEEE Communications Magazine* 102-105.

Garcia R, "'Garbage In, Gospel Out': Criminal Discovery, Computer Reliability, and the Constitution" (1991) 38 *UCLA Law Review* 1043-1145.

Goel A L, "Software Reliability Models: Assumptions, Limitations, and Applicability" (1985) 11 *IEEE Transactions on Software Engineering* 1411-1423.

Goodpaster G, "On the Theory of the American Adversary Criminal Trial" (1987) 78 *Journal of Criminal Law and Criminology* 118-153.

Hatton L, "Re-Examining the Fault Density–Component Size Connection" (1997) 14 *IEEE Software* 89-97.

Hinton M, "Unused Material and the Prosecutor's Duty of Disclosure" (2001) *Criminal Law Journal* 121-139.

Hodgson J, "A Lawyer Looks at Bayes' Theorem" (2002) 76 *Australian Law Journal* 109-118.

Imwinkelried E J, "A New Era in the Evolution of Scientific Evidence – A Primer on Evaluating the Weight of Scientific Evidence" (1981) 23 *William and Mary Law Review* 261-290.

Isoda S, "A Criticism on the Capture-and-Recapture Method for Software Reliability Assurance" (1998) 43 *Journal of Systems and Software* 3-10.

Jablon A, "'God mail': Authentication and Admissibility of Electronic Mail in Federal Courts" (1997) 34 *American Criminal Law Review* 1387-1409.

Jackson J D, "Analysing the New Evidence Scholarship: Towards a New Conception of the Law of Evidence" (1996) 16 *Oxford Journal of Legal Studies* 309-328.

Jones W and Gregory D, "Infinite-Failure Models for a Finite World: A Simulation Study of Fault Discovery" (1994) 43 *IEEE Transactions on Reliability* 520-526.

Kassin S M and Dunn M A, "Computer-Animated Displays and the Jury: Facilitative and Prejudicial Effects" (1997) 21 *Law and Human Behavior* 269-281.

Kelly M R, "Computer Generated Evidence as a Witness Beyond Cross Examination" (1995) 17 *Journal of Products and Toxics Liability* 95-115.

Kerr O S, "Computer Records and the Federal Rules of Evidence" (2001) 49 *United States Attorney's Bulletin* 25-32.

Khoshgoftaar T M, Allen E B, Halstead R, Trio G P and Flass R M, "Using Process History to Predict Software Quality" (1998) 31 *Computer* 66-72.

Kimura M, Yamada S and Osaki S, "Statistical Software Reliability Prediction and its Applicability Based on Mean Time between Failures", (1995) 22 *Mathematical and Computer Modelling* 149-155.

Kitchenham B and Pfleeger S L, "Software Quality: The Elusive Target", (1996) 13 *IEEE Software* 12-21.

Knüsel L, "On the Accuracy of Statistical Distributions in Microsoft Excel 97" (1998) 26 *Computational Statistics & Data Analysis* 375-377.

Kurzban S A, "Authentication of Computer-Generated Evidence in the United States Federal Courts" (1995) 35 *IDEA - The Journal of Law and Technology* 437-459.

Kwestel S, "The Business Records Exception to the Hearsay Rule–New is Not Necessarily Better" (1999) 64 *Missouri Law Review* 595-660.

Lanubile F, "Why Software Reliability Predictions Fail"(1996) 13 *IEEE Software* 131-137.

Laryea E T, "The Evidential Status of Electronic Data" (1999) *National Law Review* 3.

Lempert R, "After the DNA Wars: A Mopping Up Operation" (1997) 31 *Israel Law Review* 536-572.

Lempert R, "The New Evidence Scholarship: Analyzing the Process of Proof" (1986) 66 *Boston University Law Review* 439-477.

Leonard D P, "Power And Responsibility in Evidence Law" (1990) 63 *Southern California Law Review* 937-1013.

Lew K S, Dillon T S and Forward K E, "Software Complexity and its Impact on Software Reliability" (1998) 14 *IEEE Transactions on Software Engineering* 1645-1655.

Lewontin R C and Hartl D L, "Population Genetics in Forensic DNA Typing" (1991) 254 *Science* 1745-1750.

Macchiarola F J, "Finding the Truth in an American Criminal Trial: Some Observations" (1997) 5 *Cardozo Journal of International and Comparative Law* 97-111.

MacCrimmon M, "The Social Construction of Reality and the Rules of Evidence" in "A Forum on Lavellee v. R: Women and Self Defence" (1991) 25 *University of British Columbia Law Review* 23-68 at 36-50.

McCabe T J, "A Complexity Measure" (1976) 2 *IEEE Transactions on Software Engineering* 308-320.

McCullough B D and Wilson Berry, "On the Accuracy of Statistical Procedures in Microsoft Excel 2000 and Excel XP" (2002) 40 *Computational Statistics & Data Analysis* 713-721.

McCullough B D and Wilson Berry, "On the Accuracy of Statistical Procedures in Microsoft Excel 97" (1999) 31 *Computational Statistics & Data Analysis* 27-37.

McNiff F V, "Computer Documentation as Evidence: An Overview of Australian Legislation Facilitating Admissibility" (1981) 1 *Journal of Law and Information Science* 45-60.

Miller C, "Electronic Evidence—Can You Prove The Transaction Took Place?" (1992) 9 *Computer Lawyer* 21-33.

Munson J C, "Software Faults, Software Failures and Software Reliability Modeling" (1996) 38 *Information and Software Technology* 687-699.

Murray D R and Chorvot T J, "Stepping up to the Next Level: From the UETA to the URE and Beyond" (2001) 37 *Idaho Law Review* 415-439.

Nance D A, "The Best Evidence Principle" (1988) 73 *Iowa Law Review* 227-297.

Newman T R and Ahmuty S J, "Sufficiency and Weight of the Evidence" *New York Law Journal* (3 April 1996).

Parhami B, "Defect, Fault, Error, ... , or Failure?" (1997) 46 *IEEE Transactions on Reliability* 450-451.

Pasquini A, De Agostino E and Di Marco G, "Input-Domain Based Method to Estimate Software Reliability" (1996) 45 *IEEE Transactions on Reliability* 95-105.

Peritz R J, "Computer Data and Reliability: A Call for Authentication of Business Records Under the Federal Rules of Evidence" (1986) 80 *Northwestern University Law Review* 956-1002.

Quinn K, "Computer Evidence in Criminal Proceedings: Farewell to the Ill-fated Section 69 of the Police and Criminal Evidence Act 1984" (2001) 5 *International Journal of Evidence and Proof* 174-187.

Reed C, "The Admissibility and Authentication of Computer Evidence - A Confusion of Issues" [1990-91] 2 *The Computer Law and Security Report* 13-16.

Roberts A, "A Practitioner's Primer on Computer-Generated Evidence" (1974) 41 *University of Chicago Law Review* 254-280.

Schneidewind N F, "Reliability Modeling for Safety-Critical Software" (1997) 46 *IEEE Transactions on Reliability* 88-98.

Shannon C E, "A Mathematical Theory of Communication" (1948) 27 *Bell System Technical Journal* 379-423.

Shaviro D, "Statistical-Probability Evidence and the Appearance of Justice" (1989) 103 *Harvard Law Review* 530-554.

Sherer S A, "Software Fault Prediction" (1995) 29 *Journal of Systems and Software* 97-105.

Singpurwalla N D, "Failure Rate of Software: Does it exist?" (1995) 44 *IEEE Transactions on Reliability* 463-469.

Smith J C, "The Admissibility of Statements by Computer" [1981] *Criminal Law Review* 387-391.

Somani A K Vaidya N H, "Understanding Fault Tolerance and Reliability" (1997) 30 *Computer* 45-50.

Storer S, "The Weight Versus Admissibility Dilemma: Daubert's Applicability to a Method or Procedure in a Particular Case" (1998) 1 *University of Illinois Law Review* 231-252.

Storm P M, "Admitting Computer Generated Records: A Presumption of Reliability" (1984) 18 *John Marshall Law Review* 115-154.

Tapper C, "Discovery in Modern Times: A Voyage around the Common Law World" (1991) 67 *Chicago-Kent Law Review* 217-271.

Thayer J B, "Presumptions and the Law of Evidence" (1889) 3 *Harvard Law Review* 141-166.

Thomas W A, "Some Observations by a Scientist" (1987) 115 *Federal Rules Decisions* 142-144.

Thompson WC and Ford S, "DNA Typing: Acceptance and Weight of the New Genetic Identification Tests" (1989) 75 *Virginia Law Review* 45-108.

Tribe L H, "Trial by Mathematics: Precision and Ritual in the Legal Process" (1971) 84 *Harvard Law Review* 1329-1393.

Turing A, "Computing Machinery and Intelligence" (1950) 59 *Mind* 433-460.

Weinstein J B, "Some Difficulties in Devising Rules for Determining Truth in Judicial Trials" (1966) 66(2) *Columbia Law Review* 223-246.

Whittaker J A and Voas J, "Toward a More Reliable Theory of Software Reliability" (2000) 33 *IEEE Computer* 36-42.

Williams G, "The Mathematics of Proof" [1979] *Criminal Law Review* 297-308 and 340-354.

Wood A, "Predicting Software Reliability" (1996) 29 *Computer* 69-77.

Zeephongsekul P, Xia G and Kuma S, "Software-Reliability Growth Model: Primary-Failures Generate Secondary-Faults Under Imperfect Debugging" (1994) 43 *IEEE Transactions on Reliability* 408-413.

Zhang X and Pham H, "An Analysis of Factors Affecting Software Reliability" (2000) 50 *Journal of Systems and Software* 43-56.

**REPORTS**

Australian Law Reform Commission *Evidence Reference, Research Paper No. 3, Hearsay Evidence Proposal* (Sydney: Australian Law Reform Commission, 1981).

Australian Law Reform Commission, *Issues Paper 20: Review of the Adversarial System of Litigation* (Canberra: Australian Government Publishing Service, 1997).

Australian Law Reform Commission, *Report 26, Evidence (Interim)* (Canberra: Australian Government Publishing Service, 1985).

Australian Law Reform Commission, *Report 38, Evidence* (Canberra: Australian Government Publishing Service, 1987)

Australian Law Reform Commission, *Report 89, Managing Justice: A Review of the Federal Civil Justice System* (Canberra: Australian Government Publishing Service, 2000).

Bruce H W, *Internet, AARNet and Academic Work : A Longitudinal Study* (Canberra: Australian Government Publishing Service, 1996).

Law Commission of New Zealand, *Evidence Miscellaneous Paper 13: Total Recall? The Reliability of Witness Testimony* (Wellington: Law Commission, 1999).

Law Commission, *Evidence in Criminal Proceedings Hearsay and Related Topics* (London: H.M.S.O., 1997).

Law Commission, *Evidence of Bad Character in Criminal Proceedings* (London: H.M.S.O., 2001).

Law Commission, *The Hearsay Rule in Civil Proceedings* (London: H.M.S.O., 1993).

Law Reform Commission of Canada, *Report on Evidence* (Ottowa: Information Canada, 1975).

Law Reform Commission of Western Australia, *Report on the Admissibility in Evidence of Computer Records and Other Documentary Statements* (Perth: The Commission, 1978).

Law Reform Commission of Western Australia, *Review of the Criminal and Civil Justice System* (Perth: The Commission, 1999).

Law Reform Committee, *Thirteenth Report: Hearsay Evidence in Civil Proceedings* (London: H.M.S.O., 1966).

National Academy of Sciences, *The Evaluation of Forensic DNA Evidence.* Washington: National Academy Press, 1996)

New South Wales Law Reform Commission, *Report on Evidence (Business Records)* (Sydney: NSW Government Printer, 1973).

New Zealand Evidence Law Reform Committee, *Report on Business Records and Computer Output* (Wellington: The Committee, 1987).

Scottish Law Commission, *Report 149: Evidence: Report on Hearsay Evidence in Criminal Proceedings* (Edinburgh: H.M.S.O., 1995)

Woolf H K, *Access to Justice, Final Report to the Lord Chancellor on the Civil Justice System in England and Wales* (London: H.M.S.O., 1996).

Woolf H K, *Access to Justice, Interim Report to the Lord Chancellor on the Civil Justice System in England and Wales* (London: H.M.S.O., 1995).

## CONFERENCE PROCEEDINGS

Akiyama F, "An Example of Software System Debugging" *Proceedings, International Federation of Information Processing Societies Congress, 1971* (Amsterdam: North-Holland, 1971) 353-358.

Cukic B and Bastani F B, "On Reducing the Sensitivity of Software Reliability to Variations in the Operational Profile" *7th International Symposium on Software Reliability Engineering, October 30 - November 2, 1996, White Plains, New York* (Los Alamitos: IEEE Computer Society Press, 1996) 45-54.

Hamer P C and Frewin G D, "M. H. Halstead's Software Science: A Critical Examination" *Proceedings, 6th International Conference on Software Engineering, September 13-16, 1982, Tokyo, Japan* (Long Beach, California: IEEE Computer Society, 1982) 197-207.

## OTHER SOURCES

Federal Rules of Evidence, *Historical Notes and Legislative Commentary* <http://www2.law.cornell.edu/cgi-bin/foliocgi.exe/fre/query=[jump!3A!27acrule802!27]/doc/{@774}> (Visited 2 June 2003).

International Organization for Standardization, *International Standard ISO/IEC 10646-1:2000, Information technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane* (Geneva: International Organization for Standardization, 2nd ed, 2000).

International Organization for Standardization, *Standard No 14882 Programming Languages -- C++*, International Organization for Standardization, 1998.

Network Working Group, *Request for Comments: 2068, Hypertext Transfer Protocol HTTP/1.1* (Reston, Virginia: Internet Engineering Task Force (Secretariat), 1997).

Parliament of the Commonwealth of Australia, *Parliamentary Paper No. 302/1985. Report 26, Evidence (Interim))* 21 August 1985.

Postel J B, *Request for Comments: 821, Simple Mail Transfer Protocol* (Marina del Rey, California: Information Sciences Institute, University of Southern California, 1982).

Sage E R, *Israel – Excerpts from the Computer Law* 5755-1995 <http://www.rgr.co.il/English/Resources/COMPUTERr.pdf> (Visited 2 June 2003).

Uniform Law Conference of Canada, *Uniform Electronic Evidence Act Consultation Paper* (Ottawa: 1997). <http://www.law.ualberta.ca/alri/ulc/current/eelev.htm> (Visited 2 June 2003).