COMPUTING LABORATORY

On The Trustworthiness of Computing Systems

J.E. Dobson, J. McDermid, B. Randell

TECHNICAL REPORT SERIES
No 306 January, 1990
On the Trusworthiness of Computing Systems

J.E. Dobson, J. McDermid, B. Randell

Abstract

A definition of what we here call "trustworthiness" is proposed as an extension of the concept of dependability, which makes trusworthiness a judgement about the acceptability of a system rather than a property of a system. It is argued that this approach is appropriate to those large and complex systems where the rich diversity of human roles associated with the system means that the various expectations of the system cannot be reconciled into a single authoritative specification, and therefore a system can, in human terms, be judged a failure even when it behaves in conformance with some (presumably previously agreed) dependability specification. As a consequence of treating failure as subjective and trustworthiness as a judgement, we investigate new logical bases for discussing such things as the interpretation of evidence for trustworthiness, the predictability that a certain judgement will be made, the role(s) of the judge(s), the definition of system boundaries, and the criteria for acceptability.

Series Editor: M.J. Elphick

©1990 University of Newcastle upon Tyne. Printed and published by the University of Newcastle upon Tyne, Computing Laboratory, Claremont Tower, Claremont Road, Newcastle upon Tyne, NE1 7RU, England.
Bibliographical details

DOBSON, John Edward


Newcastle upon Tyne: University of Newcastle upon Tyne: Computing Laboratory, 1990.

(University of Newcastle upon Tyne, Computing Laboratory, Technical Report Series, no. 306)

Added entries
MCDERMID, John A.
RANDELL, Brian
UNIVERSITY OF NEWCASTLE UPON TYNE.
Computing Laboratory. Technical Report Series. 306

Abstract

A definition of what we here call "trustworthiness" is proposed as an extension of the concept of dependability, which makes trustworthiness a judgement about the acceptability of a system rather than a property of a system. It is argued that this approach is appropriate to those large and complex systems where the rich diversity of human roles associated with the system means that the various expectations of the system cannot be reconciled into a single authoritative specification, and therefore a system can, in human terms, be judged a failure even when it behaves in conformance with some (presumably previously agreed) dependability specification. As a consequence of treating failure as subjective and trustworthiness as a judgement, we investigate new logical bases for discussing such things as the interpretation of evidence for trustworthiness, the predictability that a certain judgement will be made, the role(s) of the judge(s), the definition of system boundaries, and the criteria for acceptability.

About the author

Mr. Dobson is a Principal Research Associate at the University of Newcastle upon Tyne, with particular interest in a system approach to security. He has been active in the field since 1982 and is the author of more than a dozen papers.

Prof. J.A. McDermid is a professor at the Department of Computer Science, University of York.

Professor Randell has been a professor of computing science at the Computing Laboratory of the University of Newcastle upon Tyne since 1969.

Suggested keywords
COMPUTING SYSTEMS
DEPENDABILITY
TERMINOLOGY
TRUSTWORTHINESS

Suggested classmarks (primary classmark underlined)
Dewey (18th): 001.642503 658.47
U.D.C. 681.322.06(038) 519.718
ON THE TRUSTWORTHINESS OF COMPUTING SYSTEMS

John Dobson*, John McDermid** and Brian Randell***

ABSTRACT

A definition of what we here call "trustworthiness" is proposed as an extension of the concept of dependability, which makes trustworthiness a judgement about the acceptability of a system rather than a property of a system. It is argued that this approach is appropriate to those large and complex systems where the rich diversity of human roles associated with the system means that the various expectations of the system cannot be reconciled into a single authoritative specification, and therefore a system can, in human terms, be judged a failure even when it behaves in conformance with some (presumably previously agreed) dependability specification. As a consequence of treating failure as subjective and trustworthiness as a judgement, we investigate new logical bases for discussing such things as the interpretation of evidence for trustworthiness, the predictability that a certain judgement will be made, the role(s) of the judges(s), the definition of system boundaries, and the criteria for acceptability.

Keywords: computing systems, dependability, terminology, trustworthiness.

Approximate Word Count: 5000

* Computing Laboratory, University of Newcastle upon Tyne, Newcastle NE1 7RU, U.K. Phone: +44 91 222 82.28 Fax: +44 91 222 82.32 email: John.Dobson@newcastle.ac.uk
** Department of Computer Science, University of York, York YO1 5DD, U.K. Phone: +44 904 43.27.26 Fax: +44 904 43.27.67 email: jam@minster.york.ac.uk
*** Computing Laboratory, University of Newcastle upon Tyne, newcastle NE1 7RU, U.K. Phone: +44 91 222 79.23 Fax: +44 91 222 82.32 email: Brian.Randell@newcastle.ac.uk
1. INTRODUCTION

1.1. Trustworthiness and Dependability

Work has recently started at the University of Newcastle and seven other collaborator institutes on an ESPRIT Basic Research Action entitled "Predictably Dependable Computing Systems". The objectives of this project are to develop unifying concepts underlying dependability which should support design decisions involving trade-offs between different approaches to dependability, to create means for the establishment and validation of dependability requirements, and to formulate stochastic techniques for assessing and predicting dependability, covering all means of attempting to prevent, remove and tolerate all types of faults. As part of the first of these objectives, we hope to show that there are reasons for wishing to revisit some of the classic areas of dependability-related terms and concepts (as explained in the paper by Laprie [10], for example).

Laprie's paper is the present culmination of an extended and very fruitful period of debate [1,2,4,7,9,12,14] about the fundamental concepts relating to reliability. This debate was initiated [11] at a time when it was becoming clear that computing systems were getting sufficiently complex that design faults were turning out to be as significant as hardware component failures. Previous definitions of reliability-related terms took as their basis the various presumed standard types of physical faults to which hardware was prone. These definitions were replaced by (recursive) definitions of terms such as "fault" and "error", whose basis was the concept of system failure with respect to a specification. Different specifications lead to different types of failure and
hence different types of what might be termed "reliability". However, Laprie successfully argued [9] that these should be collectively termed "dependability", and provided a definition which implicitly subsumes the conventional concepts of reliability, safety, security etc.

This definition of dependability, taken from [10] for example, is

**Dependability** is that property of a computing system which allows reliance to be justifiably placed on the service it delivers.

This concept of dependability does, however, depend on the notion of an 'authoritative specification', with respect to which the expected system service is defined. But even so-called "authoritative" specifications can turn out on further examination to be incomplete, ambiguous or inconsistent. This last characteristic is particularly troublesome, especially when it arises, as it often found to be the case, in a system with a very rich set of interactions with its various environments. This is because in such situations there can easily be a series of demands made on (the design of) the system which are in conflict. The 'requirements definition' phase of a project is often a matter of resolution of these demand conflicts. There is massive anecdotal evidence that such resolution is often performed insensitively, to say the least, and the fact that some users' legitimate demands have been ignored or resolved away does not mean that their expectations will be fulfilled. From their point of view, the system might be judged to have failed despite — or because of — the fact that it conforms to its (not their) specification.

In summary, then, just as new concepts and definitions were required when systems got so complex that design processes could be at fault, so we believe the time is ripe to introduce new concepts and definitions because systems are now becoming so complex that the procurement procedures and/or very specifications can also be at fault. Similar sentiments can also be found in [6],
where they are used to justify a definition of safety which encompasses the possibility of risks arising from inadequate system specification. Our own view is that such considerations can also be of relevance to all forms of dependability.

The term that we are proposing to use, at least for the purposes of this paper, for our extended notion of dependability is trustworthiness. To give a definition:

**Trustworthiness** is a judgement that a system is acceptable for use in its intended situation.

The first, and most major, difference between dependability and trustworthiness is that whereas dependability is viewed as a property of a system, trustworthiness is a judgement about a system — it is a judgement that some particular system has such a dependability property to some adequate degree.

In those situations in which it is known or assumed that the process of reaching a judgement as to a system's trustworthiness will not admit of variations, then this judgement can be regarded as a property of the system, and trustworthiness becomes equivalent to dependability. (Thus an alternative to introducing a new term, trustworthiness, would have been to argue that the concept of dependability be extended to include the notion of judgement, or to use the term "dependability" qualified by a suitable adjective. However, we have chosen for the purposes of this paper to use a different term in order to aid our discussion of the consequences of our generalisation.)

There are a number of other points of interest about the structure of our definition, each of which will have to be justified separately. These points are
i) The concept of judgement has been made paramount.

The term 'judgement' is used, according to Webster's Dictionary (2nd edition), to mean both (a) "the process of forming an opinion or evaluation" and (b) "the opinion or estimate so formed". We shall be using it in the second of these senses.

ii) The term 'acceptable' is used but not defined.

iii) The definition is deliberately vague as to the measure of trustworthiness (e.g. whether it is all-or-nothing or continuous over some range — say from zero to one).

iv) The concept 'system' is used but not defined.

v) Trustworthiness is made a judgement of a situated system, i.e. a system situated in its environment.

We will in this paper rely on the reader's intuitive understanding of the undefined terms.

1.2. Predicted Dependability

The motivation we gave earlier for the introduction of the concept of trustworthiness, with its explicit recognition of the role of judgement, was the possibility of not having an adequate system specification. Another equally important motivation, we believe, concerns the situation where one is trying to predict the dependability of some as yet unbuilt, or even incompletely designed, system. In this case, issues of judgement, and concerns about the judgement process, are very much to the fore. Thus what we might term "predicted dependability" is a particularly important case of what we have termed trustworthiness. (An appropriate definition of predicted dependability is "That property of an unrealised computing system, and the
processes to be involved in realising it, which should allow reliance to be justifiably placed on the service it will deliver."

In some cases, estimating in advance on the basis of a design what the dependability of a system is going to be, is a relatively simple matter based on standard reliability theory, which is in turn based on the mathematics of the classical calculus of chance. But in more complex cases, the underlying notion of probability changes [3] and takes on a different structure (e.g. Bayesian statistics) which is closely connected with other patterns of human reasoning, such as inductive reasoning in experimental science. This latter reasoning grades the completeness of the evidence as a way of setting standards for reasoned belief rather than assuming, as does Pascalian probability, that the evidence is complete.

In fact, even the standard notion of dependability implicitly assumes some notion of prediction. If reliance is to be justifiably placed on some system (for some purpose), then part of the justification must assume that the system will in fact behave as previous evidence suggests that it will — for example that a compiler will continue to compile code and not suddenly turn into a music-generating program, or that the operation of the computer will continue as previously and so on. In other words, the very act of establishing a justification only makes sense on certain assumptions of predictability.

There are a number of distinct aspects of the future that could be considered predictable; the following, which are not necessarily mutually exclusive, are obvious candidates:

(i) the likelihood that an (un)acceptability judgement will be made by some specific judge of some specific system in some specific situation;

(ii) the future state or behaviour of the system;
(iii) the future value of some measure defined over (some part of) the system or its environment;
(note that this differs from (ii) in bringing the environment into things, e.g. lives lost as opposed to system failures)

(iv) resources required to achieve certain values, or to constrain certain possible values, of any of the above.

We shall concentrate mainly, though not exclusively, on the first of these. In fact (i) is to us the meaning of predicted dependability, and (ii) (iii) and (iv) are simply evidence which may influence the judgement. In other words, the reason why the first seems the most important is that in practice the most important decision to be made about a system is whether or not to deploy it (including develop it, continue to develop it, decommission it, etc.). Thus the use that is made of dependability is not in deciding reliability issues concerned with some abstract property of the system (e.g. whether it is safe or secure or whatever), but in deciding trustworthiness issues concerned with the development or use of the system (e.g. whether it is fit for use in a safety-critical application). For example, the fact that a general purpose DBMS is thoroughly dependable in the sense of behaving according to its maker's specification does not necessarily mean that it is necessarily trustworthy for use in a highly secure system. Now of course the adjudged trustworthiness of a system will in practice use evidence derived from the sorts of dependability prediction measured or estimated by techniques (ii) (iii) and (iv); but these provide only support for the judgement, and are therefore in a sense secondary to it.

This does not mean that we can ignore these secondary issues. What it does mean is that we are also going to have to examine questions such as,

In what sorts of circumstances is it rational to use evidence provided by (ii) or (iii) or (iv) in making a trustworthiness judgement?
Given an agreed set of values derived from (ii) (iii) and (iv), is it necessarily the case that all competent judges would come to the same judgement?

These general questions of what sort of evidence is required for some specific sort of judgement, or to what sort of judgement is some specific sort of evidence relevant, are not questions that have to our knowledge previously been examined in the context of dependable computing systems. They are, however, questions forced on us which are immediate and important consequences of our view of what we here term trustworthiness as being a judgement about a system, rather than a property of a system.

In fact it seems that the sorts of questions that arise can usefully be categorised into three types. These three types of question will be examined in the following sections of this paper under the following headings:

Who is making the judgement?

What are the criteria for acceptability?

What is the nature of unacceptability?

2. ROLES: Who is Making the Judgement?

As indicated earlier, individuals ascribe the characteristic "trustworthy" to systems operating in some context; this context includes the organisation and the roles of those who use, or are affected by, the system. In general there will be a number of different roles associated with a system and individuals filling these various roles may make different ascriptions. Certainly they will base their judgements on different criteria. We briefly consider roles before considering the judgements and interpretations on which trustworthiness ascriptions are based.
In a previous paper [5], we distinguished functional and structural roles in the following terms. A structural role defines the responsibilities laid upon the role-holder, the relationships of the role-holder with related roles (the 'role set') and the expectations placed upon the role by the role set. The structural role therefore defines the task responsibilities of the role-holder and the associated rights and obligations; for example, we have expectations that a lawyer will treat client information as confidential, and the professional relation between lawyer and client, which is regulated by the lawyer's licensing authority, is part of the lawyer's structural role. This is different from the set of actions ('job description') that the role-holder is expected to perform. We shall term the latter the functional role. Thus by functional role we mean one which can be characterised by extensional properties such as access rights. In contrast structural roles are determined by the intensional concepts of responsibility and authority, typically deriving from a position in some organisation. Trustworthiness judgements — e.g. the system conforms to government policy, the system can be safely deployed — are typically associated with (are the responsibility of) structural roles.

Pragmatically, for any putatively trustworthy system, there will be a decision as to whether or not to deploy the system based on a prediction of its dependability. In practice there may be further decisions, e.g. to take a system out of service, or to modify it before it can continue in service. However, there will always be the initial deployment decision and it is instructive to base our analysis of judgements on this decision.

We can identify a set of typical structural roles associated with a system. Typical roles include:

- system developers — those responsible for constructing the system;
- system owner(s) — those responsible for operating the system;
• system evaluators — those responsible for judging the acceptability of the system in terms of some given criteria;
• requirements owners — those responsible for specifying the requisite functionality and properties of the system;
• system users — those responsible for utilising the outputs of, or creating the inputs to, the system.

Each of these roles has a different perception of the system and is associated with a different set of criteria for judging acceptability.

The system owner is concerned to know to what extent the system protects or supports his interests; a decision to deploy the system effectively reflects a prediction that the system will protect and support those interests to an adequate degree. Since the system will not, in general, have been operated in the user environment, this decision may have to be based on perceptions and judgements relating to the other roles. Thus for example the decision to deploy may be based on criteria (metrics) associated with the process and product of software development, and judgements made by the developers and evaluators. Thus judgements in one domain are being used to support decisions in another; these judgements must therefore bear an appropriate relationship to one another for the judgements to be sound. It may be that the concepts of measurement theory [8] have some role to play in helping to understand these relationships, since what is involved are questions of the scope of validity of such things as software quality metrics.

We can now see that the individual roles are related by responsibility for particular judgements which impinge on the deployment decision:
• that the deployment criteria ("specifications") against which the system is evaluated are of trustworthy provenance
— this is the responsibility of the requirements owner; an example is the judgement that an information-flow based model is an accurate reflection of security policy;

• that the deployment criteria form a sound basis for evaluation

— this is the responsibility of the evaluators; an example is the judgement that it is possible to interpret the available measures (perceptions) of process and product in terms of the criteria;

• that the evaluation criteria reflect the needs of protecting and supporting the interests of the system owner

— this is the responsibility of the system owner; an example is the judgement that having a system that employs an information flow-based security policy is acceptable.

3. CHARACTERISTICS: What are the Criteria for Acceptability?

3.1. Types of Trustworthiness Characteristic

In this section we shall look at the various kinds of system trustworthiness, or characteristics as we shall call them, and examine what kinds of evidence for each might count in arriving at a judgement that a system is acceptable with respect to that particular characteristic.

There are many trustworthiness characteristics, and it is probably not possible to enumerate them all. Examples are correctness, appropriateness, timeliness, safety, soundness, predictive accuracy, and so on. In order to understand them from the point of view of having to make a judgement as to whether, or to what extent, a system can be regarded as acceptable, we shall propose a classification scheme over these characteristics.
There are four dimensions to our classification scheme. Briefly, these are

- whether the characteristic is intrinsic to a system or extrinsic — i.e. whether it is (perhaps partly) determined by the situation in which a system is placed;

- whether the characteristic can be controlled by itself or whether control of it is dependent on controlling some other characteristic(s);

- the criteria that might be used in determining a judgement concerning the characteristic;

- the type of process that is involved in determining a judgement concerning the characteristic.

3.2. Intrinsic/Extrinsic Characteristics

By an intrinsic characteristic we mean one that can at least in principle be determined by examining a system in isolation from its environment. A good example is the soundness of a theorem prover, or reliability with respect to a fault hypothesis that is internal to a system, for example the formal correctness of an algorithm for atomic RPC in the presence of a single node failure.

By contrast, an extrinsic characteristic is one that depends for its elaboration on circumstances external to the system, e.g. safety or security, where the very definition of what these terms are taken to mean depends on the policies of the problem owner.

The distinction that is being made here is whether or not it makes sense to talk of the system as exhibiting or supporting a particular characteristic. Intrinsic characteristics are those that are exhibited by a system; they are expressed in terms internal to the system and are measured by intrinsic
means. Extrinsic characteristics are supported by the system; they are expressed in terms which involve concepts external to the system and are measured in general by value judgements.

We note in passing that this distinction between intrinsic and extrinsic allows us to explain the distinction that is commonly made between the terms 'safety' and 'security' in what is to us at least an intuitively appealing way: assuming for the moment the unanalysed notion of a resource, we can say that security is concerned with the protection of intrinsic resources (e.g. data files, cpu cycles) whereas safety is concerned with the protection of extrinsic resources (e.g. human life and welfare).

3.3. Independently Controllable Characteristics

The second dimension of our classification is whether the characteristic can be independently controlled or whether it is dependent on other characteristics.

Reliability can be controlled independently of other characteristics, using standard techniques of fault avoidance, fault removal, and fault tolerance. So can such characteristics such as soundness and predictive accuracy (by suitable choice of logic and algorithm). Safety, however, is dependent on reliability, and security on both reliability and soundness (of the logical model of security).

It might seem that this dimension is closely related to the previous one; and indeed it is. But they are in fact orthogonal, as can be seen from the following example:
<table>
<thead>
<tr>
<th>Independent</th>
<th>Dependent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intrinsic</td>
<td>Soundness</td>
</tr>
<tr>
<td>Extrinsic</td>
<td>Timeliness</td>
</tr>
</tbody>
</table>

The justification for this table is that the soundness (in the proof sense) and timeliness of a system can be controlled independently of other characteristics using appropriate structures and mechanisms. Prevention of denial of service to a given user, however, can only be achieved provided one has a way of controlling the other users — in other words, the users themselves must enter into some agreements outside the system being considered if denial of service is to be controlled [16]. Under these conditions, however, denial of service is a system characteristic that can indeed be defined in terms solely involving system behaviour. (The arguments concerning the placement in of the table of ‘security’ have already been made).

### 3.4. Criteria for Judgements

There are two forms which a criterion for a judgement can take: it can be defined in terms either of a specification of the system (e.g. its behaviour) or of an expectation (e.g. that it must pass some particular test imposed by a licensing authority). In the first case, the specification is a direct guide to the designer and proof by refinement of specification is (at least in principle) possible; whereas in the second it provides no further constraints on the designer (other than to pass the test of course). This is an example of the distinction between analytic and synthetic properties which we noted in our previous paper [5].
3.5. Types of Epistemological Process

An epistemological process is a process by which one comes to know things. In the present context, this means the process by which one comes to a judgement about the acceptability of a system. The reason for describing these processes is that for each trustworthiness characteristic, only some process types are relevant to making a judgement concerning that characteristic. Thus for example, predictive accuracy can be measured by counting the number of accurate predictions, whereas security is hardly to be measured by counting the number of security violations.

There seem to us to be five relevant types of epistemological process. These types are:
induction, deduction, measurement, elicitation and abduction.

(i) Induction

It is possible to believe that a system is acceptable on the basis of experience of other, similar, systems. This is, of course, the basis of probability measures and other types of statistics. Compared with the other types of process being discussed in this subsection, induction seems to be a useful guide to deciding on the trustworthiness of a system with respect to some particular characteristic for only a limited number of characteristics, such as certain types of reliability.

(ii) Deduction

Deduction is the process of following a logical chain of argument from correct premises. It is the basis of provably correct systems. Not all characteristics are
suitable for deductive proofs, though for some, such as security — or more correctly, conformance to a stated security policy — it is (in principle, if not always in practice) both feasible and desirable.

(iii) Measurement

The process of determining some characteristic by measuring it is one that is perhaps not as widely understood in computing as in other engineering disciplines. There are good reasons for this, of course, some of which stem from the fact that what is being measured is not some physical property of the world, but an artefact of our own making within it. What is not always understood in computing is the relation between what can be measured (e.g. number of lines of code) and what we want to know about (e.g. the reliability of the code) — but this is the province of measurement theory.¹

(iv) Elicitation

Elicitation is the process of asking people about things. As a basis of determining some property of some known system, it is of course perfectly acceptable; as a basis of predicting some property of some unknown system, it is probably less reliable, but also quite possibly more reliable than might at first be thought if the system is of a kind that is understood by the people being asked the questions.

¹For an excellent introduction, see Kaposi [8].
(v) Abduction

Abduction, according to Webster (2nd edition), is the process of arguing from premises, and therefore arriving at a conclusion, whose validity is a matter of probability (e.g. replacing "x is a y" in a syllogism by "x might be a y"). Its application to the making of a judgement is obvious, even if the manner of its application is not. Mathematical bases, of various degrees of plausibility and rigour, have been established for abduction.

There is some work to be done, and which we are in fact planning to do, in deciding which of these various forms of process are appropriate for arriving at a judgement concerning the various kinds of trustworthiness characteristic. We hope that the results of this work, with supporting arguments, will be the subject of a future paper.

4. BREAKDOWN : What is the Nature of Unacceptability?

In the literature on dependability, there is much talk of failures (and associated words such as fault and error). We hope to examine these concepts also in more detail in the light of our concept of trustworthiness in a subsequent paper, but we are here concerned to point out that what is usually meant by 'failure' is in fact but one instance of a more general concept which we shall term 'breakdown'\(^2\).

Whereas a "failure" is the failure of a system with respect to a specification, and can therefore in some sense be formally analysed, a breakdown is a human

\(^2\)This term is taken from Heidegger, who has given us perhaps the most penetrating analysis of the function of breakdown in helping us to understand things. A very readable introduction to his thought, and its application to the design of computer systems, is to be found in the book by Winograd and Flores [15].
state of affairs in which things are not as they are desired to be in a way that may defy analysis. In a sense, breakdowns function in a positive rather than a negative way. New designs can be created and implemented in the space that emerges as a result of previous breakdown. A design is thus to some extent an interpretation of (previous) breakdown and a committed attempt to anticipate future breakdown, and modelling of a design can be seen as a way of asking questions of the model concerning possible breakdowns both of the design and of the human design process.

There seems to be not one, but three distinct forms of breakdown that are relevant to our concept of trustworthiness as judgement. These are

(1) The system might in fact not be acceptable for reasons internal to the system (e.g. it fails to behave according to its specification or some other acceptability test);

(2) The judgement about a system might be incorrect (e.g. its predicted or actual behaviour is wrongly judged to be acceptable);

(3) The entity being judged might not be a system (e.g. it might be a component or an incomplete system).

We shall call these breakdowns in mechanism, judgement, and systematicity, respectively. Breakdown in mechanism is certainly the best understood of these. It has been, and still is, the subject of an enormous amount of work in the computer systems community. Breakdown in judgement has been widely recognised in such social domains as the legal system, the medical system, and so on, all of which have developed procedures to tolerate it, but its application in the computer domain has not yet received the attention it deserves. Breakdown in systematicity is most frequently manifest when too narrow a view is taken of a computer system, forgetting the organisational context in which it is placed. See, for example, the book by Rasmussen et al [13], which contains several examples of error arising from interaction
between subsystems (some of them human), where no single subsystem could unarguably be said to have failed.

5. CONCLUSIONS

In this final section we shall state two morals drawn from our examination of concepts and vocabulary. The first moral is that failure to state explicitly a fault hypothesis in dependability analysis is about as egregious an error as failure to state explicitly a null hypothesis in statistical analysis. The second moral is that means of evaluating judgements are as important as means of making judgements.

5.1. Stating the Fault Hypothesis

Given the standard definition of dependability, perhaps the most important question to be asked is "On what am I relying?" Answering this question requires understanding the various forms of breakdown that might occur, recognising that hypothesised and actual faults might be different. Techniques such as fault tree analysis and penetration ('tiger team') attacks force explicit statement of fault hypotheses, and are therefore methodologically preferable to random or even heroic testing (unless the testing is much more soundly based than is customary) if the purpose is to provide evidence of the trustworthiness of the system to an evaluator.

5.2. Evaluating Judgements

The main way in which judgements can be evaluated is to examine firstly the role of the person making the judgement: Is this kind of person appropriate for this kind of judgement? and secondly the epistemological process involved in
arriving at the judgement: *Is this kind of process appropriate for this kind of dependability characteristic?* Here, the questions of appropriateness are themselves matters of judgement, and are therefore subject to breakdown in just the same way as any other judgement. Generalised fault-tolerant structures and procedures (but not, of course, fault-avoidance or fault-removal procedures) therefore apply to these decisions also.

5.3. Summary

If we really take judgement seriously, if with Frege (at least the Frege of the *Begriffsschrift*) we believe that judgeability lies at rock bottom in logic (i.e. that logical analysis *begins* by taking the judgeable-content of something as its data), then dependability properties, relations and concepts have no existence *independent* of judgement; and so the terminology used for such properties, relations and concepts must be explained in a way that reflects their origin in the judgement process. A statement such as "This program is correct (with respect to its specification)" is not a theorem to be proved but a judgement to be made in the light of certain evidence. When an incident occurs which casts doubt on the correctness of the judgement, it is therefore crucial to be able to track back the evidence on which the faulty judgement was based. It is only this feedback which can be used to improve the quality of future judgements, and hence make dependable systems more trustworthy.

Acknowledgements

We thank Jean-Claude Laprie for many discussions on this and related topics. The ESPRIT Basic Research Action (PDCS) is a collaborative project involving City University (London), IEI del CNR (Pisa), Universitaet Karlsruhe, LAAS-CNRS (Toulouse), University of Newcastle upon Tyne, Universite Paris-Sud,
Technische Universitaet Wien and University of York. An earlier draft of this paper was presented at a PDCS workshop.

References


