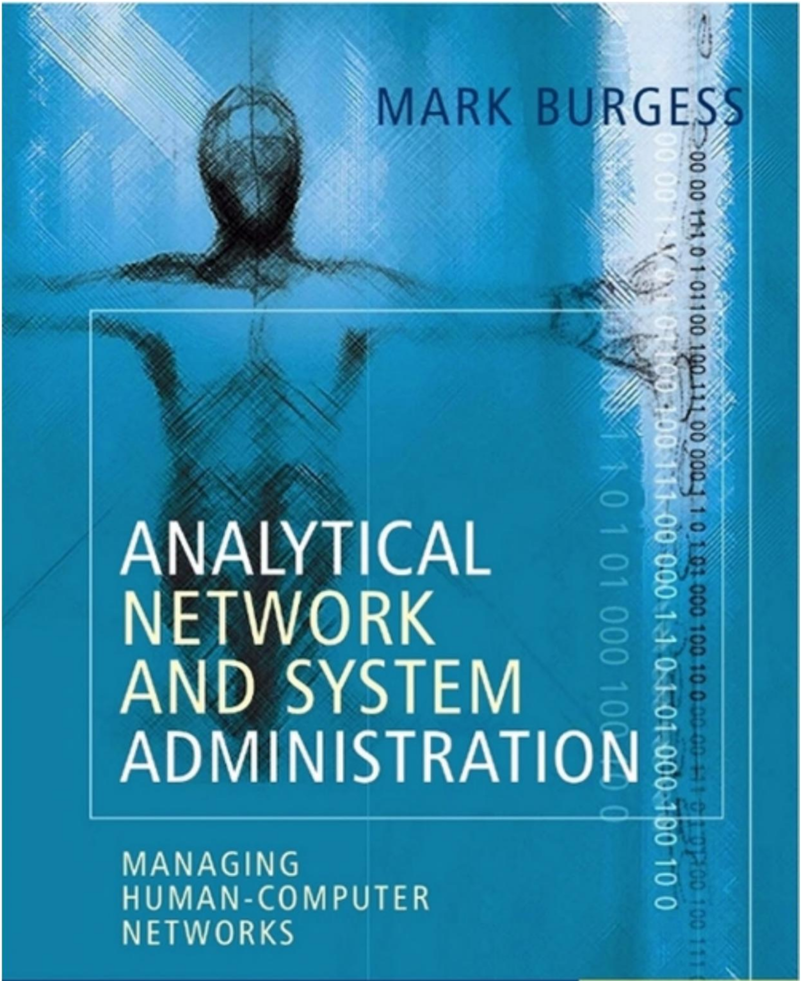


MARK BURGESS

ANALYTICAL NETWORK AND SYSTEM ADMINISTRATION

MANAGING HUMAN-COMPUTER NETWORKS

WILEY

Contents

Cover

Half Title page

Title page

Copyright page

Foreword

Preface

Chapter 1: Introduction

- 1.1 What is system administration?
- 1.2 What is a system?
- 1.3 What is administration?
- 1.4 Studying systems
- 1.5 What's in a theory?
- 1.6 How to use the text
- 1.7 Some notation used

Chapter 2: Science and its methods

- 2.1 The aim of science
- 2.2 Causality, superposition and dependency
- 2.3 Controversies and philosophies of science
- 2.4 Technology
- 2.5 Hypotheses
- 2.6 The science of technology
- 2.7 Evaluating a system—dependencies
- 2.8 Abuses of science

Chapter 3: Experiment and observation

3.1 Data plots and time series
3.2 Constancy of environment during measurement
3.3 Experimental design
3.4 Stochastic (random) variables
3.5 Actual values or characteristic values
3.6 Observational errors
3.7 The mean and standard deviation
3.8 Probability distributions and measurement
3.9 Uncertainty in general formulae
3.10 Fourier analysis and periodic behaviour
3.11 Local averaging procedures
3.12 Reminder

Chapter 4: Simple systems

4.1 The concept of a system
4.2 Data structures and processes
4.3 Representation of variables
4.4 The simplest dynamical systems
4.5 More complex systems
4.6 Freedoms and constraints
4.7 Symmetries
4.8 Algorithms, protocols and standard ‘methods’
4.9 Currencies and value systems
4.10 Open and closed systems: the environment
4.11 Reliable and unreliable systems

Chapter 5: Sets, states and logic

5.1 Sets
5.2 A system as a set of sets
5.3 Addresses and mappings
5.4 Chains and states
5.5 Configurations and macrostates
5.6 Continuum approximation

- 5.7 Theory of computation and machine language
- 5.8 A policy-defined state

Chapter 6: Diagrammatical representations

- 6.1 Diagrams as systems
- 6.2 The concept of a graph
- 6.3 Connectivity
- 6.4 Centrality: maxima and minima in graphs
- 6.5 Ranking in directed graphs
- 6.6 Applied diagrammatical methods

Chapter 7: System variables

- 7.1 Information systems
- 7.2 Addresses, labels, keys and other resource locators
- 7.3 Continuous relationships
- 7.4 Digital comparison

Chapter 8: Change in systems

- 8.1 Renditions of change
- 8.2 Determinism and predictability
- 8.3 Oscillations and fluctuations
- 8.4 Rate of change
- 8.5 Applications of the continuum approximation
- 8.6 Uncertainty in the continuum approximation

Chapter 9: Information

- 9.1 What is information?
- 9.2 Transmission
- 9.3 Information and control
- 9.4 Classification and resolution
- 9.5 Statistical uncertainty and entropy
- 9.6 Properties of the entropy
- 9.7 Uncertainty in communication

9.8 A geometrical interpretation of information
9.9 Compressibility and size of information
9.10 Information and state
9.11 Maximum entropy principle
9.12 Fluctuation spectra

Chapter 10: Stability
10.1 Basic notions
10.2 Types of stability
10.3 Constancy
10.4 Convergence of behaviour
10.5 Maxima and minima
10.6 Regions of stability in a graph
10.7 Graph stability under random node removal
10.8 Dynamical equilibria: compromise
10.9 Statistical stability
10.10 Scaling stability
10.11 Maximum entropy distributions
10.12 Eigenstates
10.13 Fixed points of maps
10.14 Metastable alternatives and adaptability
10.15 Final remarks

Chapter 11: Resource networks
11.1 What is a system resource?
11.2 Representation of resources
11.3 Resource currency relationships
11.4 Resource allocation, consumption and conservation
11.5 Where to attach resources?
11.6 Access to resources
11.7 Methods of resource allocation
11.8 Directed resources: flow asymmetries

Chapter 12: Task management and services

12.1 Task list scheduling
12.2 Deterministic and non-deterministic schedules
12.3 Human–computer scheduling
12.4 Service provision and policy
12.5 Queue processing
12.6 Models
12.7 The prototype queue M/M/1
12.8 Queue relationships or basic ‘laws’
12.9 Expediting tasks with multiple servers M/M/*k*
12.10 Maximum entropy input events in periodic systems
12.11 Miscellaneous issues in scheduling

Chapter 13: System architectures

13.1 Policy for organization
13.2 Informative and procedural flows
13.3 Structured systems and *ad hoc* systems
13.4 Dependence policy
13.5 System design strategy
13.6 Event-driven systems and functional systems
13.7 The organization of human resources
13.8 Principle of minimal dependency
13.9 Decision-making within a system
13.10 Prediction, verification and their limitations
13.11 Graphical methods

Chapter 14: System normalization

14.1 Dependency
14.2 The database model
14.3 Normalized forms

Chapter 15: System integrity

15.1 System administration as communication?
15.2 Extensive or strategic instruction
15.3 Stochastic semi-groups and martingales

15.4 Characterizing probable or average error
15.5 Correcting errors of propagation
15.6 Gaussian continuum approximation formula

Chapter 16: Policy and maintenance

16.1 What is maintenance?
16.2 Average changes in configuration
16.3 The reason for random fluctuations
16.4 Huge fluctuations
16.5 Equivalent configurations and policy
16.6 Policy
16.7 Convergent maintenance
16.8 The maintenance theorem
16.9 Theory of back-up and error correction

Chapter 17: Knowledge, learning and training

17.1 Information and knowledge
17.2 Knowledge as classification
17.3 Bayes' theorem
17.4 Belief versus truth
17.5 Decisions based on expert knowledge
17.6 Knowledge out of date
17.7 Convergence of the learning process

Chapter 18: Policy transgressions and fault modelling

18.1 Faults and failures
18.2 Deterministic system approximation
18.3 Stochastic system models
18.4 Approximate information flow reliability
18.5 Fault correction by monitoring and instruction
18.6 Policy maintenance architectures
18.7 Diagnostic cause trees
18.8 Probabilistic fault trees

Chapter 19: Decision and strategy

- 19.1 Causal analysis
- 19.2 Decision-making
- 19.3 Game theory
- 19.4 The strategic form of a game
- 19.5 The extensive form of a game
- 19.6 Solving zero-sum games
- 19.7 Dominated strategies
- 19.8 Nash equilibria
- 19.9 A security game
- 19.10 The garbage collection game
- 19.11 A social engineering game
- 19.12 Human elements of policy decision
- 19.13 Coda: extensive versus strategic configuration management

Chapter 20: Conclusions

Appendix A: Some Boolean formulae

- A.1 Conditional probability
- A.2 Boolean algebra and logic

Appendix B: Statistical and scaling properties of time-series data

- B.1 Local averaging procedure
- B.2 Scaling and self-similarity
- B.3 Scaling of continuous functions

Appendix C: Percolation conditions

- C.1 Random graph condition
- C.2 Bi-partite form
- C.3 Small-graph corrections

Bibliography

Index

Analytical Network and System Administration

Analytical Network and System Administration

Managing Human–Computer Networks

Mark Burgess
Oslo University College, Norway

John Wiley & Sons, Ltd

John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex
PO19 8SQ, England

Telephone (+44) 1243 779777

Email (for orders and customer service enquiries): cs-books@wiley.co.uk
Visit our Home Page on www.wileyeurope.com or www.wiley.com

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold on the understanding that the Publisher is not engaged in rendering professional services. If professional advice or other expert assistance is required, the services of a competent professional should be sought.

Other Wiley Editorial Offices

John Wiley & Sons Inc., 111 River Street, Hoboken, NJ 07030, USA

Jossey-Bass, 989 Market Street, San Francisco, CA 94103-1741, USA

Wiley-VCH Verlag GmbH, Boschstr. 12, D-69469 Weinheim, Germany

John Wiley & Sons Australia Ltd, 33 Park Road, Milton, Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01, Jin Xing Distripark, Singapore 129809

John Wiley & Sons Canada Ltd, 22 Worcester Road, Etobicoke, Ontario, Canada M9W 1L1

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN 0-470-86100-2

Foreword

It is my great honor to introduce a landmark book in the field of network and system administration. For the first time, in one place, one can study the components of network and system administration as an evolving and emerging discipline and science, rather than as a set of recipes, practices or principles. This book represents the step from 'mastery of the practice' and 'scientific understanding', a step very similar to that between historical alchemy and chemistry.

As recently as ten years ago, many people considered 'network and system administration' to comprise remembering and following complex recipes for building and maintaining systems and networks. The complexity of many of these recipes—and the difficulty of explaining them to non-practitioners in simple and understandable terms—encouraged practitioners to treat system administration as an 'art' or 'guild craft' into which practitioners are initiated through apprenticeship.

Current master practitioners of network and system administration are perhaps best compared with historical master alchemists at the dawn of chemistry as a science. In contrast to the distorted popular image of alchemy as seeking riches through transmutation of base metals, historical research portrays alchemists as master practitioners of the subtle art of combining chemicals towards particular results or ends. Practitioners of alchemy often possessed both precise technique and highly developed observational skills. Likewise, current master practitioners of network and system administration craft highly reliable networks from a mix of precise practice, observational skills and the intuition that comes from careful observation of network behaviour over long time periods. But both alchemists and master practitioners lack the common language that makes it easy to exchange valuable information with others: the language of science.

Alas, the alchemy by which we have so far managed our networks is no longer sufficient. When networks were simple in structure, it was possible to maintain them through the use of relatively straightforward recipes, procedures and practices. In the post-Internet world, the administrator is now faced with managing and controlling networks that can dynamically adapt to changing conditions and requirements quickly and, perhaps, even unpredictably. These adaptive networks can exhibit 'emergent properties' that are not predictable in advance. In concert with adapting networks to serve human needs, future administrators must adapt themselves to the task of management by developing an ongoing, perpetually evolving, and shared understanding.

In the past, it was reasonable to consider a computer network as a collection of cooperating machines functioning in isolation. Adaptive networks cannot be analysed in this fashion; their human components must also be considered. Modern networks are not communities of machines, but rather communities of humans inextricably linked by machines; what the author calls 'cooperating ecologies' of users and machines. The behaviour of humans must be considered along with the behaviour of the network for making conclusions about network performance and suitability.

These pressures force me to an inescapable conclusion. System administrators cannot continue to be alchemist-practitioners. They must instead develop the language of science and evolve from members of a profession to researchers within a shared scientific discipline. This book shows the way.

Though we live thousands of miles apart, the author and I are 'kindred spirits'—forged by many of the same experiences, challenges and insights. In the late 1980s and early 1990s, both of us were faculty, managing our own computer networks for teaching and research. Neither of us had access to the contemporary guilds of system administration (or each other), and had to learn how to administer networks the hard way—by reading the documentation and creating our own recipes for success. Both of us realized (completely independently) that there were simple concepts behind the recipes that, once discovered, make the recipes easy to remember, reconstruct and understand. Concurrently and independently, both of us set out to create software tools that would avoid repeated manual configuration.

Although we were trained in radically differing academic traditions (the author from physics and myself from mathematics and computer science), our administrative tools, developed completely in isolation from one another, had very similar capabilities and even accomplished tasks using the same methods. The most striking similarity was that both tools were based upon the same 'principles'. For the first time, it very much looked like we had found an invariant principle in the art of system and network administration: the 'principle of convergence'. As people would say in the North Carolina backwoods near where I grew up, 'if it ain't broke, don't fix it'.

The road from alchemy to discipline has many steps. In the author's previous book, *Principles of Network and System Administration*, he takes the first step from practice ('what to do') to principles ('why to do it'). Recipes are not created equal; some are better than others. Many times the difference between good and poor recipes can be expressed in terms of easily understood principles. Good recipes can then be constructed top–down, starting at the principles. Practitioners have approached the same problem bottom-up, working to turn their tested and proven recipes into sets of 'best practices' that are guaranteed to work well for a particular site or application. Recently, many practitioners have begun to outline the 'principles' underlying their practices. There is remarkable similarity between the results of these two seemingly opposing processes, and the author's 'principles', and the practitioners' 'best practices' are now quickly meeting on a common middle ground of principles.

In this book, for the first time, the author identifies principles of scientific practice and observation that anyone can use to become proficient 'analysts' of network and system administration practices. This will not make one a better practitioner, but rather will allow one to discuss and evaluate the practice with others in a clear and concise manner. The reader will not find any recipes in this book. The reader will not find principles of practice. Rather, the book explains the principles behind the science and chemistry of cooking, so that one can efficiently derive one's own efficient and effective recipes for future networks. Proficient system administrators have always been capable of this kind of alchemy, but have found it challenging to teach the skill to others. This book unlocks the full power of the scientific method to allow sharing of analyses, so that future administrators can look beyond recipe, to shared understanding and discipline. In this way, now-isolated practitioners can form a shared scientific community and discipline whose knowledge is greater than the sum of its parts.

Looking at the table of contents, one will be very surprised to note that the traditional disciplines of 'computer science' and 'computer engineering'—long considered the inseparable partners of system administration—are not the basis of the new science. Rather, experimental physics has proven to be the Rosetta Stone that unlocks the mysteries of complex systems. To understand why, we must examine the fundamental differences in economics between the disciplines of computer science and engineering and the disciplines of network and system administration.

Traditional computer science and engineering (and, particularly, the sciences involved in building the systems that system administrators manage) are based upon either an operational or axiomatic semantic model of computing. Both models express 'what a program does' in an ideal computing environment. Software developers build complex systems in layers, where each subsequent layer presumes the correct function of layers upon which it is built. Program correctness at a given layer is a mathematical property based upon axioms that describe the behaviour of underlying layers. Fully understanding a very complex system requires understanding of each layer and its interdependencies and assumptions in dealing with other layers.

System administrators have a differing view of the systems they manage compared to that of the developers who designed the systems. It is not economically feasible to teach the deep knowledge and mathematical understanding necessary to craft and debug software and systems to large populations of human system administrators. System administrators must instead base their actions upon a high-level set of initial experimental hypotheses called the 'system documentation'. The documentation consists of hypotheses to be tested, not axioms to be trusted. As administrators learn how to manage a system, they refine their understanding top-down, by direct observation and ongoing evaluation of hypotheses.

Turning system and network administration into a discipline requires one to learn some skills, previously considered far removed from the practice. Evaluating hypotheses requires a rudimentary knowledge of statistics and the experimental method. These hypotheses are built not upon operational or axiomatic semantic models of computing, but upon specialized high-level mathematical models that describe behaviour of a complex system. With this machinery in hand, several advanced methods of analysis—prevalent in experimental physics and other scientific disciplines—are applied to the problem of understanding management of complex systems.

Proficient system administrators are already skilled experimental scientists;

they just do not acknowledge this fact and cannot effectively communicate their findings to others. This book takes a major step towards understanding the profession of system and network administration as a science rather than as an art. While this step is difficult to take, it is both rewarding and necessary for those pioneers who will manage the next generation of networks and services. Please read on, and seek to understand the true nature of networking—as a process that involves connecting humans, not just computers.

Alva Couch
Tufts University, USA

Preface

This is a research document and a textbook for graduate students and researchers in the field of networking and system administration. It offers a theoretical perspective on human–computer systems and their administration. The book assumes a basic competence in mathematical methods, common to undergraduate courses. Readers looking for a less theoretical introduction to the subject may wish to consult (Burgess (2000b)).

I have striven to write a short book, treating topics briefly rather than succumbing to the temptation to write an encyclopædia that few will read or be able to lift. I have not attempted to survey the literature or provide any historical context to the development of these ideas (see Anderson et al. (2001)). I hope this makes the book accessible to the intelligent lay reader who does not possess an extensive literacy in the field and would be confused by such distractions. The more advanced reader should find sufficient threads to follow to add depth to the material. In my experience, too much attention to detail merely results in one forgetting why one is studying something at all. In this case, we are trying to formulate a descriptive language for systems.

A theoretical synthesis of system administration plays two roles: it provides a descriptive framework for systems that should be available to other areas of computer science and proffers an analytical framework for dealing with the complexities of interacting components. The field of system administration meets an unusual challenge in computer science: that of approximation. Modern computing systems are too complicated to be understood in exact terms.

In the flagship theory of physics, quantum electrodynamics, one builds everything out of two simple principles:

1. Different things can exist at different places and times.

2. For every effect, there must be a cause.

The beauty of this construction is its lack of assumptions and the richness of the results. In this text, I have tried to synthesize something like this for human–computer systems. In order to finish the book, and keep it short and readable, I have had to compromise on many things. I hope that the result nevertheless contributes in some way to a broader scientific understanding of the field and will inspire students to further serious study of this important subject.

Some of this work is based on research performed with my collaborators Geoff Canright, Frode Sandnes and Trond Reitan. I have benefited greatly from discussions with them and others. I am especially grateful for the interest and support of other researchers, most notably Alva Couch for understanding my own contributions when no one else did. Finally, I would like to thank several for reading the draft versions of the manuscript and commenting: Paul Anderson, Lars Kristiansen, Tore Jonassen, Anil Somayaji and Jan Bergstra.

Mark Burgess

Chapter 1

Introduction

> Technology: the science of the mechanical and industrial arts.
> [Gk. *tekhne* art and *logos* speech].
>
> —Odhams dictionary of the English language

1.1 What is system administration?

System administration is about the design, running and maintenance of human–computer systems. Human–computer systems are ‘communities’ of people and machines that collaborate actively to execute a common task. Examples of human–computer systems include business enterprises, service institutions and any extensive machinery that is operated by, or interacts with human beings. The human players in a human–computer system are often called the *users* and the machines are referred to as *hosts*, but this suggests an asymmetry of roles, which is not always the case.

System administration is primarily about the technological side of a system: the architecture, construction and optimization of the collaborating parts, but it also occasionally touches on softer factors such as user assistance (help desks), ethical considerations in deploying a system, and the larger implications of its design for others who come into contact with it. System administration deals first and foremost with the system as a whole, treating the individual components as black boxes, to be opened only when it is possible or practical to do so. It does not conventionally consider the design of user-tools such as third-party computer programs, nor does it attempt to design enhancements to the available software, though it does often discuss meta tools and improvised software systems that can be used to monitor, adjust or even govern the system. This omission is mainly because user-software is acquired beyond the control of a system administrator; it is written by third parties, and is not open to local modification. Thus, users’ tools and software are treated as ‘given quantities’ or ‘boundary conditions’.

For historical reasons, the study of system administration has fallen into two camps: those who speak of *network management* and discuss its problems in terms of software design for the management of black box devices by humans (e.g. using SNMP), and those who speak of *system administration* and concern themselves with practical strategies of machine and software configuration at all levels, including automation, human–computer issues and ethical considerations. These two viewpoints are complementary, but too often ignore one another. This book considers human–computer systems in general, and refers to specific technologies only by example. It is therefore as much about purely human administrative systems as it is about computers.

1.2 What is a system?

A system is most often an organized effort to fulfil a goal, or at least carry out some predictable behaviour. The concept is of the broadest possible generality. A system could be a mechanical device, a computer, an office of workers, a network of humans and machines, a series of forms and procedures (a bureaucracy) etc. Systems involve themes, such as *collaboration* and *communication* between different actors, the use of *structure* to represent information or to promote efficiency, and the laws of *cause and effect.* Within any mechanism, *specialization* of the parts is required to build significant innovation; it is only through strategy of divide and conquer that significant problems can be solved. This implies that each division requires a special solution.

A computer system is usually understood to mean a system composed primarily of computers, using computers or supporting computers. A human–computer system includes the role of humans, such as in a business enterprise where computers are widely used. The principles and theories concerning systems come from a wide range of fields of study. They are synthesized here in a form and language that is suitable for scholars of science and engineering.

1.3 What is administration?

The word *administration* covers a variety of meanings in common parlance. The American Administration is the government of the United States, that is, a political leadership. A university administration is a bureaucracy and economic resource department that works on behalf of a board of governors to implement the university's policy and to manage its resources. The administrative department of a company is generally the part that handles economic procedures and payment transactions. In human–computer system administration, the definition is broadened to include all of the organizational aspects and also engineering issues, such as system fault diagnosis. In this regard, it is like the medical profession, which combines checking, management and repair of bodily functions. The main issues are the following:

- System design and rationalization

- Resource management
- Fault finding.

In order to achieve these goals, it requires

- Procedure
- Team work
- Ethical practices
- Appreciation of security.

Administration comprises two aspects: *technical solutions* and *arbitrary policies.* A technical solution is required to achieve goals and sub-goals, so that a problem can be broken down into manageable pieces. Policy is required to make the system, as far as possible, *predictable*: it pre-decides the answers to questions on issues that cannot be derived from within the system itself. Policy is therefore an arbitrary choice, perhaps guided by a goal or a principle.

The arbitrary aspect of policy cannot be disregarded from the administration of a system, since it sets the boundary conditions under which the system will operate, and supplies answers to questions that cannot be determined purely on the grounds of efficiency. This is especially important where humans are involved: human welfare, permissions, responsibilities and ethical issues are all parts of policy. Modelling these intangible qualities formally presents some challenges and requires the creative use of abstraction.

The administration of a system is an administration of temporal and resource development. The administration of a network of localized systems (a so-called *distributed system*) contains all of the above, and, additionally, the administration of the location of and communication between the system's parts. Administration is thus a flow of activity, information about resources, policy making, record keeping, diagnosis and repair.

1.4 Studying systems

There are many issues to be studied in system administration. Some issues are of a technical nature, while others are of a human nature. System administration confronts the human–machine interaction as few other branches of computer science do. Here are some examples:

- *System design* (e.g. how to get humans and machines to do a particular job as efficiently as possible. What works? What does not work? How does one know?)
- *Reliability studies* (e.g. failure rate of hardware/software, evaluation of policies and strategies)
- *Determining and evaluating methods for ensuring system integrity* (e.g. automation, cooperation between humans, formalization of policy, contingency planning etc.)
- *Observations that reveal aspects of system behaviour that are difficult to predict* (e.g. strange phenomena, periodic cycles)

- *Issues of strategy and planning.*

Usually, system administrators do not decide the purpose of a system; they are regarded as supporting personnel. As we shall see, this view is, however, somewhat flawed from the viewpoint of system design. It does not always make sense to separate the human and computer components in a system; as we move farther into the information age, the fates of both become more deeply intertwined.

To date, little theory has been applied to the problems of system administration. In a subject that is complex, like system administration, it is easy to fall back on *qualitative* claims. This is dangerous, however, since one is easily fooled by qualitative descriptions. Analysis proceeds as a dialogue between theory and experiment. We need theory to interpret results of observations and we need observations to back up theory. Any conclusions must be a consistent mixture of the two. At the same time, one must not believe that it is sensible to demand hard-nosed Popper-like falsification of claims in such a complex environment. Any numbers that we can measure, and any models we can make must be considered valuable, provided they actually have a sensible interpretation.

Human–computer interaction

The established field of human–computer interaction (HCI) has grown, in computer science, around the need for reliable interfaces in critical software scenarios (see for instance Sheridan (1996); Zadeh (1973)). For example, in the military, real danger could come of an ill-designed user interface on a nuclear submarine; or in a power plant, a poorly designed system could set off an explosion or result in blackouts.

One can extend the notion of the HCI to think less as a programmer and more as a physicist. The task of physics is to understand and describe what happens when different parts of nature interact. The interaction between fickle humans and rigid machinery leads to many unexpected phenomena, some of which might be predicted by a more detailed functional understanding of this interaction. This does not merely involve human attitudes and habits; it is a problem of systemic complexity—something that physics has its own methods to describe. Many of the problems surrounding computer security enter into the equation through the HCI. Of all the parts of a system, humans bend most easily: they are often both the weakest link and the most adaptable tools in a solution, but there is more to the HCI than psychology and button pushing. The issue reaches out to the very principles of science: what are the relevant timescales for the interactions and for the effects to manifest? What are the sources of predictability and unpredictability? Where is the system immune to this interaction, and where is the interaction very strong? These are not questions that a computer science analysis alone can answer; there are physics questions behind these issues. Thus, in reading this book, you should not be misled into thinking that physics is merely about electrons, heat and motion: it is a broad methodology for 'understanding phenomena', no matter where they occur, or how they are described. What computer science lacks from its attachment to technology, it must regain by appealing to the physics of systems.

Policy

The idea policy plays a central role in the administration of systems, whether they are dominated by human or technological concerns.

Definition 1 (Policy—heuristic) *A policy is a description of what is intended and desirable about a system. It includes a set of ad hoc choices, goals, compromises, schedules, definitions and limitations about the system. Where humans are involved, compromises often include psychological considerations, and welfare issues.*

A policy provides a frame of reference in which a system is understood to operate. It injects a relativistic aspect into the science of systems: we cannot expect to find absolute answers, when different systems play by different rules and have different expectations. A theory of systems must therefore take into account policy as a basic axiom. Much effort is expended in the chapters that follow to find a tenable definition of policy.

Stability and instability

It is in the nature of almost all systems to change with time. The human and machine parts of a system change, both in response to one another, and in response to a larger environment. The system is usually a predictable, known quantity; the environment is, by definition, an unknown quantity. Such changes tend to move the system in one or two directions: either the system falls into disarray or it stagnates. The meaning of these provocative terms is different for the human and the machine parts:

- Systems will fall into a stable repetition of behaviour (a limit cycle) or reach some equilibrium at which point further change cannot occur without external intervention.
- Systems will eventually invalidate their assumptions and fail to fulfil their purpose.

Ideally, a machine will perform, repetitively, the same job over and over again, because that is the function of mechanisms: stagnation is good for machines. For humans, on the other hand, this is usually regarded as a bad thing, since humans are valued for their creativity and adaptability. For a system mechanism to fall into disarray is a bad thing.

The relationship between a system and its environment is often crucial in determining which of the above is the case. The inclusion of human behaviour in systems must be modelled carefully, since humans are not deterministic in the same way that machines (automata) can be. Humans must therefore be considered as being part system and part environment. Finally, policy itself must be our guide as to what is desirable change.

Security

Security is a property of systems that has come to the forefront of our attention in recent times. How shall we include it in a theory of system administration?

> **Definition 2 (Security)** *Security concerns the possible ways in which a system's integrity might be compromised, causing it to fail in its intended purpose. In other words, a breach of security is a failure of a system to meet its specifications.*

Security refers to 'intended purpose', so it is immediately clear that it relates directly to *policy* and that it is a property of the entire system in general. Note also that, while we associate security with 'attacks' or 'criminal activity', natural disasters or other occurrences are equally to be blamed for the external perturbations that break systems.

A loss of integrity can come from a variety of sources, for example, an internal fault, an accident or a malicious attack on the system. Security is a property that requires the analysis of assumptions that underpin the system, since it is these areas that one tends to disregard and that can be exploited by attackers, or fail for diverse reasons. The system depends on its components in order to function. Security is thus about an analysis of *dependencies.* We can sum this up in a second definition:

> **Definition 3 (Secure system)** *A secure system is one in which every possible threat has been analysed and where all the risks have been assessed and accepted as a matter of policy.*

1.5 What's in a theory?

This book is not a finished theory, like the theory of relativity, or the theory of genetic replication. It is not the end of a story, but a beginning. System administration is at the start of its scientific journey, not at its end.

Dramatis personae

The players in system administration are the following:

- The computer
- The network
- The user
- The policy
- The system administrator.

We seek a clear and flexible language (rooted in mathematics) in which to write their script. It will deal with basic themes of

- time (when events occur or should occur),
- location (where resources should be located),
- value (how much the parts of a system contribute or are worth),
- randomness and predictability (our ability to control or specify).

It must answer questions that are of interest to the management of systems. We can use two strategies:

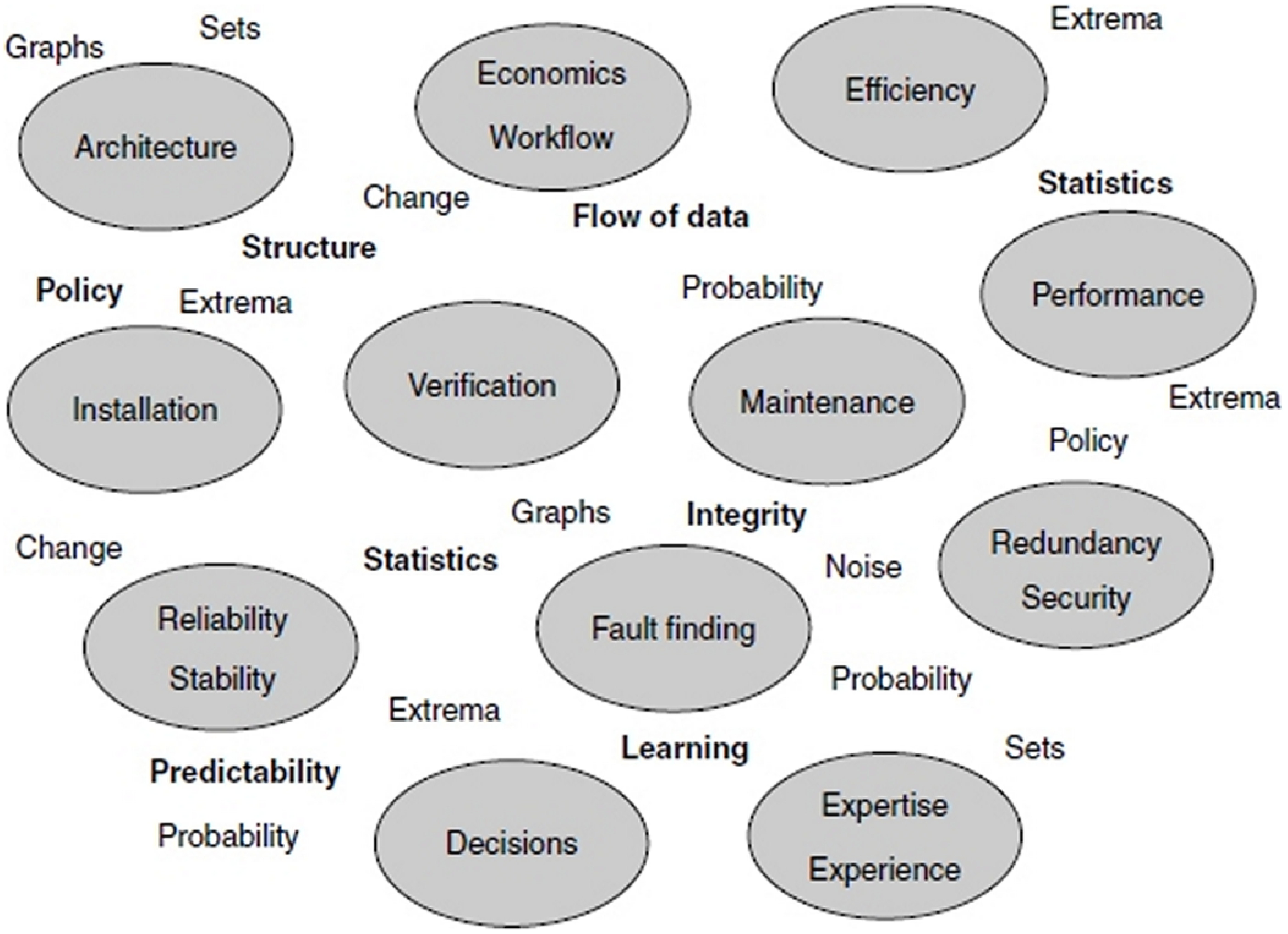
- Type I (pure science) models that describe the behaviour of a system without attempting to interpret its value or usefulness. These are ‘vignettes’ that describe what we can observe and explain in impartial terms. They provide a basic understanding of phenomena that leads to expertise about the system.
- Type II (applied science) models add interpretations of value and correctness (policy) to the description. They help us in making decisions by impressing a rational framework on the subjectivities of policy.

A snapshot of reality

The system administrator rises and heads for the computer, grabs coffee or cola and proceeds to catch up on e-mail. There are questions, bug reports, automatic replies from scripted programs, spam and lengthy discussions from mailing lists.

The day proceeds to planning, fault finding, installing software, modifying system parameters to implement (often ad hoc) policy that enables the system to solve a problem for a user, or which makes the running smoother (more predictable)—see fig. 1.1. On top of all of this, the administrator must be thinking about what users are doing. After all, they are the ones who need the system and the ones who most often break it. How does ‘the system’ cope with them and their activities as they feed off it and feed back on it? They are, in every sense, a part of the system. How can their habits and skills be changed to make it all work more smoothly? This will require an appreciation of the social interactions of the system and how they, in turn, affect the structures of the logical networks and demands placed on the machines.

Figure 1.1: The floating islands of system administration move around on a daily basis and touch each other in different ways. In what framework shall we place these? How can we break them down into simpler problems that can be ‘solved’? In courier font, we find some primitive concepts that help to describe the broader ideas. These will be our starting points.



There are decisions to be made, but many of them seem too uncertain to be able to make a reliable judgement on the available evidence. Experimentation is required, and searching for advice from others. Unfortunately, you never know how reliable others' opinions and assertions will be. It would be cool if there were a method for turning the creative energy into the optimal answer. There is ample opportunity and a wealth of tools to collect information, but how should that information be organized and interpreted? What is lacking is not software, but theoretical tools.

What view or philosophy could unify the different facets of system administration: design, economics, efficiency, verification, fault-finding, maintenance, security and so on? Each of these issues is based on something more primitive or fundamental. Our task is therefore to use the power of abstraction to break down the familiar problems into simpler units that we can master and then reassemble into an approximation of reality. There is no unique point of view here (see next chapter).

Theory might lead to better tools and also to better procedures. If it is to be of any use, it must have predictive power as well as descriptive power. We have to end up with formulae and procedures that make criticism and re-evaluation easier and more effective. We must be able to summarize simple 'laws' about system management (thumb-rules) that are not based only on vague experience, but have a theoretical explanation based on reasonable cause and effect.

How could such a thing be done? For instance, how might we measure how much work will be involved in a task?

- We would have to distinguish between the work we actually do and how much work is needed in principle (efficiency and optimization).
- We would look for a mathematical idea with the characteristics or properties of work. We find that we can map work into the idea of ‘information’ content in some cases (now we have something concrete to study).
- Information or work is a statistical concept: information that is transmitted often can be compressed on average—if we do something often, efficiencies can be improved through economies of scale.

By starting down the road of analysis, we gain many small insights that can be assembled into a deeper understanding. That is what this book attempts to do.

The system administrator wonders if he or she will ever become redundant, but there is no sign of that happening. The external conditions and requirements of users are changing too quickly for a system to adapt automatically, and policy has to be adjusted to new goals and crises. Humans are the only technology on the planet that can address that problem for the foreseeable future. Besides, the pursuit of pleasure is a human condition, and part of the enjoyment of the job is that creative and analytical pursuit.

The purpose of this book is to offer a framework in which to analyse and understand the phenomenon of human–computer management. It is only with the help of theoretical models that we can truly obtain a deeper understanding of system behaviour.

Studies

The forthcoming chapters describe a variety of languages for discussing systems, and present some methods and issues that are the basis of the author’s own work. Analysis is the scientific method in action, so this book is about analysis. It has many themes:

1. *Observe*—we must establish a factual basis for discussing systems.
2. *Deduce cause*—we establish probable causes of observed phenomena.
3. *Establish goals*—what do we want from this information?
4. *Diagnose ‘faults’*—what is a fault? It implies a value judgement, based on policy.
5. *Correct faults*—devise and apply strategies.

Again, these concepts are intimately connected with ‘policy’, that is, a specification of right and wrong. In some sense, we need to know the ‘distance’ between what we would like to see and what we actually see.

This is all very abstract. In the day-to-day running of systems, few administrators think in such generalized, abstract terms—yet this is what this book asks you to do.

Example 1 (A backup method) *A basic duty of system administrators is to perform a backup of data and procedures: to ensure the integrity of the system under natural or unnatural threats. How shall we abstract this and turn it into a scientific enquiry?*

We might begin by examining how data can be copied from one place to another. This adds a chain of questions: (i) how can the copying be made efficient? (ii) what does efficient mean? (iii) how often do the data change, and in what way? What is the best strategy for making a copy: immediately after every change, once per day, once per hour? We can introduce a model for the change, for example, a mass of data that is more or less constant, with small random fluctuating changes to some files, driven by random user activity. This gives us something to test against reality. Now we need to know how users behave, and what they are likely to do. We then ask: what do these fluctuations look like over time? Can they be characterized, so that we can tune a copying algorithm to fit them? What is the best strategy for copying the files?

The chain of questions never stops: analysis is a process, not an answer.

Example 2 (Resource management) *Planning a system's resources, and deploying them so that the system functions optimally is another task for a system administrator. How can we measure, or even discuss, the operation of a system to see how it is operating? Can important (centrally important) places be identified in the system, where extra resources are needed, or the system might be vulnerable to failure? How shall we model demand and load? Is the arrival of load (traffic) predictable or stochastic? How does this affect our ability to handle it? If one part of the system depends on another, what does this mean for the efficiency or reliability? How do we even start asking these questions analytically?*

Example 3 (Pattern detection) *Patterns of activity manifest themselves over time in systems. How do we measure the change, and what is the uncertainty in our measurement? What are their causes? How can they be described and modelled? If a system changes its pattern of behaviour, what does this mean? Is it a fault or a feature?*

In computer security, intrusion detection systems often make use of this kind of idea, but how can the idea be described, quantified and generalized, hence evaluated?

Example 4 (Configuration management) *The initial construction and implementation of a system, in terms of its basic building blocks, is referred to as its configuration. It is a measure of the system's state or condition. How should we measure this state? Is it a fixed pattern, or a statistical phenomenon? How quickly should it change? What might cause it to change unexpectedly? How big a change can occur before the system is damaged? Is it possible to guarantee that every configuration will be stable, perform its intended function, and be implementable according to the constraints of a policy?*

In each of the examples above, an apparently straightforward issue generates a stream of questions that we would like to answer. Asking these questions is what science is about; answering them involves the language of mathematics and logic in concert with a scientific inquiry: science is about extracting the essential features from complex observable phenomena and modelling them in order to make predictions. It is based on observation and approximate verification. There is no 'exact science' as we sometimes hear about in connection with physics or chemistry; it is always about suitably idealized approximations to the truth, or ‘uncertainty management’. Mathematics, on the other hand, is not to be confused with science—it is about rewriting assumptions in different ways: that is, if one begins with a statement that is assumed true (an axiom) and manipulates it according to the rules of mathematics, the resulting statement is also true by the same axiom. It contains no more information than the assumptions on which it rests. Clearly, mathematics is an important language for expressing science.

1.6 How to use the text

Readers should not expect to understand or appreciate everything in this book in the short term. Many subtle and deep-lying connections are sewn in these pages that will take even the most experienced reader some time to unravel. It is my hope that there are issues sketched out here that will provide fodder for research for at least a decade, probably several. Many ideas about the administration of systems are general and have been discussed many times in different contexts, but not in the manner or context of system administration.

The text can be read in several ways. To gain a software-engineering perspective, one can replace ‘the system’ with ‘the software’. To gain a business management perspective, replace ‘the system’ with ‘the business’, or ‘the organization’. For human–computer administration, read ‘the system’ as ‘the network of computers and its users’.

The first part of the book is about observing and recording observations about systems, since we aim to take a scientific approach to systems. Part 2 concerns abstracting and naming the concepts of a system’s operation and administration in order to place them into a formal framework. In the final part of the book, we discuss the physics of information systems, that is, the problem of how to model the time-development of all the resources in order to determine the effect of policy. This reflects the cycle of development of a system:

- Observation
- Design (change)
- Analysis.

1.7 Some notation used

A few generic symbols and notations are used frequently in this book and might be unfamiliar.

The function $q(t)$ is always used to represent a ‘signal’ or quality that is varying in the system, that is, a scalar function describing any value that changes in time. I have found it more useful to call all such quantities by the same symbol, since they all have the same status.

$q(x, t)$ is a function of time and a label x that normally represents a spatial position, such as a memory location. In structured memory, composed of multiple objects with finite size, the addresses are multi-dimensional and we write $q(\vec{x}, t)$, where $\vec{x} = (x_1, \ldots, x_\ell)$ is an ℓ-dimensional vector that specifies location within a structured system, for example, (6,3,8) meaning perhaps bit 6 of component 3 in object 8.

In describing averages, the notation $\langle \ldots \rangle$ is used for mean and expectation values, for example, $\langle X \rangle$ would mean an average over values of X. In statistics literature, this is often written as $E(X)$.

In a local averaging procedure, a large set X is reduced to a smaller set x of compounded objects; thus, it does not result in a scalar value but a smaller set whose elements are identified by a new label. For example, suppose we start with a set of 10 values, X. We could find the mean of all values $\langle X \rangle_{10}$ giving a single value. Group them into five groups of two. Now we average each pair and end up with five averaged values: $\langle X(x) \rangle_2$. This still has a label x, since it is a set of values, where $x = 1 \ldots 5$.

Applications and Further Study 1

- *Use these broad topics as a set of themes for categorizing the detailed treatments in forthcoming chapters.*

Chapter 2

Science and its methods

Science is culture,

Technology is art.

—Author's slogan.

A central theme of this book is the application of scientific methodologies to the design, understanding and maintenance of human–computer systems. Ironically, 'Computer Science' has often lacked classical scientific thinking in favour of reasoned assertion, since it has primarily been an agent for technology and mathematics. The art of observation has concerned mainly those who work with performance analysis.

While mathematics is about reasoning (it seeks to determine logical relationships between assumed truths), the main purpose of science is to interpret the world as we see it, by looking for suitably idealized descriptions of observed phenomena and quantifying their uncertainty. Science is best expressed with mathematics, but the two are independent. There are many philosophies about the meaning of science, but in this book we shall be pragmatical rather than encyclopedic in discussing these.

2.1 The aim of science

Let us define science in a form that motivates its discussion in relation to human–computer systems.

Principle 1 (Aim of science) *The principal aim of science is to uncover the most likely explanation for observable phenomena.*

Science is a procedure for making sure that we know what we are talking about when discussing phenomena that occur around us. It is about managing our uncertainty. Science does not necessarily tell us what the correct explanation for a phenomenon is, but it provides us with tools for evaluating the likelihood that a given explanation is true, given certain experimental conditions. Thus, central to science is the act of observation.

Observation is useless without interpretation, so experiments need theories and models to support them. Moreover, there are many strategies for understanding observable phenomena: it is not necessary to have seen a phenomenon to be able to explain it, since we can often predict phenomena just by guesswork, or imagination[1]. The supposed explanation can then be applied and tested once the phenomenon has actually been observed.

The day-to-day routine of science involves the following themes, in approximately this order:

Observation of phenomena

Normally, we want to learn something about a system, for example, find a pattern of behaviour so that we might predict how it will behave in the future, or evaluate a property so that we can make a choice or a value judgement about it. This might be as simple as measuring a value, or it might involve plotting a set of values in a graph against a parameter such as time or memory.

Example 5 *Performance analysts measure the rate at which a system can perform its task. They do this with the larger aim of making things faster or more efficient. Computer anomaly detectors, on the other hand, look for familiar patterns of behaviour so that unusual occurrences can be identified and examined more closely for their significance.*

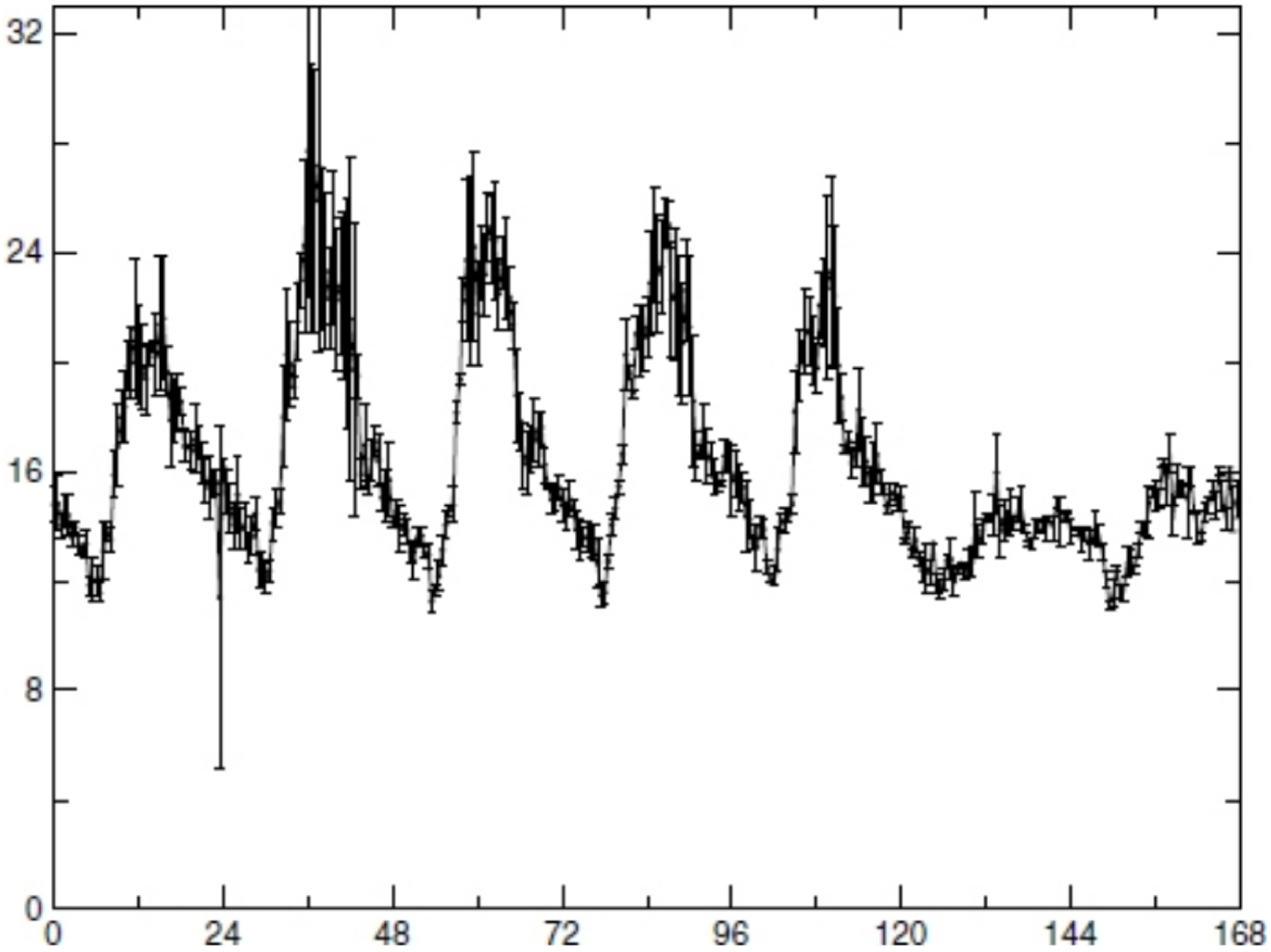
Estimation of experimental error

In observing the world, we must be cautious about the possibility of error in procedure and interpretation: if we intend to base decisions on observations, we need to know how certain we are of our basis. Poor data can mislead (garbage in; garbage out). Any method of observation admits the possibility of error in relation to one's assumptions and methods.

- We make a mistake in measurement (either at random or repeatedly).
- The measuring apparatus might be unreliable.
- The assumptions of the experiment are violated (e.g. inconstant environmental conditions).

Although it is normal to refer to this as 'experimental error', a better phrase is *experimental uncertainty.* We must quantify the uncertainty in the experimental process itself, because this contributes an estimation of how correct our speculations about the results are. Uncertainties are usually plotted as 'error bars' (see fig. 2.1).

Figure 2.1: A pattern of process behaviour. The solid curve is the measured expectation value of the behaviour for that time of week. The error bars indicate the standard deviation, which also has a periodic variation that follows the same pattern as the expectation value; that is, both moments of the probability distribution of fluctuations has a daily and a weekly period.



Identification of relationships

Once we know the main patterns of behaviour, we try to quantify them by writing down mathematical relationships. This leads to empirical relationships between variables, that is, it tells us how many of the variables we are able to identify are *independent*, and how many are *determined.*

Example 6 *It is known that the number of processes running on a college web server is approximately a periodic function (see fig. 2.1). Using these observations, we could try to write down a mathematical relationship to describe this. For example,*

$$f(t) = A + Be^{-\gamma(t-t_0)} \sin(\omega t), \tag{2.1}$$

where t is time along the horizontal axis, and f(t) is the value on the vertical axis, for constants A, B, ω, γ, t_0.

In the example above, there are far too many parameters to make a meaningful fit. It is always possible to fit a curve to data with enough parameters ('enough parameters to fit an elephant' is a common phrase used to ridicule students); the question is how many are justified before an alternative explanation is warranted?

Speculation about mechanisms

Expressing observations in algebraic form gives us a clue about how many parameters are likely to lie behind the explanation of a phenomenon. Next, we speculate about the plausible explanations that lead to the phenomena, and formulate a theory to explain the relationships. If our theory can predict the relationships and the data we have provided, it is reasonable to call the speculation a *theory.*

Confirmation of speculations

One must test a theory as fully as possible by comparing it to existing observations, and by pushing both theory and observation to try to predict something that we do not already know.

Quantification of uncertainty

In comparing theory and observation, there is much uncertainty. There is a basic uncertainty in the data we have collected; then there is a question of how accurately we expect a theory to reproduce those data.

Example 7 *Suppose the formula above for fig. 2.1, in eqn. (2.1) can be made to reproduce the data to within 20% of the value on either side, that is, the approximate form of the curve is right, but not perfect. Is this an acceptable description of the data? How close do we have to be to say that we are close enough? This 'distance from truth' is our uncertainty.*

In a clear sense, science is about uncertainty management. Nearly all systems of interest (and every system involving humans) are very complex and it is impossible to describe them fully. Science's principal strategy is therefore to simplify things to the point where it is possible to make some concrete characterizations about observations. We can only do this with a certain measure of uncertainty. To do the best job possible, we need to control those uncertainties. This is the subject of the next chapter.

2.2 Causality, superposition and dependency

In any dynamical system in which several processes can coexist, there are two possible extremes:

- Every process is independent of every other. System resources change additively (linearly) in response to new processes.
- The addition of each new process affects the behaviour of the others in a non-additive (non-linear) fashion.

The first case is called *superposition*, that is, that two processes can coexist without interfering. This is not true or possible in general, but it can be a useful viewpoint for approximating some system regimes. The latter case is more general and often occurs when a system reaches some limitation, or constraint on its behaviour, such as when there is contention over which process has the use of critical resources.

The principle of causality governs all systems at a fundamental level. It is simply stated as follows:

Principle 2 (Causality) *Every change or effect happens in response to a cause, which precedes it.*

This principle sounds intuitive and even manifestly obvious, but the way in which cause and effect are related in a dynamical system is not always as clear as one might imagine. We would often like to be able to establish a causal connection between a change of a specific parameter and the resulting change in the system. This is a central skill in fault finding, for instance; however, such causal links are very difficult to determine in complex systems. This is one of the reasons why the administration of systems is hard.

2.3 Controversies and philosophies of science

Science and philosophy have long been related. Indeed, what we now call science was once ‘natural philosophy’, or pondering about the natural world. Those who practice science today tend to think little about its larger meaning, or even its methodology. Science has become an ‘industry’—the high ideals that were afforded to it in the seventeenth century have since been submerged in the practicalities of applying it to real problems.

Here are some assertions that have been made of science by philosophers (Horgan (1996)):

- ‘Science cannot determine the truth of an explanation, only its likelihood’.
- ‘Science can only determine the falsity of a theory, not whether it is true’.
- ‘We must distinguish between truth, which is objective and absolute, and certainty which is subjective’.

To the casual technologist, such assertions are likely to draw only scepticism as to the value of philosophy. However, those willing to reflect more deeply on the whole investigative enterprise will find many ideas in the philosophy of science that are both interesting and of practical importance. The difficulty in presenting the labours of careful thought in such a brief and summarized form is that it is easy to misrepresent the philosophers’ detailed arguments[2]. No doubt they would be horrified by this summary if they were alive to read it.

One of the first modern philosophers of science was Sir Francis Bacon, of the sixteenth century. Bacon (who died of pneumonia after stuffing a chicken with ice to see if it would preserve its flesh—thus anticipating the deep freeze) maintained that the task of science is to uncover a thing's character, by noting the presence or the absence of telltale qualities. Thus, to understand heat, for instance, we must examine a list of hot and cold things and discern what features are relevant and irrelevant to the production of heat; for example, exposure to sunlight is relevant, but the width of an object is not. Next, we would examine instances in which a phenomenon is present in varying degrees, noting what circumstances also vary. For example, to understand heat, we must observe things at different temperatures and note what circumstances are present in varying degrees. Bacon recognized that we cannot examine an endless number of instances: at some point we must stop and survey the instances so far.

Especially in the seventeenth century, philosophy became intertwined with mathematics, or analytical thinking. The philosopher Descartes used geometry for his inspiration as to how best to conduct an impartial inquiry. John Locke, an understudy of Isaac Newton, hoped to draw inspiration from the phenomenal success of Newton's laws of motion and the calculus, and derive an analytical way of addressing a 'method of inquiry'—what, today, we would call a 'scientific method'. His philosophy, now called *empiricism*, implies a reliance on experience as the source of ideas and knowledge.

Newton was a significant source of inspiration to philosophers because, for the first time, his work had made it possible to calculate the outcome of a hypothetical situation that no one had ever observed before, that is, predict the future for idealized physical systems. During the Enlightenment, philosophers even came to believe that scientific inquiry could yield truths about human nature and thus that ethical principles might be best derived from such truths; this would therefore be a basis for a new order of society.

In the eighteenth century, others began to realize that this vision was flawed. David Hume discovered an important twist, namely that predictions about events that are not observed cannot be *proven* to be true or false, not even to be probable, since observation alone cannot see into the future, and cannot attempt to assess the *cause* of a phenomenon. He asserted that there are two sources of knowledge: analytical knowledge that is certain (provable assertions) but which cannot directly represent reality, and empirical knowledge or observations that are uncertain but which apply to the real world.

The empirical observation that releasing a stone causes it to fall to the ground is insufficient to prove, beyond doubt, that every stone will always fall to the ground in the future. This is a good example of how our limited experience shapes our view of the world. Before humans went into space, the assertion was always true; however, away from gravity, in the weightlessness of space, the observation becomes meaningless. Hume's point is that we do not know what we don't know, so we should not make unwarranted assumptions.

Although Hume's ideas had an impact on philosophy, they were not generally accepted in science. Immanuel Kant and John Stuart Mill made attempts to solve some of Hume's problems. Kant claimed to solve some of them by assuming that certain facts were to be regarded as axioms, that is, articles of faith that were beyond doubt; that is, that one should always set the stage by stating the conditions under which conclusions should be deemed ‘true’.

Kant supposed, moreover, that our perception of the world is important to how we understand it. In what sense are things real? How do we know that we are not imagining everything? Thus, how do we know that there are not many equally good explanations for everything we see? His central thesis was that the possibility of human knowledge presupposes the participation of the human mind. Instead of trying, by reason or experience, to make our concepts match the nature of objects, Kant held that we must allow the structure of our concepts shape our experience of objects.

Mill took a more pragmatic line of inquiry and argued that the truth of science is not absolute, but that its goals were noble; that is, science is a self-correcting enterprise that does not need axiomatic foundations per se. If experience reveals a flaw in its generalities, it can be accommodated by a critical revision of theory. It would eventually deal with its own faults by a process of refinement.

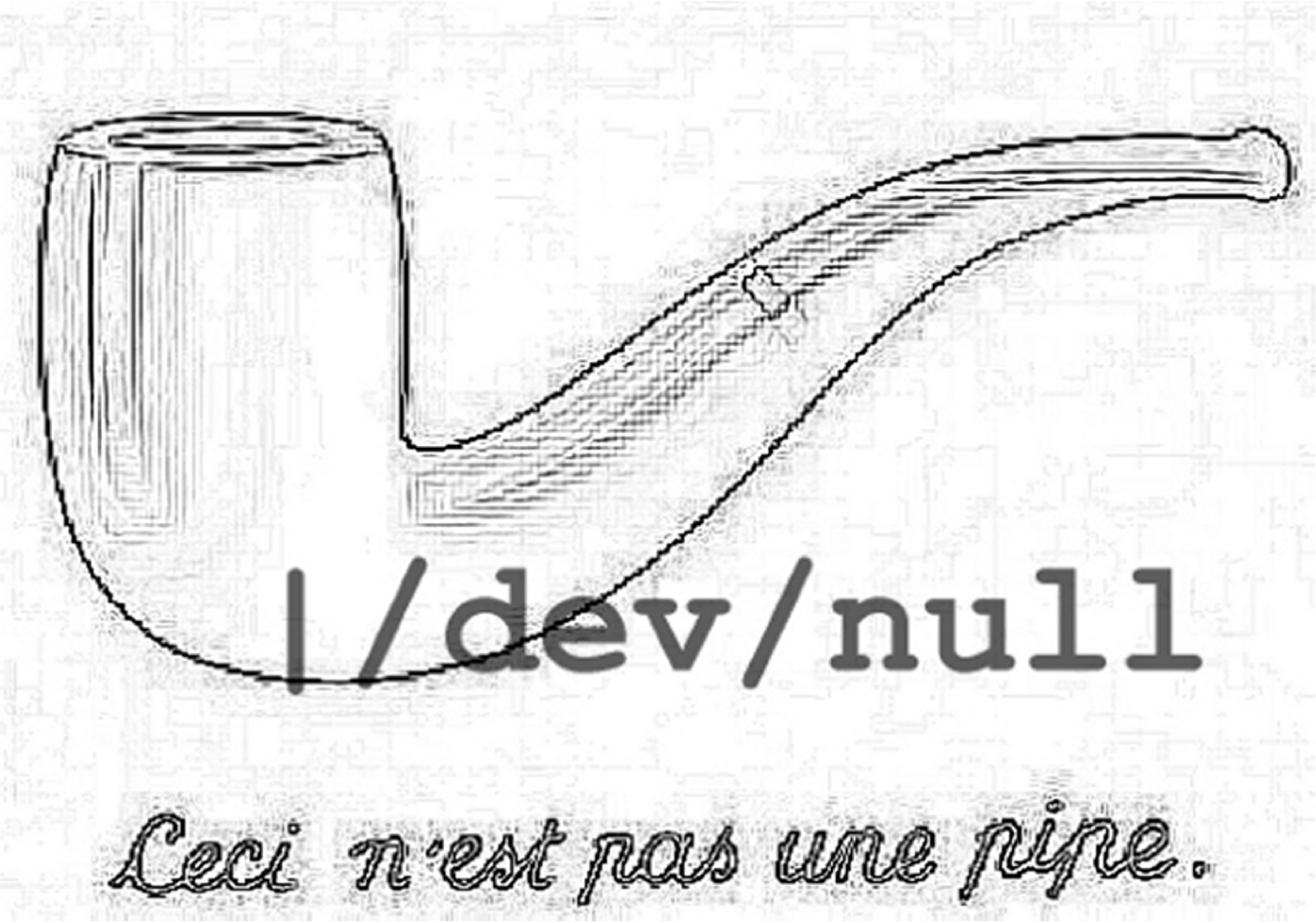
Epistemology is a branch of philosophy that investigates the origins and nature, and the extent of human knowledge. Although the effort to develop an adequate theory of knowledge is at least as old as Plato, epistemology has dominated Western philosophy only since the era of Descartes and Locke, largely as an extended dispute between *rationalism* and *empiricism.* Rationalism believes that some ideas or concepts are independent of experience and that some truth is known by reason alone (e.g. parallel lines never meet). Empiricism believes truth must be established by reference to experience alone.

Logical positivism is a twentieth-century philosophical movement that used a strict principle of *verifiability* to reject non-empirical statements of metaphysics, theology and ethics. Under the influence of Hume and others, the logical positivists believed that the only meaningful statements were those reporting empirical observations; the tautologies of logic and mathematics could not add to these, but merely re-express them. It was thus a mixture of rationalism and empiricism.

The *verifiability principle* is the claim that the meaning of a proposition is no more than the set of observations that would determine its truth, that is, that an empirical proposition is meaningful only if it either actually has been verified or could at least in principle be verified. Analytic statements (including mathematics) are non-empirical; their truth or falsity requires no verification. Verificationism was an important element in the philosophical program of logical positivism.

One of the most influential philosophers of science is Karl Popper. He is sometimes referred to as the most important philosopher of science since Francis Bacon. Karl Popper’s ideas have proven to be widely influential for their pragmatism and their belief in the rational. Popper rejected that knowledge is a social phenomenon—it is absolute. He supposed that we cannot be certain of what we see, but if we are sufficiently critical, we *can* determine whether or not we are wrong, by deductive *falsification* or a process of conjecture and refutation (see fig. 2.2).

Figure 2.2: A pastiche of Rene Magritte's famous painting 'Ceci n'est pas une pipe'. The artist's original paintings and drawings are pictures of a pipe, on which is written the sentence 'this is not a pipe'. The image flirts with paradox and illustrates how uncritical we humans are in our interpretation of things. Clearly the picture is not a pipe—it is a picture that represents a pipe. However, this kind of pedantic distinction is often important when engaging in investigative or analytical thought.



Popper believed that theories direct our observations. They are a part of our innate desire to impose order and organization on the world, that is, to systematize the phenomena we see, but we are easily fooled and therefore we need to constantly criticize and retest every assumption to see if we can *falsify* them. Hume said we can never prove them right, but Popper says that we can at least try to see if they are wrong.

Paul Feyerabend later argued that there is no such thing as an objective scientific method. He argued that what makes a theory true or false is entirely a property of the world view of which that assertion is a part. This is *relativism*, that is, objectivity is a myth. We are intrinsically locked into our own world view, perceiving everything through a particular filter, like a pair of sunglasses that only lets us see particular things.

We need only one flaw in an explanation to discount it; but we might need to confirm hundreds of facts and details to be sure about its validity, that is, 'truth'. In the context of this book, science itself is a system that we shall use to examine others. We summarize with a pragmatic view of science:

Principle 3 (Controlled environment) *Science provides an impartial method for investigating and describing phenomena within an idealized environment, under controlled conditions.*

2.4 Technology

Science, we claim, is an investigative enterprise, whose aim is to characterize what is already there. Technology, on the other hand, is a creative enterprise: it is about tool-building.

The relationship between science and technology is often presented as being problematical by technologists, but it is actually quite clear. If we do not truly understand how things work and behave, we cannot use those things to design tools and methods. In technology, we immediately hit upon an important application of science, namely its role in making *value judgements.* A value judgement is a subjective judgement, for example, one tool can be better than another, one system or method can be better than another—but how are such judgements made? Science cannot answer these questions, but it can assist in evaluating them, if the subjectivity can be defined clearly.

The situation is somewhat analogous to that faced by the seventeenth century philosophers who believed that ethics could be derived from scientific principles. Science cannot tell us whether a tool or a system is ‘good’ or ‘bad’, because ‘good’ and ‘bad’ have no objective definitions. Science craves a discipline in making assertions about technology, and perhaps even guides us in making improvements in the tools we make, by helping us to clarify our own thoughts by quantification of technologies.

2.5 Hypotheses

Although science sometimes springs from serendipitous discovery, its systematic content comes from testing existing ideas or theories and assertions. Scientific knowledge advances by undertaking a series of studies, in order to either verify or falsify a hypothesis. Sometimes these studies are theoretical, sometimes they are empirical and frequently they are a mixture of the two. *Statistical reproducibility* is an important criterion for any result, otherwise it is worthless, because it is *uncertain.* We might be able to get the same answer twice by accident, but only repeated verification can be trusted.

In system administration, software tools and human methods form the technologies that are used. Progress in understanding is made with the assistance of the tools only if investigation leads to a greater predictive power or a more efficient solution to a problem.

- Scientific progress is the gradual refinement of the conceptual model that describes the phenomenon we are studying. In some cases, we are interested in modelling tools. Thus, technology is closely related to science.

- Technological progress is the gradual creative refinement of the tools and methods referred to by the technology. In some cases, the goal is the technology itself; in other situations, the technology is only an implement for assisting the investigation.

All problems are pieces of a larger puzzle. A complete scientific study begins with a *motivation*, followed by an *appraisal* of the problems, the construction of a *theoretical model* for understanding or solving the problems and finally an *evaluation* or *verification* of the *approach used* and the *results obtained.* Recently, much discussion has been directed towards finding suitable methods for evaluating technological innovations in computer science as well as to encouraging researchers to use them. Nowadays, many computing systems are of comparable complexity to phenomena found in the natural world and our understanding of them is not always complete, in spite of the fact that they were designed to fulfil a specific task. In short, technology might not be completely predictable, hence there is a need for experimental verification.

2.6 The science of technology

In technology, the act of observation has two goals: (i) to gather information about a problem in order to motivate the design and construction of a technology which solves it, and (ii) to determine whether or not the resulting technology fulfils its design goals. If the latter is not fulfilled in a technological context, the system may be described as faulty, whereas in natural science there is no right or wrong. In between these two empirical bookmarks lies a theoretical model that hopefully connects the two.

System administration is a mixture of science, technology and sociology. The users of computer systems are constantly changing the conditions for observations. If the conditions under which observations are made are not constant, then the data lose their meaning: the message we are trying to extract from the data is supplemented by several other messages that are difficult to separate from one another. Let us call the message we are trying to extract *signal* and the other messages that we are not interested in *noise.* Complex systems are often characterized by very noisy environments.

In most disciplines, one would attempt to reduce or eliminate the noise in order to isolate the signal. However, in system administration, it would be no good to eliminate the users from an experiment, since it is they who cause most of the problems that one is trying to solve. In principle, this kind of noise in data could be eliminated by statistical sampling over very long periods of time, but in the case of real computer systems, this might not be possible since seasonal variations in patterns of use often lead to several qualitatively different types of behaviour that should not be mixed. The collection of reliable data might therefore take many years, even if one can agree on what constitutes a reasonable experiment. This is often impractical, given the pace of technological change in the field.

2.7 Evaluating a system—dependencies

Evaluating a model of system administration is a little bit like evaluating the concept of a bridge. Clearly, a bridge is a structure with many components, each of which contributes to the whole. The bridge either fulfils its purpose in carrying traffic past obstacles or it does not. In evaluating the bridge, should one then consider the performance of each brick and wire individually? Should one consider the aesthetic qualities of the bridge? There might be many different designs, each with slightly different goals. Can one bridge be deemed better than another on the basis of objective measurement? Perhaps only the bridge's maintainer is in a position to gain a feeling for which bridge is the most successful, but the success criterion might be rather vague: a collection of small differences that make the perceptible performance of the bridge optimal, but with no measurably significant data to support the conclusion. These are the dilemmas of evaluating a complex technology.

The options we have for performing experimental studies are as follows:

- Measurements
- Simulations
- User surveys.

with all of the incumbent difficulties which these entail.

Simplicity

Conceptual and practical simplicity are often deemed to be positive attributes of systems and procedures. This is because simple systems are easy to understand and their behaviours are easy to predict. We prefer that systems that perform a function do so predictably.

Evaluation of individual mechanisms

For individual pieces of a system, it is sometimes possible to evaluate the efficiency and correctness of the components. Efficiency is a relative concept and, if used, it must be placed in a context. For example, efficiency of low-level algorithms is conceptually irrelevant to the higher levels of a program, but it might be practically relevant, that is, one must say what is meant by efficiency before quoting results. The correctness of the results yielded by a mechanism/algorithm can be measured in relation to its design specifications. Without a clear mapping of input/output, the correctness of any result produced by a mechanism is a heuristic quality. Heuristics can only be evaluated by experienced users expressing their informed opinions.

2.8 Abuses of science

Science is about constantly asking questions and verifying hypotheses to see if one's world view holds up to scrutiny. However, the authority that science has won is not always been wielded in a benign way. History is replete with illegitimate ideas that have tried to hide behind the reputation of science, by embracing its terminology without embracing its forms.

Marketeers are constantly playing this game with us, inventing scientific names for bells and whistles on their products, or claiming that they are 'scientifically proven' (an oxymoron). By quoting numbers or talking about 'ologies', there are many uncritical forces in the world who manipulate our *beliefs*, assuming that most individuals will not be able to verify them or discount them[3]. In teaching a scientific method, we must be constantly aware of abuses of science.

Applications and Further Study 2 The observation and analysis of systems involves these themes:

- Variables or measurables
- Determinism or causality
- Indeterministic, random or stochastic influences
- Systems and their environments
- Accounting and conservation.

[1] This is how black holes were 'discovered' in astrophysics. It is now believed that there is unambiguous evidence for black holes.

[2] At this point, it would be natural to give a reference to a book in which a nice summary was presented. Alas, I have yet to find a clear exposition of the philosophy of science printed in English.

[3] Eugenics is one classic example where the words and concepts discovered by science were usurped for illegitimate means to claim that certain individuals were genetically superior to others. This was a classic misunderstanding of a scientific concept that was embraced without proper testing or understanding.

Chapter 3

Experiment and observation

Trust, but verify!

—Russian Proverb

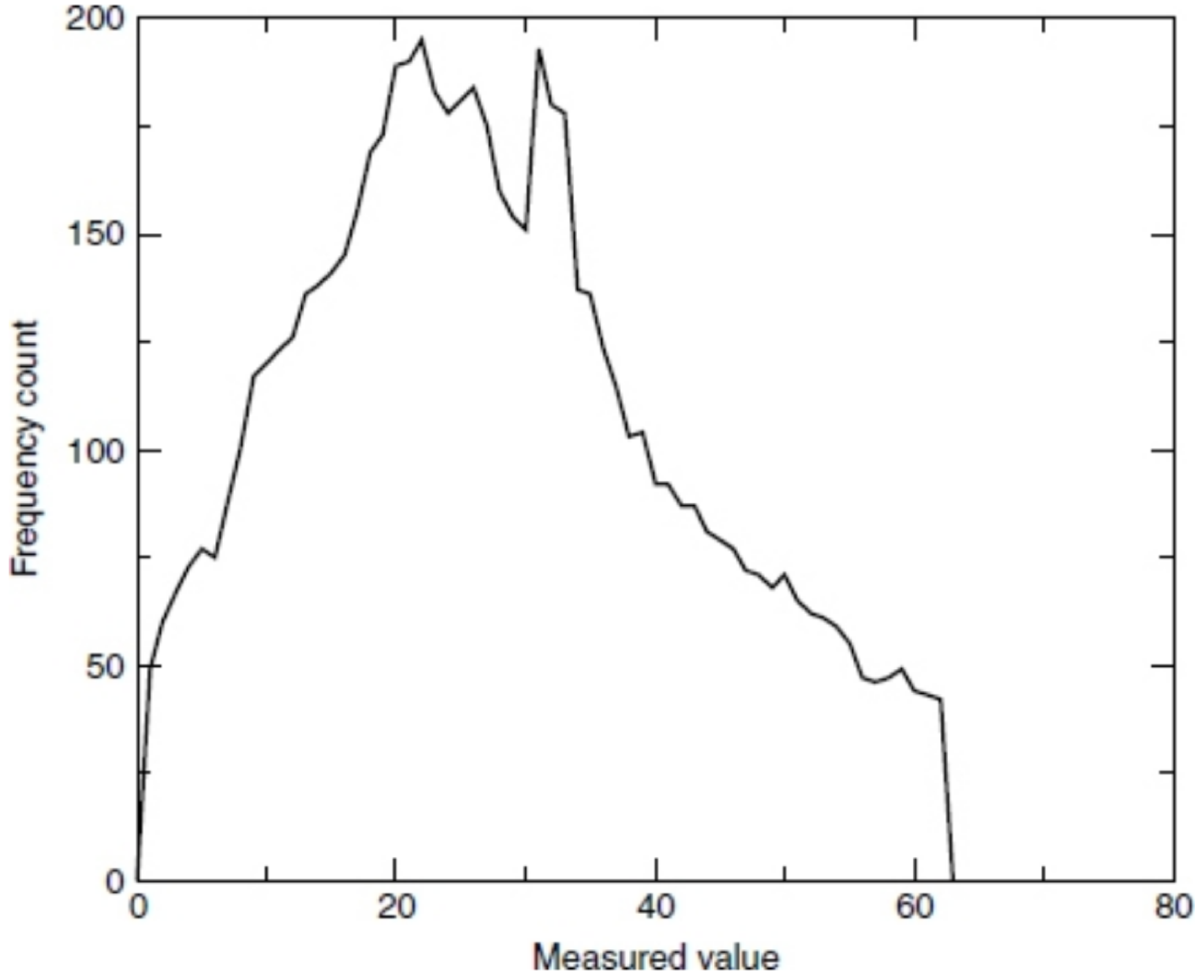
Collecting data to support an idea or a hypothesis is central to the scientific method. We insist on the existence of evidence that can be examined and related analytically (by mathematics or other reasoning) to the phenomenon under consideration, because our trust in random observation or hearsay is only limited. The paraphrased proverb, ‘Trust but verify’ is often cited in connection with system security, but it is equally pertinent here. In a sense, the scientific method is the security or quality assurance system for ‘truth’.

To study human–computer systems, we draw on analytical methods from the diverse branches of science, but our conclusions must be based on observed fact. Reliable observational evidence is most easily obtained where one can perform experiments to gather numerical data, then derive relationships and conclusions. Descriptive sciences do not always have this luxury and are forced to use a form of data collection that involves visual observation, classification or even interview. This is less focused and therefore harder to use to support specific conclusions.

Example 8 *A zoologist might find no problem in measuring the weight of animals, but might find it difficult to classify the colours of animals in order to relate this to their behaviour. When is red really brown? Fuzzy classifiers from day-to-day experience lead to difficulties for science—qualitative descriptions are prone to subjective interpretation.*

Example 9 *In human–computer systems, it is easy to measure numerical quantities such as rate of change of data, but qualitative features such as ‘lawfulness’ of users seem too vague to quantify.*

Difficulties with qualitative characterizations can sometimes be eliminated by going to a lower level, or to a smaller scale of the system: for example, the classification of animals might be done more precisely by looking at their DNA, and the lawfulness of a user might be measured by examining the policy conformance of each file and the changes made by the user.



3.2 Constancy of environment during measurement

In science, our aim is to take small steps, by stripping away everything down to single cause–effect relationships, and then gradually putting things back together. Einstein is famous for having said that everything should be made as simple as possible, but no simpler. By this, he meant that we should neither overcomplicate nor oversimplify an explanation.

Most phenomena are governed by a number of parameters; for example, suppose the rate of a computer is affected by three parameters:

(3.1) $R = R(c, m, s)$

where *c* is the CPU rate, *m* is the amount of memory and *s* is the speed of memory. If we want to discover just how *R* depends on each of these, we must test each parameter individually, holding the others constant, else we might mix up the dependence on each parameter. Science ends up with neat formulae relating measurables, only because this isolation is possible. Such formulae describe the real world, but they do not really describe the 'real environment' because the environment is messy. Science therefore strives to ensure idealized environmental conditions for investigating phenomena, in order to take one thing at a time.

In the real world of human–computer systems, there are many variables and influences that affect a system, so we must strive to maintain constant conditions in all variables but the one we would like to test. This is rarely possible, and thus there is an inevitable *uncertainty* or *experimental error* in any experiment. An important task of science is to quantify this uncertainty.

Principle 4 *Scientific observation strives to isolate single cause–effect relationships, by striving to keep environmental conditions constant during measurement. The impossibility of completely constant external conditions makes it necessary to quantify the uncertainty in each measurement.*

Note that by isolating 'single' cause–effect relationships, we do not mean to imply that there is always a single variable that controls a process, only that each independent change can be identified with an independent parameter.

The way we do this for simple measurable values is relatively easy and is described in this chapter. However, not all situations are so easily quantifiable. Qualitative experiments, such as those of biology (e.g. classifying types of behaviour) also occur in the study of human–computer systems. If we do not actually begin with hard numbers, the estimate of uncertainty has to be made by finding a numerical scale, typically through a creative use of *classification* statistics; for example, how many animals have exhibited behaviour *A* and how many behaviour *B*? Or how far is behaviour *A* from behaviour *B* on some arbitrary scale, used only for comparison?

All scales are arbitrary in science (that is why we have many different units for weight, height, frequency etc.), what is important is how we relate these scales to observables.

3.3 Experimental design

The cleverness of an experiment's design can be crucial to its success in providing the right information. Our aim is to isolate a single channel of cause–effect at a time. We must ensure that the experimental observation does not interfere with the system we are measuring. Often, an experiment yields unexpected obstacles that must be overcome. There can be a lot of work involved in answering even a simple question. (For examples from computer performance analysis, see Jain (1991).)

Example 10 *Suppose we wish to compare the behaviour of two programs for mirroring (copying) files, for backup. We notice that one program seems to complete its task very quickly, presenting a high load to the source and destination machines. The other takes much longer but presents almost no load. How shall we determine the reason?*

We might begin by finding some data to copy. Data is composed of files of *different sizes. Size might be important, so we shall be interested in how size affects the rate of copying, if at all. The first time we copy the files, every file must be transferred in full. On subsequent updates, only changes need to be copied. One program claims to copy only those bytes that are different; the other has to copy a whole file, even if only one byte has changed, so file size again becomes important.*

We could investigate how the total time for copying is related to the total amount of data (i) of all files, (ii) of files that are copied. We might also be interested in what dependencies the programs have: do they use the Internet Protocol with TCP or UDP, IPv4 or IPv6? Does the computer kernel or operating system affect the performance of the two programs?

The stream of questions never ceases; we must decide when to stop. Which questions are we interested in, and when have they been sufficiently answered? This is a value judgement that requires experience and inquisitiveness from the investigator.

3.4 Stochastic (random) variables

Our inability to control, or even follow every variable in a system's environment means that some of the changes appearing in the system seem random, or inexplicable.

Definition 4 (Random process) *A random process is one in which there are too many unknowns to be able to trace the channels of cause and effect.*

A *stochastic* or *random* variable is a variable whose value depends on the outcome of some underlying random process. The range of values of the variable is not at issue, but which particular value the variable has at a given moment is random. We say that a stochastic variable X will have a certain value x with a probability $P(x)$.

Usually, in an experiment, a variable can be said to have a certain random component (sometimes called its *error* from the historical prejudice that science is deterministic and the only source of randomness is the errors incurred by the experimental procedure) and an average stable value. We write this as

$$x = \langle x \rangle + \Delta x, \quad (3.2)$$

where x is the actual value measured, $\langle x \rangle$ is the *mean* or *expectation value* of all measurements (often written $E(x)$ in statistical literature), and Δx is the deviation from the mean. The mean value changes much more slowly than Δx. For example:

- Choices made by large numbers of users are not predictable, except on average.

- Measurements collected over long periods of time are subject to a variety of fluctuating conditions.

Measurements can often appear to give random results, because we do not know all of the underlying mechanisms in a system. We say that such systems are *non-deterministic* or that there are *hidden variables* that prevent us from knowing all the details. If a variable has a fixed value, and we measure it often enough and for long enough, the random components will often fall into a *stable distribution*, by virtue of the *central limit theorem* (see, for instance Grimmett and Stirzaker (2001)). The best-known example of a stable distribution is the Gaussian type of distribution.

3.5 Actual values or characteristic values

There is a subtle distinction in measurement between an observable that has an actual 'true' value and one that can only be characterized by a typical value.

For example, it is generally assumed that the rest mass of an electron has a 'true' value that never changes. Yet when we measure it, we get many different answers. The conclusion must be that the different values result from *errors* in the measurement procedure. In a different example, we can measure the size of a whale and we get many different answers. Here, there is no 'true' or 'standard' whale and the best we can do is to measure a typical or *expected* value of the size.

In human–computer systems, there are few, if any, measurements of the first type, because almost all values are affected by some kind of variation. For example, room temperature can alter the maximum transmission rate of a cable. We must therefore be careful about what we claim to be constant, and what is the reason for the experimental variation in the results. Part of the art in science is in the interpretation of results, within the constraints of cause and effect.

3.6 Observational errors

All measurements involve certain errors. One might be tempted to believe that, where computers are involved, there would be no error in collecting data, but this is false. Errors are not only a human failing; they occur because of unpredictability in the measurement process, and we have already established throughout this book that computer systems are nothing but unpredictable. We are thus forced to make estimates of the extent to which our measurements can be in error. This is a difficult matter, but approximate statistical methods are well known in the natural sciences, methods that become increasingly accurate with the amount of data in an experimental sample.

The ability to estimate and treat errors should not be viewed as an excuse for constructing a poor experiment. Errors can only be minimized by design. There are several distinct types of error in the process of observation.

The simplest type of error is called *random error.* Random errors are usually small deviations from the 'true value' of a measurement that occur by accident, by unforeseen jitter in the system, or by some other influence. By their nature, we are usually ignorant of the cause of random errors, otherwise it might be possible to eliminate them. The important point about random errors is that they are distributed evenly about the mean value of the observation. Indeed, it is usually assumed that they are distributed with an approximately *normal* or *Gaussian* profile about the mean. This means that there are as many positive as negative deviations and thus random errors can be averaged out by taking the mean of the observations.

It is tempting to believe that computers would not be susceptible to random errors. After all, computers do not make mistakes. However, this is an erroneous belief. The measurer is not the only source of random errors. A better way of expressing this is to say that random errors are a measure of the unpredictability of the measuring process. Computer systems are also unpredictable, since they are constantly influenced by outside agents such as users and network requests.

The second type of error is a *personal error.* This is an error that a particular experimenter adds to the data unwittingly. There are many instances of this kind of error in the history of science. In a computer-controlled measurement process, this corresponds to any particular bias introduced through the use of specific software, or through the interpretation of the measurements.

The final and most insidious type of error is the *systematic error.* This is an error that runs throughout all of the data. It is a systematic shift in the true value of the data, in one direction, and thus it cannot be eliminated by averaging. A systematic error leads also to an error in the mean value of the measurement. The sources of systematic error are often difficult to find, since they are often a result of misunderstandings, or of the specific behaviour of the measuring apparatus.

Principle 5 *In a system with finite resources, the act of measurement itself leads to a change in the value of the quantity one is measuring.*

Example 11 *In order to measure the pressure of a bicycle tyre, we have to release some of the pressure. If we continue to measure the pressure, the tyre will eventually be flat.*

In order to measure the CPU usage of a computer system, for instance, we have to start a new program that collects that information, but that program inevitably uses the CPU also and therefore changes the conditions of the measurement. These issues are well known in the physical sciences and are captured in principles such as Heisenberg's Uncertainty Principle, Schrödinger's cat and the use of infinite idealized heat baths in thermodynamics. We can formulate our own verbal expression of this for computer systems:

actually be equal to μ; rather, they are scattered around the average value in some pattern, called their distribution *P*(*x*). The normalization factor is usually chosen so that the area under the curve is unity, giving a probabilistic interpretation.

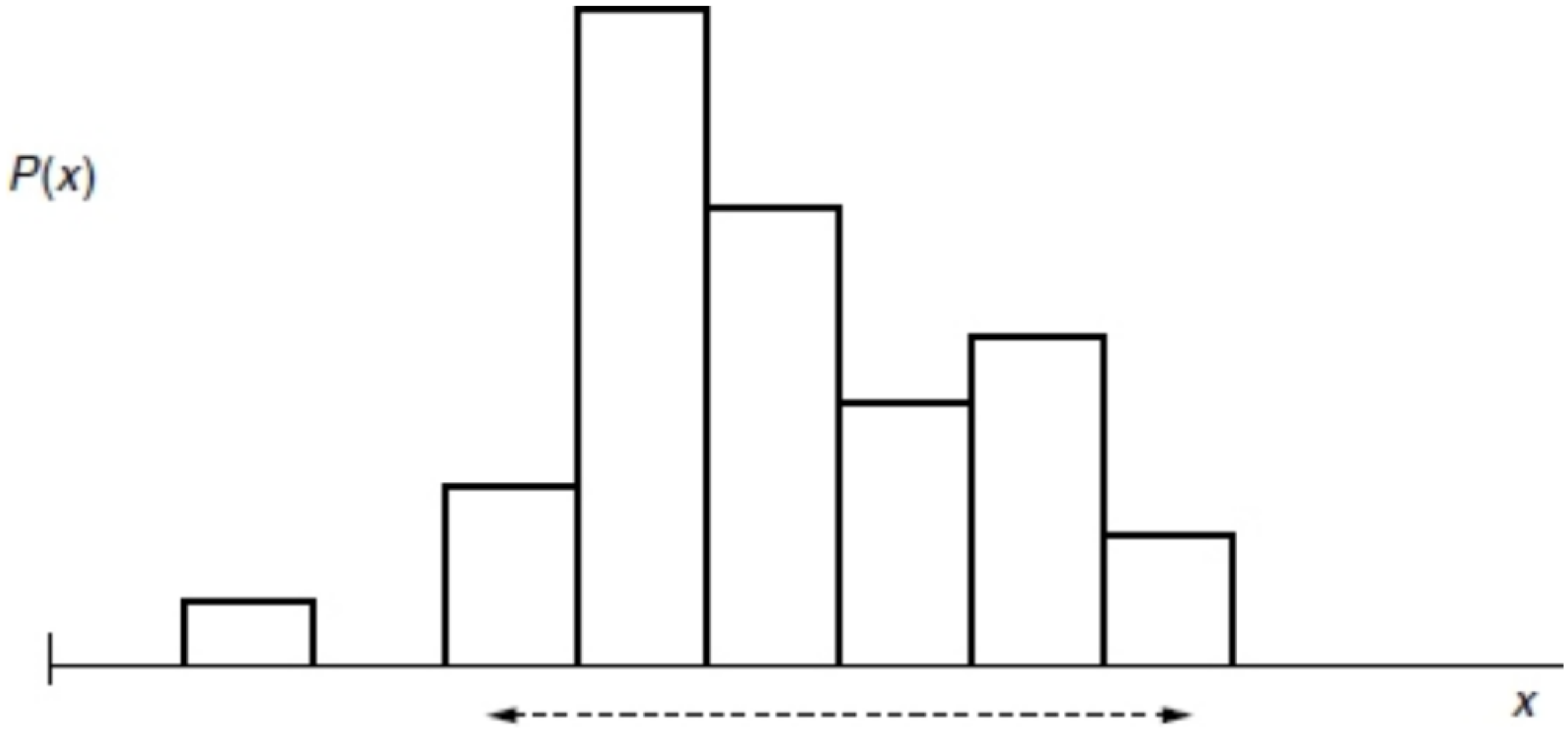
Definition 6 (Probability) *The probability P*(*x*) *of measuring a value x in original data set is defined to be the fraction of values that fell into the range x* ± Δ*x*/2, *for some class width* Δ*x.*

$$P(x) = \frac{N(x - \Delta x/2, x + \Delta x/2)}{N_{\text{total}}}. \tag{3.6}$$

Here, N(*x, y*) *is the number of observations between x and y.*

This probability distribution is the histogram shown in fig. 3.3.

Figure 3.3: The scatter is an estimate of the width of the populated regions of the probability distribution.



There are two extremes of distribution: complete certainty (fig. 3.4) and complete uncertainty (fig. 3.5). If a measurement always gives precisely the same answer, then we say that there is no error. This is never the case in real measurements. Then the curve is just a sharp spike at the particular measured value. If we obtain a different answer each time we measure a quantity, then there is a spread of results. Normally, that spread of results will be concentrated around some more or less stable value (fig. 3.6). This indicates that the probability of measuring that value is biased, or tends to lead to a particular range of values. The smaller the range of values, the closer we approach fig. 3.4. But the converse might also happen: in a completely random system, there might be no fixed value of the quantity we are measuring. In that case, the measured value is completely uncertain, as in fig. 3.5. To summarize, a flat distribution is unbiased, or completely random. A non-flat distribution is biased, or has an expectation value, or probable outcome. In the limit of complete certainty, the distribution becomes a spike, called the *delta distribution.*

Figure 3.4: The delta distribution represents complete certainty. The distribution has a value of 1 at the measured value.

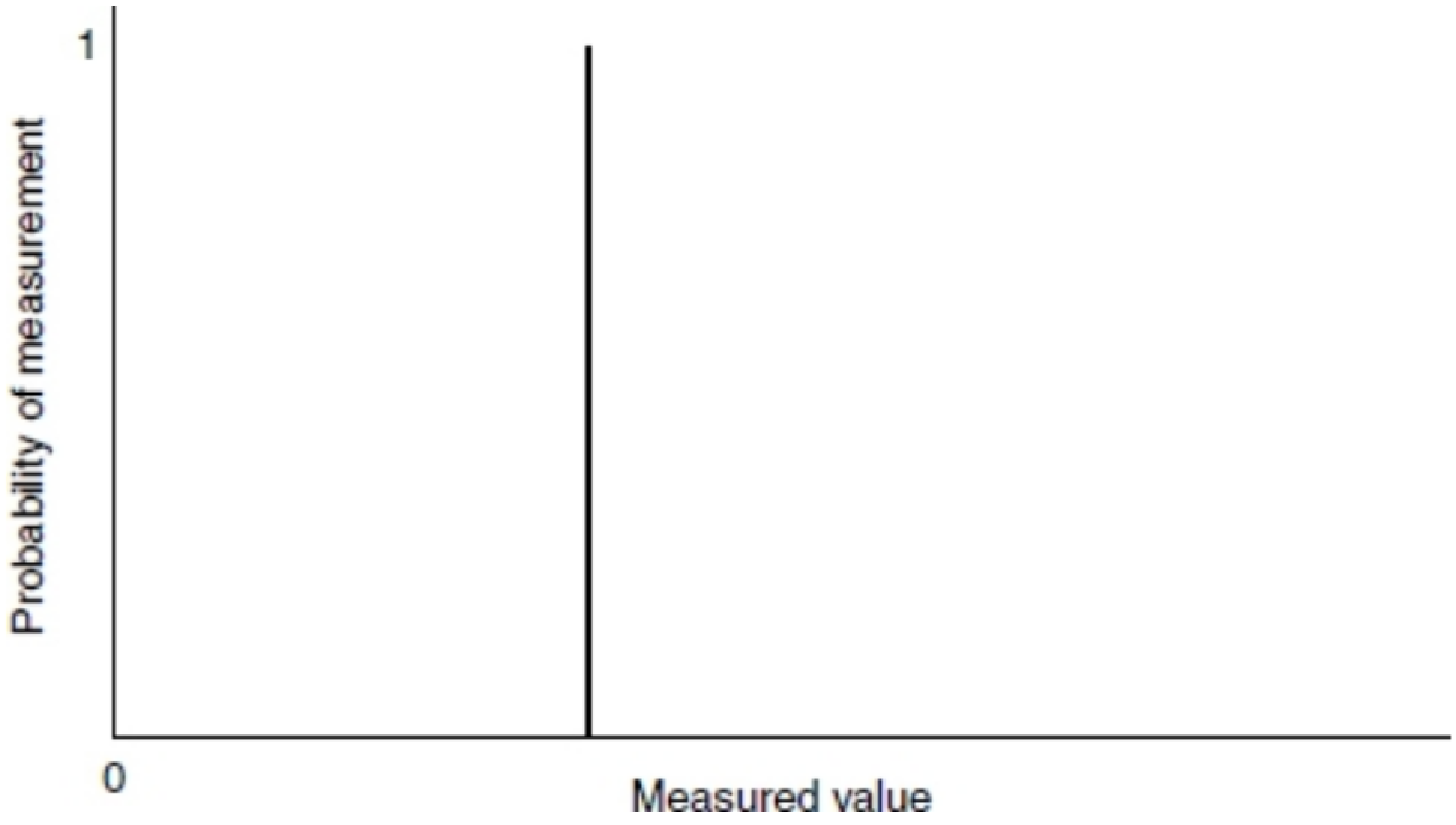


Figure 3.5: The flat distribution is a horizontal line indicating that all measured values, within the shown interval, occur with equal probability.

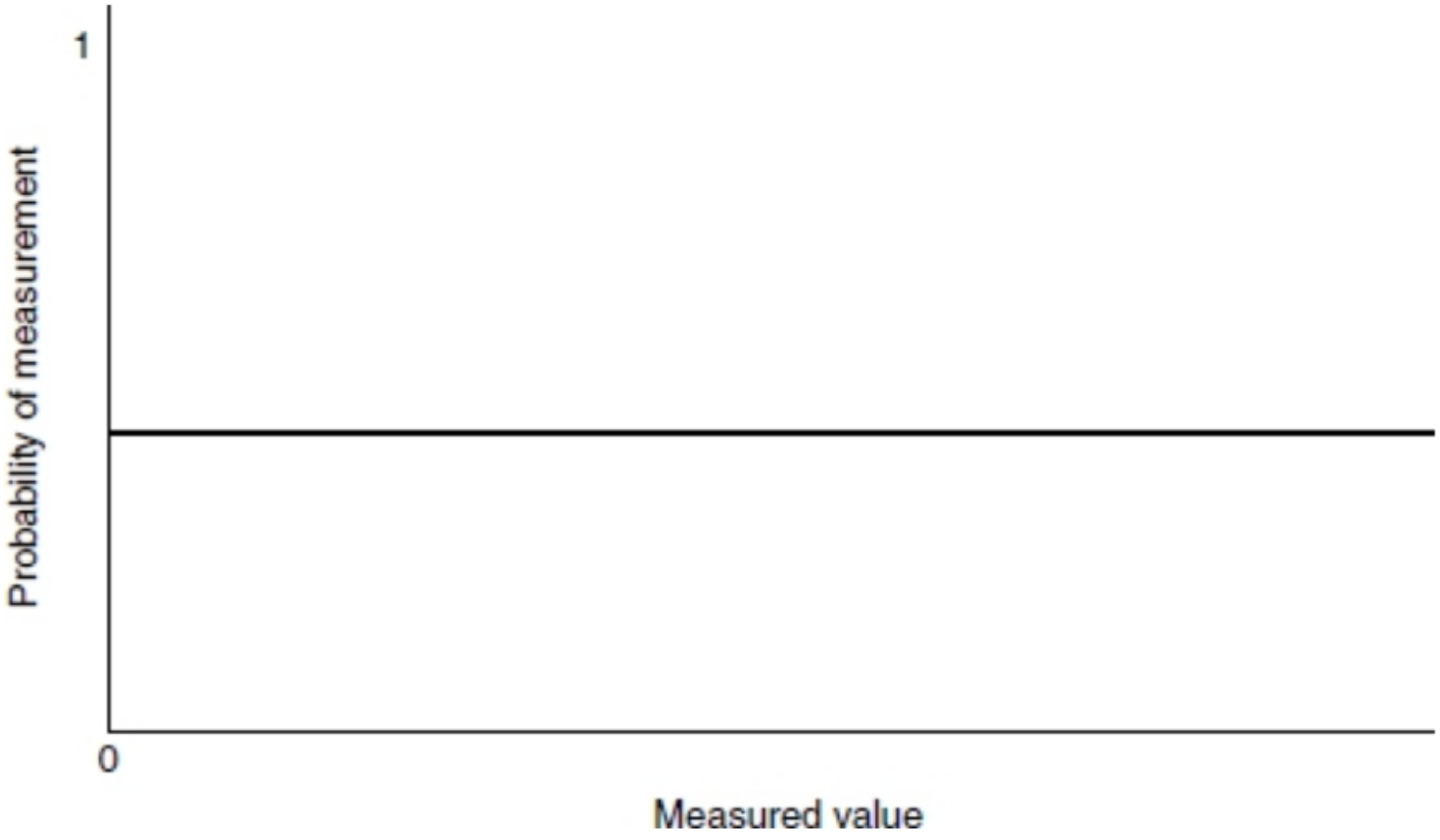
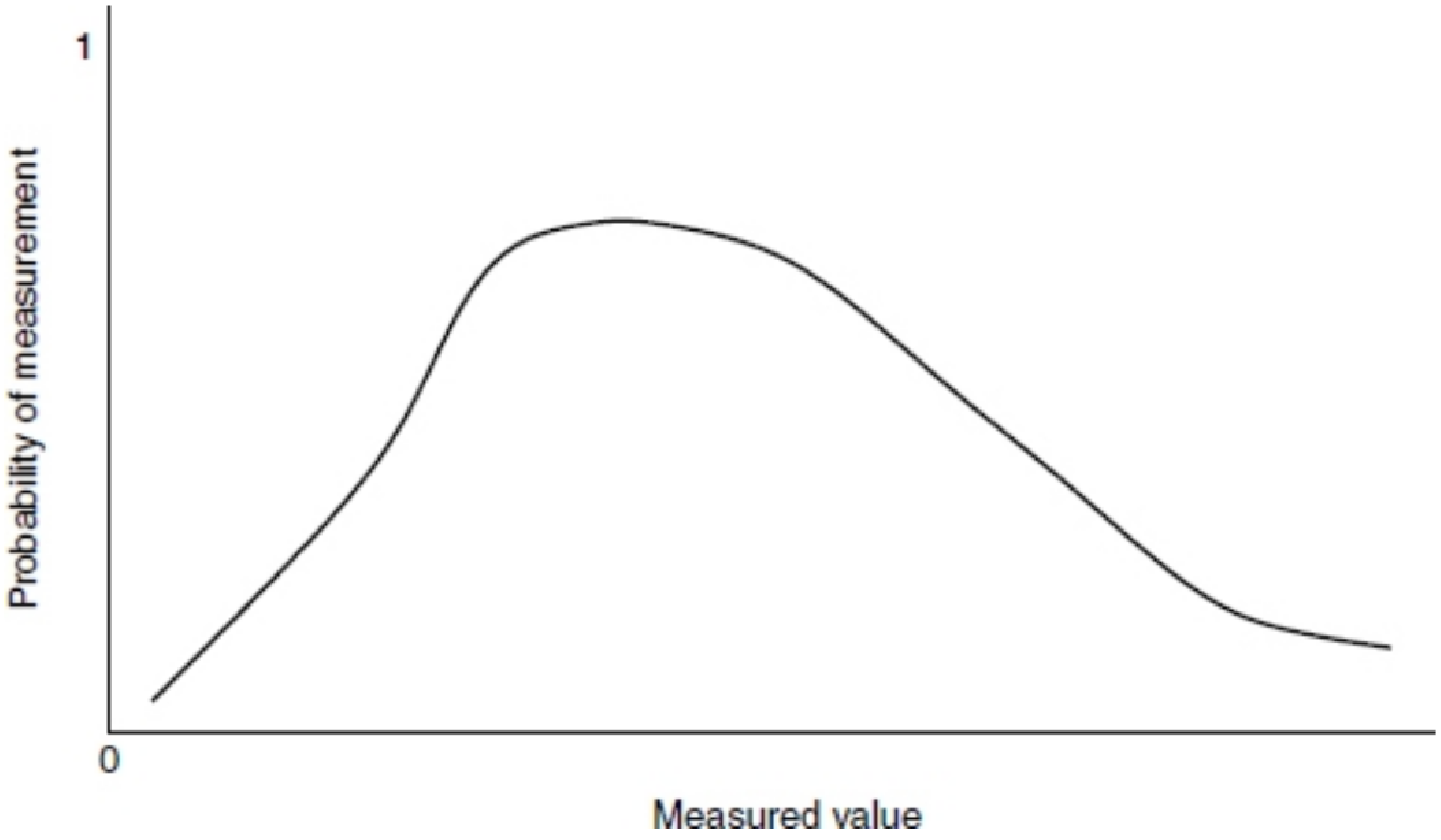


Figure 3.6: Most distributions peak at some value, indicating that there is an expected value (expectation value) that is more probable than all the others.



We are interested in determining the shape of the distribution of values on repeated measurement for the following reason. If the variation of the values is symmetrical about some preferred value, that is, if the distribution peaks close to its mean value, then we can probably infer that the value of the peak or of the mean is the true value of the measurement and that the variation we measured was due to random external influences. If, on the other hand, we find that the distribution is very asymmetrical, some other explanation is required and we are most likely observing some actual physical phenomenon that requires explanation.

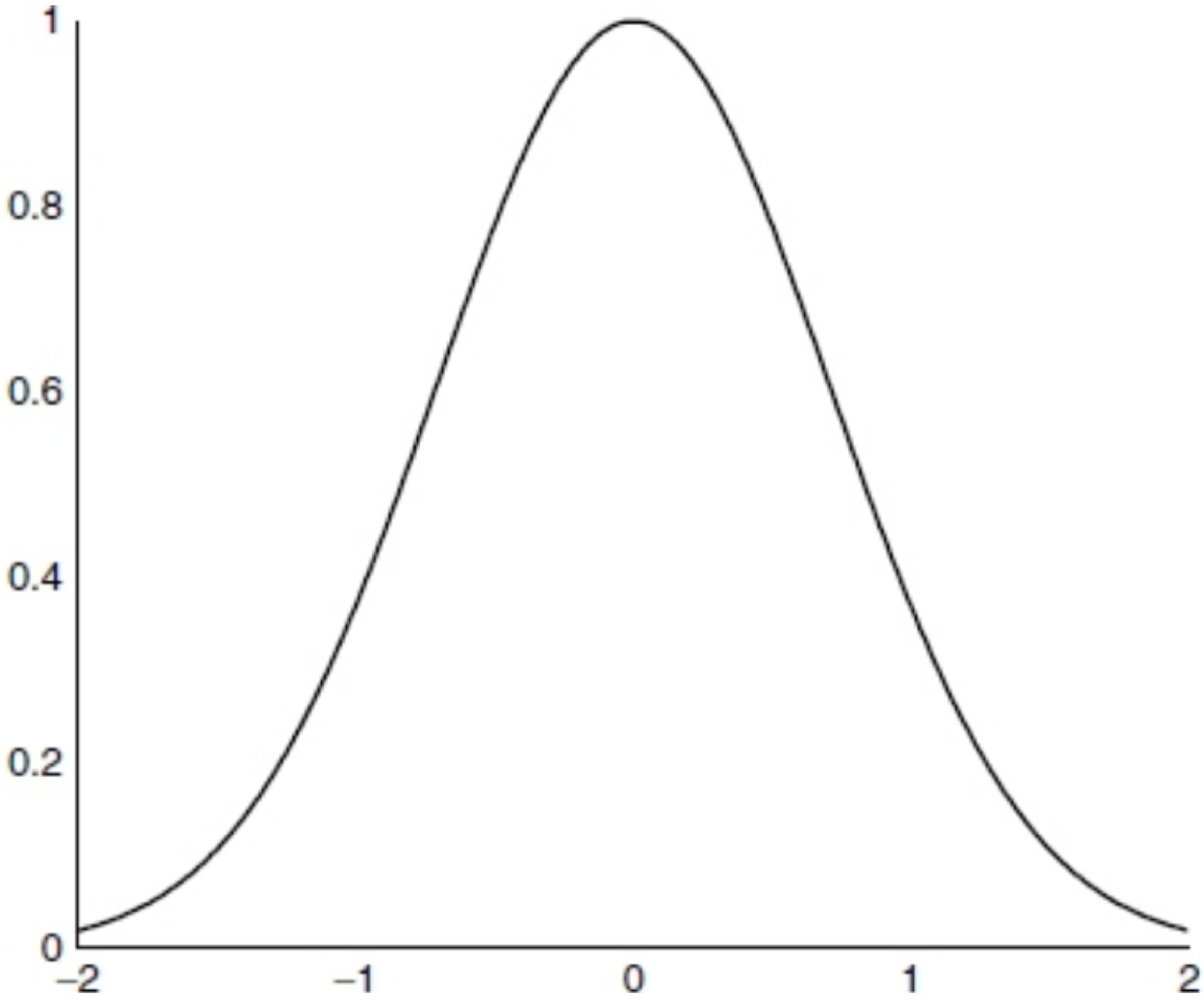
3.8.1 Scatter and jitter

The term *scatter* is often used to express the amount of variation in the measurements about the mean. It is estimated as the 'width' of the histogram $P(x)$. The term *jitter* is often used when describing the scatter of arrival times between measurements in the time series. Decades of artificial courses on statistics have convinced many scientists that the distribution of points about the mean must follow a Gaussian 'normal' distribution in the limit of large numbers of measurements. This is not true, however; there are ample cases in which the scatter is asymmetric or less uniform than the 'normal distribution'.

3.8.2 The 'normal' distribution

It has been stated that 'Everyone believes in the exponential law of errors; the experimenters because they think it can be proved by mathematics; and the mathematicians because they believe it has been established by observation' (Whittaker and Robinson (1929)). Some observational data in science closely satisfy the normal law of error, but this is by no means universally true. The main purpose of the normal error law is to provide an adequate idealization of error treatment that applies to measurements with a 'true value' (see section 3.5), which is simple to deal with, and which becomes increasingly accurate with the size of the data sample.

Figure 3.7: The Gaussian normal distribution, or bell curve, peaks at the arithmetic mean. Its width characterizes the standard deviation. It is therefore the generic model for all measurement distributions.



The normal distribution was first derived by DeMoivre in 1733 while dealing with problems involving the tossing of coins; the law of errors was deduced theoretically in 1783 by Laplace. He started with the assumption that the total error in an observation was the sum of a large number of independent deviations, which could be either positive or negative with equal probability, and could therefore be added according to the rule explained in the previous sections. Subsequently, Gauss gave a proof of the error law based on the postulate that the most probable value of any number of equally good observations is their arithmetic mean. The distribution is thus sometimes called the Gaussian distribution, or the bell curve.

The Gaussian normal distribution is a smooth curve that is used to model the distribution of discrete points distributed around a mean. The probability density function *P*(*x*) tells us with what probability we would expect measurements to be distributed about the mean value $\bar{x}$ (see fig. 3.7).

$$P(x_i) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{(x_i - \overline{x})^2}{2\sigma^2}\right).$$

It is based on the idealized limit of an infinite number of points.

3.8.3 Standard error of the mean

No experiments have an infinite number of points, so we need to fit a finite number of points to a normal distribution as well as we can. It can be shown that the most probable choice is to take the mean of the finite set to be our estimate of the mean of the ideal set. Of course, if we select at random a sample of *N* values from the idealized infinite set, it is not clear that they will have the same mean as the full set of data. If the number in the sample *N* is large, the two will not differ greatly, but if *N* is small, they might. In fact, it can be shown that if we take many random samples of the ideal set, each of size *N*, they will have mean values that are themselves normally distributed, with a standard deviation equal to σ/√*N.* The quantity

$$\alpha = \frac{\sigma}{\sqrt{N}},$$

where σ is the standard deviation, is therefore called the *standard error of the mean.* This is clearly a measure of the accuracy with which we can claim that our finite sample mean agrees with the actual mean. In quoting a measured value *which we believe has a unique or correct value* (e.g. the height of the Eiffel Tower), it is therefore normal to write the mean value, plus or minus the standard error of the mean:

$$\text{(3.7)}\quad \text{Result} = \overline{x} \pm \sigma/\sqrt{N} \text{ (for } N \text{ observations)},$$

where *N* is the number of measurements. Otherwise, if we believe that the measured value should have a distribution of values (e.g. the height of a river on the first of January of each year), one uses the standard deviation as a measure of the error. Many transactional operations in a computer system do not have a fixed value (see next section).

The law of errors is not universally applicable, without some modification, but it is still almost universally applied, for it serves as a convenient fiction that is mathematically simple[1].

3.8.4 Other distributions

Another distribution that appears in the periodic rhythms of system behaviour is the exponential form. There are many exponential distributions, and they are commonly described in textbooks. Exponential distributions are used to model component failures in systems over time, that is, most components fail quickly or live for a long time.

The Planck distribution is one example that can be derived theoretically as the

1046 ~ 200*) being administrative transactions, and the time for measurement was* 200 *seconds, give or take a few milliseconds, then:*

$$\begin{aligned}\Delta S &= \sqrt{(1/200)^2 \times 200^2 + (1046/4000)^2 \times 0.001^2},\\ &\simeq \frac{200}{200},\\ &\simeq 1.\end{aligned} \tag{3.14}$$

Thus, we quote the value for S to be

$$S = 1046/200 \pm 1 = 523 \pm 1. \tag{3.15}$$

Note that this is an estimate based on a continuum approximation, since N and T are both discrete, non-differentiable quantities. As we are only estimating, this is acceptable.

3.10 Fourier analysis and periodic behaviour

Many aspects of computer system behaviour have a strong periodic quality, driven by the human perturbations introduced by users' daily rhythms. Other natural periods follow from the largest influences on the system from outside. For instance, hourly updates, or automated backups. The source might not even be known: for instance, a potential network intruder attempting a stealthy port scan might have programmed a script to test the ports periodically over a length of time. Analysis of system behaviour can sometimes benefit from knowing these periods. For example, if one is trying to determine a causal relationship between one part of a system and another, it is sometimes possible to observe the signature of a process that is periodic and thus obtain direct evidence for its effect on another part of the system.

Periods in data are the realm of Fourier analysis. What a Fourier analysis does is to assume that a data set is built up from the superposition of many periodic processes. Any curve can be represented as a sum of sinusoidal-waves with different frequencies and amplitudes. This is the complex Fourier theorem:

$$f(t) = \int d\omega\, f(\omega) e^{-i\omega t},$$

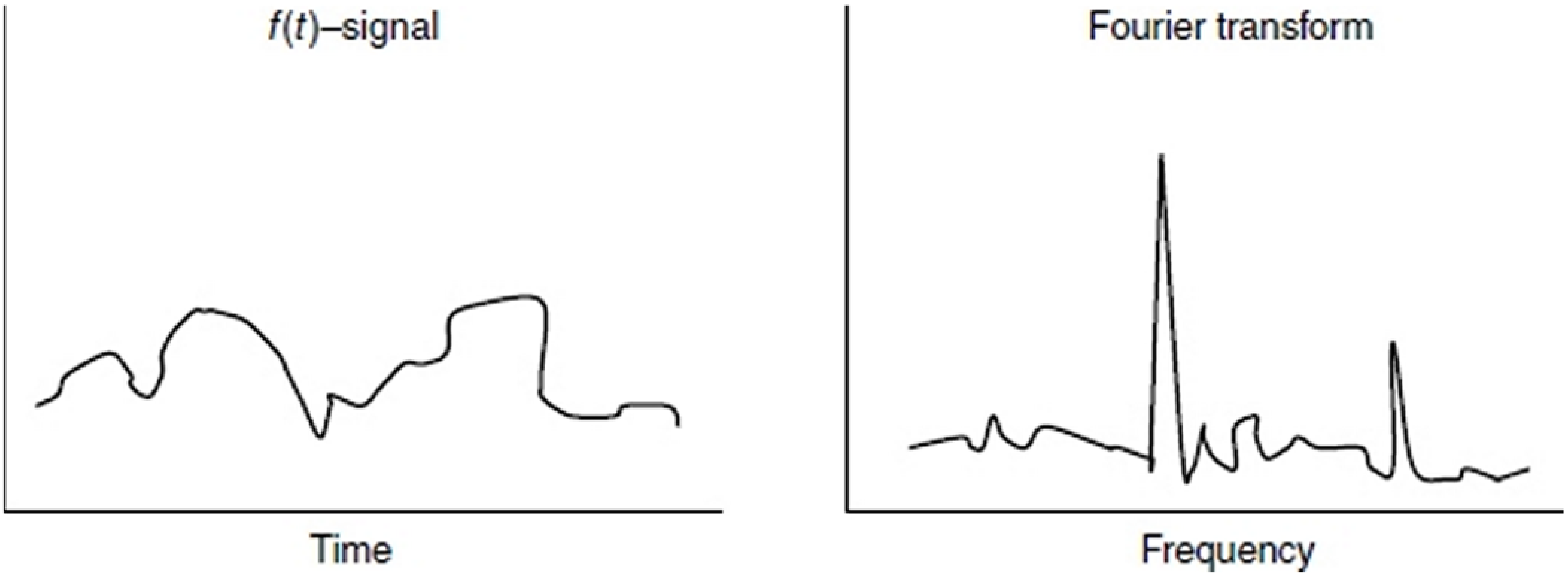
where *f*(ω) is a series of coefficients. For strictly periodic functions, we can represent this as an infinite sum:

$$f(t) = \sum_{n=0}^{\infty} c_n e^{-2\pi i\, nt/T},$$

where *T* is some timescale over which the function *f*(*t*) is measured. What we are interested in determining is the function *f*(ω), or equivalently the set of coefficients c_n that represent the function. These tell us how much of which frequencies are present in the signal *f*(*t*), or its *spectrum.* It is a kind of data prism, or spectral analyser, like the graphical displays one finds on some music players. In other words, if we feed in a measured sequence of data and Fourier-analyse it, the spectral function show the frequency content of the data that we have measured.

The whys and wherefores of Fourier analysis are beyond the scope of this book; there are standard programs and techniques for determining the series of coefficients. What is more important is to appreciate its utility. If we are looking for periodic behaviour in system characteristics, we can use Fourier analysis to find it. If we analyse a signal and find a spectrum such as the one in fig. 3.9, then the peaks in the spectrum show the strong periodic content of the signal. To discover these smaller signals, it will be necessary to remove the louder ones (it is difficult to hear a pin drop when a bomb explodes nearby).

Figure 3.9: Fourier analysis is like a prism, showing us the separate frequencies of which is signal is composed. The sharp peaks in this figure illustrate how we can identify periodic behaviour that might otherwise be difficult to identify. The two peaks show that the input source conceals two periodic signals.



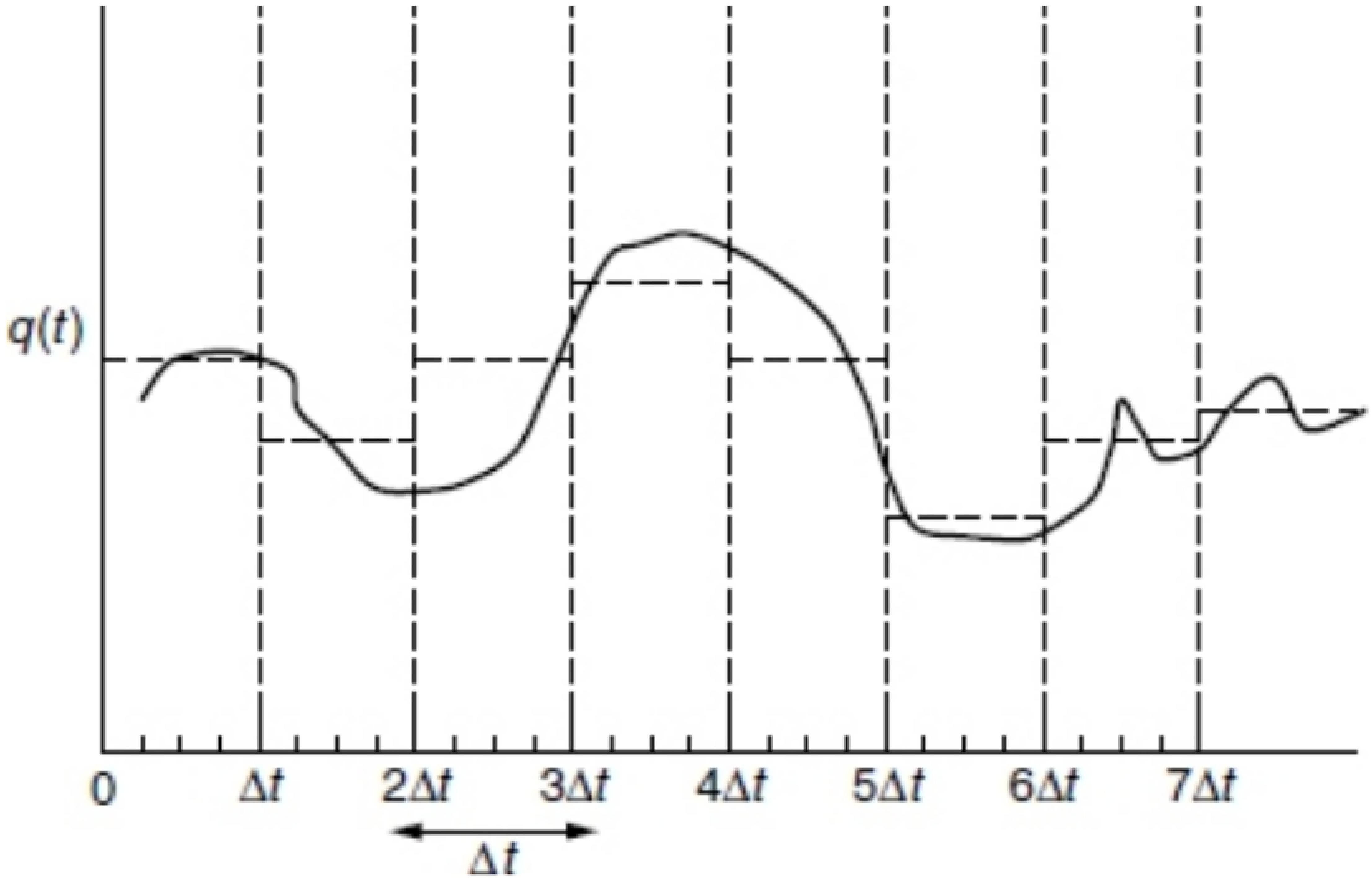
3.11 Local averaging procedures

One of the most important techniques for analysing data in time series is that of coarse graining, or local averaging. This is a smoothing procedure in which we collect together a sequence of measurements from a short interval of time Δ*t* and replace them with a single average value for that interval. It is a way of smoothing out random fluctuations in data and extracting the trends. It also used as a way of characterizing the pattern of change in a time series.

Computer systems and human systems have often quite different patterns of behaviour. When they are combined, the result is often complex and hence local averaging is a straightforward approach to extracting or suppressing detail about the signal.

Let us define a local averaging procedure using fig. 3.10. See also Appendix B for more details.

Figure 3.10: A coarse-graining, or local averaging procedure involves averaging over intervals larger than the basic resolution of the data. The flat horizontal lines represent the coarse-grained histogrammatic representation of the function. The scaling hypothesis say that if one ‘zooms out’ far enough, and views the fundamental and coarse-grained representations from a sufficiently high level ($\delta t \gg \Delta t$), then they are indistinguishable for all calculational purposes.



The local averaging procedure re-averages data, moving from a detailed view to a less detailed view, by grouping neighbouring data together. In practice, one always deals with data that are sampled at discrete time intervals, but the continuous time case is also important for studying the continuum approximation to systems.

Discrete time data

Consider the function $q(t)$ shown in figs. 3.10 and 3.11. Let the small ticks on the horizontal axis represent the true sampling of the data, and label these by $i = 0, 1, 2, 3, \ldots, I$. These have unit spacing. Now let the large ticks, which are more coarsely spread out, be labelled by $k = 1, 2, 3, \ldots, K$. These have spacing $\Delta t = m$, where m is some fixed number of the smaller ticks. The relationship between the small and the larger ticks is thus

$$i = (k - 1)\Delta t = (k - 1)m. \tag{3.16}$$

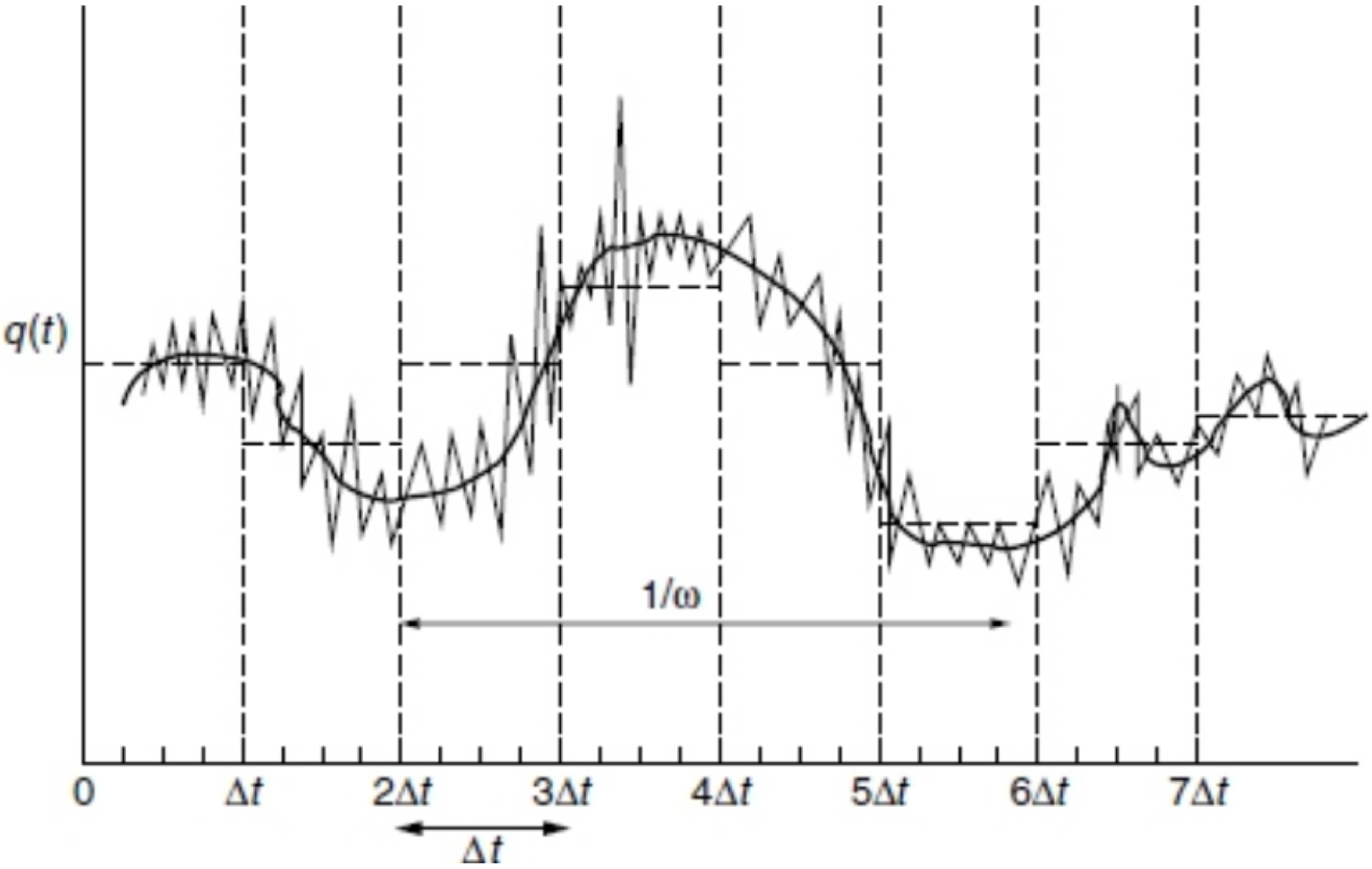
In other words, there are $\Delta t = m$ small ticks for each large one. To perform a coarse-graining, we replace the function $q(t)$ over the whole kth cell with an average value, for each non-overlapping interval Δt. We define this average by

$$\langle q(k)\rangle_{\mathrm{m}} \equiv \frac{1}{\Delta t} \sum_{i=(k-1)\Delta t+1}^{k\Delta t} q(i). \tag{3.17}$$

We have started with an abstract function $q(t)$, sampled it at discrete intervals, giving $q(i)$, and then coarse-grained the data into larger contiguous samples $\langle q(\mathrm{k})\rangle_{\mathrm{m}}$:

$$q(t) \to q(i) \to \langle q(k)\rangle_{\mathrm{m}}. \tag{3.18}$$

Figure 3.11: A jagged signal can be separated into local fluctuations plus a slowly varying local average, only if the variance is always finite.



Continuous time data

We can now perform the same procedure using continuous time. This idealization will allow us to make models using continuous functions and functional methods, such as functional integrals. Referring once again to the figure, we define a local averaging procedure by

$$\langle q(\bar{t})\rangle_{\Delta t} = \frac{1}{\Delta t} \int_{\bar{t}-\Delta t/2}^{\bar{t}+\Delta t/2} q(\bar{t}')\, d\bar{t}'. \tag{3.19}$$

The coarse-grained variable $\bar{t}$ is now the more slowly varying one. It is convenient to define the parameterization

$$\tilde{t} = (t - t') \tag{3.20a}$$

$$\bar{t} = \frac{1}{2}(t + t'), \tag{3.20b}$$

on any interval between points *t* and *t*′. The latter is the mid-point of such a cell, and the former is the offset from that origin.

3.12 Reminder

Although much of the remainder of the book explores mathematical ways of describing and understanding information from human–computer systems, assuming that observations have been made, one should not lose sight of the importance of measurement. Science demands measurement. Mathematics alone only re-describes what we feed into it. Thus, at every stage of investigation into human–computer systems, one should ask: how can I secure an empirical basis for these assertions?

Applications and Further Study 3

- *Developing critical and analytical thinking.*
- *Formulating and planning experiments to gather evidence about systems.*
- *Estimating the uncertainties inherent in observational knowledge.*
- *Diagnostic investigations into anomalous occurrences.*

[1] The applicability of the normal distribution can, in principle, be tested with a χ^2 test, but this is seldom used in physical sciences, since the number of observations is usually so small as to make it meaningless.