On the Meaning of Safety and Security

A. BURNS1, J. McDERMID1 AND J. DOBSON2*
1 University of York, 2 Computing Laboratory, University of Newcastle upon Tyne, Newcastle upon Tyne NE1 7RU

We consider the distinction between the terms 'safety' and 'security' in terms of the differences in causal structure and in terms of the differences in the degree of harm caused. The discussion is illustrated by an analysis of a number of cases of system failure where the safety and security issues seem, at least at first sight, to be difficult to disentangle.

Received July 1991

I. INTRODUCTION

The terms 'safety' and 'security' are often used to characterise computer systems when used in a context where reliance is placed upon them. These terms are normally regarded as representing distinct properties, yet there is considerable ambiguity in the terms. At a linguistic level, the common phrase 'safe and secure' indicates a limited distinction and, in German, no real distinction can be made as the term sicherheit means both safety and security (although some technical terminology has been introduced to make distinctions between the two notions). Further, in many dictionaries, safety is defined in terms of security — and vice versa.

There are also some difficulties in practice in deciding when to use which term. For example, the following problems might be treated as problems of safety or security (or both):

- unauthorised modification to the contents of an ROM in a car Automatic Braking System (ABS) leading to a fatal accident;
- a software design fault in a programmed trading system which causes a bank to buy many times its assets, hence bankrupting the company;
- a syringe pump whose setting was altered by an unauthorised (and untrained) individual to give a fatal dose of drugs.

All of these examples have characteristics with connotations of security (e.g. unauthorised access or modification) and characteristics with connotations of safety (e.g. causes of fatalities) and it is unclear whether these systems are concerned with safety, security or both. Perhaps the second example has the least obvious safety characteristics — but it certainly was not safe, in the sense of causing irremediable damage, for the bank to use the system.

A full analysis of the above examples would require us to consider more carefully system boundaries, and so on. However, we believe that there are some underlying principles which help us to carry out such an analysis, and here we are primarily concerned with those principles. Specifically we believe that safety and security can be distinguished in terms of the nature of the harm caused and the nature of the causal relationships between events leading to a safety or security 'incident'. Our aim here is to throw light on these ideas.

1.1 The value of the distinction

It is reasonable to enquire whether it is worthwhile making such distinctions, especially as there is now a growing trend to use the term 'dependability' and to treat safety and security simply as special cases of dependability. However, we believe there are a number of benefits, as we will endeavour to show.

In the abstract, it is much more satisfying to have clear distinctions than relying purely on intuition, so the analysis is of intellectual benefit. For example, some say that safety is a special case of security, and our analysis enables us to suggest a resolution of this issue. More pragmatically, safety-critical and (supposedly) secure systems are subject to different legislation and standards regarding their use, assessment and construction. It certainly is useful to know which legislation applies — assuming that the relevant bodies will accept the definitions we propose.

Secondly, there is, we believe, some importance in the concepts we use to make the distinction, namely the nature of the causal relationships involved and the degree of harm caused. Certainly our own understanding of the use of these concepts has been improved by our attempts to apply them to understanding the concepts of safety and security.

Finally, at an engineering level it is important to be able to determine what are safety- and security-relevant components to aid in designing a system so as to minimise the number of critical components. Similarly, different techniques are appropriate for analysing requirements and designs where safety is our concern, rather than security (typically the distinction is between failure mode behaviour and failure mode consequences, e.g. unauthorised information flow). Also we believe that, given clearer understanding of the terms, we can produce better analysis techniques for dealing with safety and security — indeed, this is the prime motivation behind our work.

1.2 Existing definitions

We must stress at the start that we are not trying so much to define the terms 'safety' and 'security' as to show the nature of the distinctions between them. The main problem with the definitional approach is that, as church history has all too often shown, it can give rise to religious wars. We do not wish to start another one. Rather we wish to argue that there are two separate discriminants that can be applied in separating security issues from safety issues, and we wish to suggest a way in
which these discriminants can be combined. Unfortunately, providing definitions seems to be the clearest way of showing the distinctions. We have tried, however, to be consistent in giving the definitions in pairs in order to emphasise, not the definitions themselves, but the differences between them.

Although we are concerned with drawing distinctions between safety and security as concepts we recognise that many systems have both safety and security facets, or aspects (see for example the system discussed in Ref. 1), and so for simplicity we shall talk about safety critical and secure systems in much of the following.

In the introduction we implicitly used intuitive, and normal dictionary, meanings of the terms safety and security. The problems and ambiguities we have identified arise, at least in part, from ‘overlaying’ technical meanings onto everyday words. Unsurprisingly therefore, in addition to the common definitions of the terms, a number of ‘technical’ definitions have been produced which are intended to clarify both the meanings of the terms and the distinctions between them. Our belief is that these definitions are still rather unsatisfactory. For example it is common to define security in terms of intentional (deliberate) faults, and safety in terms of damage to resources including people and the environment. However, these definitions do not provide clear distinctions since we can have intentional safety-related failures (e.g. through vandalism to protection systems by ‘pressure groups’), and so these definitions do not really help in analysing the above examples.

Similarly it is common to say that security is concerned with logical resources, e.g. money, and safety is concerned with physical (including, but not limited to, human) resources – but how then do we classify the failure of an automatic teller machine (ATM) which emits large numbers of banknotes (physical entities) when it should not? Clearly the logical/physical distinction is limited in applicability as there is a difficulty in deciding what level of abstraction is appropriate for making the judgement, and physical entities can be carriers of information. So far as we are aware, other existing definitions suffer from similar problems – by which we mean that the definitions make distinctions between safety and security which contradict our intuitions, or do not enable us to make distinctions in all relevant circumstances. It is these problems that make us believe that it is appropriate to seek new definitions and analysis techniques.

1.3 Content of the paper

It seems rather unfruitful to debate at length existing definitions of safety and security (see, for example Ref. 2 for safety and Ref. 3 for security). One problem with such definitions (and there are many of them) is that these definitions of safety and security have been developed separately, and hence do not exhibit any interesting parallels or similarities of concept. The fact that in some languages the same word is used for both suggests that there are some conceptual similarities and it is there that we wish to explore; hence we shall present and refine our understanding of safety and security in parallel.

The rest of the paper is structured as follows. We first present, in Section 2, an informal discussion of the concepts of safety and security by looking at two possible alternative approaches. In Section 3 we discuss rather more rigorous definitions for safety and security, refine our definitions and give a commentary on the definitions. In Section 4 we analyse a few simple examples so as to try to provide verisimilitude to our definitions. Finally, Section 5 draws some conclusions about the utility of the new definitions.

2. AN INFORMAL CHARACTERISATION OF SAFETY AND SECURITY

In presenting our understanding of safety and security we shall first give some intuitive, but not altogether satisfactory, definitions and then refine them to a state where we feel that they are ‘satisfactory’ informal definitions. By ‘satisfactory’ we mean that they fulfil the criterion of respecting the common intuitions regarding the terms and that they will suffice as a precursor to the more rigorous definitions in Section 3. As we are concerned with showing the distinction between the terms we develop definitions for each term in parallel. We suggested above that causal definitions and the consequences of failure are distinguishing characteristics of safety and security, so we treat these two notions in turn.

2.1 A causal basis for defining safety and security

Our first intuitions regarding the distinction between safety and security are most easily represented in causal terms. Specifically they relate to the extent to which a system failure affects us immediately, or its effects are delayed (or only potential). Here we use immediacy in the sense of causal consequence, not temporal immediacy, but we note that causal immediacy often entails temporal immediacy. To avoid complexity in definition we proceed on the basis that either normal behaviour or failures can lead to harm – this obviates problematic discussions of what specification should be used as a basis for judging failure, where we use the word ‘failure’ to mean any behaviour that has an undesired effect (even if such behaviour was in accordance with, or did not contradict, a specification). We can thus define the terms as follows:

- a safety critical system is one whose failure could do us immediate, direct harm;
- a security critical system is one whose failure could enable, or increase the ability of, others to harm us.

From these two definitions we immediately derive two more definitions which we shall need:

- a safety critical system is one whose failure could do immediate, direct harm;
- a security critical system is one whose failure could enable, or increase the ability of, others to harm is.

These definitions seem to reflect basic intuitions – a system is not safe if it can harm us; it is not secure if it gives others the means of harming us. Note that since real systems are never perfectly reliable (their failure rates are never nil), in the terms just introduced, a safety critical system is not safe. In practice we would say that a system was safe either if it was not safety critical

* Here and elsewhere, we are not naïvely assuming that a system has only one failure mode. Read ‘failure’ as being short for ‘any of whose conceivable possible failure modes’. By failure, incidentally, we mean ‘any undesired behaviour’.
(intrinsic safety), or it was safety critical but the likelihood that it would fail and harm us was acceptably low (engineered safety). Similar interpretations apply to the definition of security.

Our main interest is in articulating clearly the distinction between the concepts of safety and security, and not in formulating all of the related definitions, so we restrict ourselves to refining the notions of safety critical systems and security critical systems as this simplifies the exposition of our ideas.

Our next refinement is to note that the implicit distinction is not just one of immediacy of cause, but also of sufficiency — with safety we are concerned with sufficiency of causality whereas with security we are concerned with sufficiency for enabling causality. Thus an alternative way of stating our definitions is:

- a system is safety critical if failure could be sufficient to cause us harm;
- a system is security critical if failure could not be sufficient to cause us harm, but could increase the number of possibilities, or likelihood of existing possibilities, for others to intentionally cause us harm.

There are a few corollaries which we can usefully draw before we go into more detail.

First, we can see that the definitions help us with regard to the classification of systems — this can be done through classification of the consequences of failure. Second, a system can have both security and safety implications through the causal consequences of different potential failures, or even through different causal consequences of a single failure. Third, the definition of security encompasses the notion of other agents and this has some overlap with the notion of intentional faults, although it places them in the context of the causal consequences of the failure, not the causal antecedents. This does not, of course, mean that the failures cannot be deliberately caused, rather that the distinguishing characteristic of security is what is done after, rather than before, the failure (a malicious agent could still exploit information which he gains through happenstance).

Fourth, we must be concerned with specific failures, not all failures, as safety and security critical systems are not usually critical in all respects. We will take into account the above observations when refining our definitions later in the paper.

The main limitations of the above definitions relate to the nature of failure, the nature of the causal relations and the scope of the causal consequences. We have, as yet, been vague about what is meant by failure and, more importantly, our implicit definition is not obviously compatible with the standard definition set out in Ref. 4 (as that definition is based on the notion of a specification against which failure is judged,* and ours does not explicitly rest on such a concept). Further our notion of causality has not been explored and there is considerable subtlety in modelling causality for computerised safety and security critical systems. Finally, limiting the definitions to systems that harm us, whether interpreted individually or corporately, is over-restrictive as we may still deem a system safety critical if it can harm others. We resolve this latter problem below, and address each of the remaining issues in Section 3.

* Even though it accepts that specifications can be at fault.

2.2 Failure consequences as a basis for defining safety and security

Our definitions introduced the notion that safety and security critical systems are those whose failure can lead, directly or indirectly, to harm. However, this does not take into account the fact that there are different degrees of severity of failure. Even safety critical and secure systems can have relatively benign failures. Sometimes the term safety-related is used to refer to systems which have safety 'responsibilities' and where the consequences of failure are slight (e.g. small-scale injury rather than death) although the distinction between safety-related and safety-critical systems is inevitably difficult to draw precisely. We wish to draw out a rather more fundamental distinction, which is between what we term relative and absolute harm.

We shall use the terms 'absolute' and 'relative' to denote scales of value in the following way.

An absolute value is when the value ascribed to that which is to be protected is not quantified but considered to be either all or nothing. i.e. after system operation (including the possibility of system failure), the value is either what it was before (normal or intended operation) or else valueless (failure or unintended operation). Human life is like this. So are legal judgements — they are either correct or incorrect, hence the term "unsafe" in English law to describe a verdict which is felt by competent judges to be incorrect (in a certain legally defined sense of running counter to the evidence).

A relative value is when the value can quantitatively degrade as a result of system operation, either continuously or on a stepwise basis. Money is like this; so is the value ascribed to information. Indeed most security critical systems are concerned with protecting money or information, but we believe that this is a consequence of the nature of the concept, not a defining property.

Absolute harm usually occurs when some service or resource is impaired. Clearly loss of life is such an impairment, but a major spill from an oil tanker would also be seen as an impairment to a variety of resources as there is irrecoverable loss of (wild)life, loss of natural habitat, and so on. Relative harm usually occurs in adversarial situations (e.g. when the objectives of two enterprises are in conflict) and relates to a gain or loss in competitive advantage. Here loss of the designs for a new product by one company and its acquisition by another causes relative harm — absolute harm would only entail if the competitor exploited this knowledge so as to impair the first company seen in its own right (e.g. by making it go out of business). Further, loss of the information would not be harmful if there were no competitors (and if it were impractical for any to be created to exploit the information gained).

There is potentially a complication to the notion of 'degree of harm' as the judgement about degree will depend on context and point of view. For example, a syringe pump may be more critical when used on a ward for administering drugs than when used for administering anaesthetics in an operating theatre (because the patient's vital signs will be monitored continuously in the latter case but not the former). More significantly, what is deemed to be absolute harm from one point of view may be classified as relative from another point of view. To take a commercial example, a computer failure which
caused loss of a single customer’s account probably would be considered an issue of relative (and probably minor) harm by a bank, but an issue of absolute harm by the customer (unless there was some way for the customer to regain the funds). This is a classical case where we must apply techniques of hazard and risk analysis, but we need to take into account the relevant point of view in assessing the results of the analysis. Note that ‘point of view’ is orthogonal to ‘degree of harm’.

Having introduced the notions of absolute and relative harm, we now propose definitions (which for the time being we will treat as alternative to the previous ones) of the terms safety-critical and security-critical systems based on these notions:

- a safety-critical system is one whose failure could cause absolute harm;
- a security-critical system is one whose failure could only cause relative harm.

Given the above discussion on degrees of harm it should be clear that a fundamental point is being made, and we are not simply picking out extremes of a spectrum. The point is that the distinction between safety- and security-critical is being explained in terms of the nature of the harm caused, as seen from a particular point of view. It is perfectly reasonable for a system to be viewed as safety critical from one point of view and security critical from another, since the degree of harm may be absolute from one point of view and relative from another.

It might be argued that the definitions are difficult to apply in a number of contexts, e.g. military systems which are intended to do harm. However, we can always delimit the scope of the services and resources which are considering depending on the objectives of the organisation using the system. Thus, clearly, protection of one’s own troops, equipment and so on is the concern in military situations and failures which harmed them would rightly be considered safety issues, although harm to the opposition would not—in fact ‘harm to the enemy’ is likely to be taken to be equivalent to ‘of benefit to us’, and this is why the point of view represented by ‘us’ is an important indexical. Interestingly we can see both safety and security issues in the weapon system example. Imagine an artillery system that used ranging shells before firing the main weapon. Firing the ranging shells gives rise to relative harm—information is given to the enemy which may enable them to do absolute harm to the artillery piece or the troops. However, successful firing of the weapon may increase safety. Note that we have here a trade-off between security and safety and that security can be increased by the normal behaviour (this is true of many ‘risky’ pursuits), whereas safety is such that normal behaviour should be safe. The relationship between safety and security is discussed more fully in Section 4.

As a further example, we briefly return to the banking system situation outlined above. Loss of some of a bank’s funds is an example of relative harm (to the bank treated as a resource) but not absolute harm as the bank, its funds, its profits, and so on can all still be talked about meaningfully (this is a sufficient condition for harm not to be absolute). However, a competitor may be able to ‘make capital’ out of the incident and lure customers away from the troubled bank. From the point of view of a depositor whose funds were lost in their entirety (and where there is no recovery mechanism) the issue is one of safety—the depositor may lose his house, be unable to buy food, and so on. It certainly was not safe for him to entrust his funds to the bank. Thus we can see that this concept enables us to take into account issues of context and point of view quite naturally. Also this simple example illustrates why we believe safety to be an appropriate attribute for the second example given in the introduction (the programmed trading system).

We have now proposed two apparently different views of the notions of safety and security (one causal, and the other based on the notion of classification of degrees of harm). It is now appropriate to consider the relationship between these two views of safety and security and to see whether or not they can be reconciled.

### 2.3 Relationship of the two views

The above two views of, or ways of defining, safety and security appear very different and there seems to be no obvious way of telling whether or not they are compatible. There is no a priori reason to suppose that a distinction in terms of the kind of damage caused (absolute or relative harm) is the same as the distinction in terms of the causal structure (direct or indirect harm). However, the fact that the two distinctions appear to be extensionally equivalent (i.e. to produce the same classification in practice even though on different logical bases) does perhaps explain some of the puzzlement and difficulty we experience when in trying to separate the concepts.

At this stage there are two strategies upon to us. We can either observe that there are two separate distinctions and introduce two separate pairs of words to classify situations according to which distinction is being invoked; or we can note the extensional equivalence and provide a definition which combines the distinctions. We choose, not entirely confidently, the second of these approaches. One reason for our choice is the observation that in an adversarial context, relative harm usually occurs when competitive advantage is given—and this is equivalent to increasing the number (or likelihood) of ways to give opportunities to others of causing harm. Not all situations fall into an adversarial context, of course, but it is difficult to use the word ‘security’ plausibly without having any connotations of ‘the enemy’ (who wishes to harm us). We also note that the relative/absolute distinction enables us to remove the restriction of harm as something that applies solely to us. If in addition we now explicitly introduce the notion of context and judgement, we have:

- a system is judged to be safety-critical in a given context if its failure could be sufficient to cause absolute harm;

* As usual there is an issue of system boundary here. We are implicitly saying that there is no recovery mechanism within the system being considered (bank plus depositor). If there was a further system which provided means of recovery from the problem—say the government—then we would have been guilty of a failure in systematicity in analysis (we should have included the government), and hence in using the term ‘absolute harm’.

† The problem of treating failure as a judgement is subtle and difficult, and is often ignored by computer scientists (perhaps because of its subtlety and difficulty). We are examining the issues involved and hope to report on our investigations in due course.
a system is judged to be security-critical in a given context if its failure could be sufficient to cause relative harm, but never sufficient to cause absolute harm.

Again we must remember that a system can have both safety and security connotations, and we need to be more careful to distinguish different failures, and different causal consequences of failures, when formalising the ideas. We also need to take into account the fact that a system can support many services, so we shall make a final refinement of the definitions at a later stage. The definitions appear to make security 'subordinate' to safety in that security failure can clearly lead to (be a partial cause of) a safety incident in a wider system. The issue is slightly more subtle and we return to the notion when we have presented our formalisation of the concepts.

3. A NOTATION FOR REPRESENTING THE CONCEPTS

Our aim here is to put our definitions on a firmer footing from the point of view of a model of causality – which of course means that we can capture both aspects of the definitions.

We do not attempt to model the (quantitative) level of harm, as opposed to the (qualitative) degree of harm, as we believe this is inevitably a subjective issue, so that the best we could do would be to present a treatise on utility theory and leave the application of the theory up to the reader. However, as we will show, our causal definition captures one fundamental aspect of the notion of relative/absolute harm, by distinguishing between what we term agent and event (action) causality. We express the distinction in terms of intentionality: agent causality is where the causal relation is intended, and even causality is where there is no deliberate intention for the causal relation to hold. The validity of this way of describing the distinction depends on scope, i.e. the agents we consider, but it is acceptable if we restrict ourselves to the operational environment for the system.

We adopt here a simple formalisation for representing causal relations. We use simple predicates to represent conditions under which an event will occur, and define dependencies between events which represent a stage in a 'causal chain'. It would be possible to develop and to use a causal logic (of which there are several in existence and under development). However, since we are concerned here with clarifying concepts, a simpler approach will suffice. First, however, we make a few observations on causality to clarify the nature and purpose of some of the causal structures, although we strive to avoid the philosophical quagmire of the nature of causality itself.

3.1 Basic causal notions

When we say, informally, the some event causes another we usually only give a partial description of the events, and of the conditions under which the causal relations hold. There are two basic issues here: one relating to even descriptions and the other concerning necessity and sufficiency of causal relations. These two points are very strongly related but we discuss them in turn rather than together at the risk of slight repetition.

Consider lighting a match. We would usually describe the event that caused the match to light as 'striking a match', but there are other aspects of the 'event' of striking the match which we could have described, but did not, e.g. the speed of movement, the nature of the surface, and so on. Thus we can have different descriptions of the same event. In our definitions we will assume that we can adequately characterise distinct events, but we recognise that distinguishing and identifying events is difficult in practice.

Taking the second point, we can also use the match example as a basis of our discussion. Normally we would say that striking a match was sufficient cause for it to light – but actually this is only a partial cause and the presence of oxygen, the fact that the match and the box were not wet, and so on, all are contributory causes. Further, striking the match is not necessary – we could light it using a blowtorch. Similar considerations apply in safety and security – the failure of a computer system can only lead to harm if that system is connected to, and controlling, dangerous equipment, or the like. Thus such a failure is neither necessary nor sufficient. Determining a causal relationship seems to be a matter of judgement, but one for which we can give some structure. We will be concerned with describing events and causal relations in terms of necessary and sufficient conditions and showing the relationship between such conditions.

We now introduce some terminology and structure to our causal definitions which we will use extensively later.

The failure of a computer process control system which leads to a harmful incident in the controlled process will normally only be a partial cause of the incident. Typically we would refer to the failure as the cause as this is the aspect of the situation that changed and thus apparently caused the incident. The remaining causal factors which usually are taken for granted are sometimes called, for that reason, standing conditions.

It is useful to investigate the relationship between the failure (partial cause) and the standing condition in more detail. Typically computer system failures will be an insufficient but necessary part of some larger causal condition which is, itself, unnecessary but sufficient to cause harm (known as an INUS condition). Put another way, there are many conditions which are sufficient to cause the overall system to fail causing harm, but none of them are necessary, i.e. there are a number of possible failure modes. More technically there are several minimal sufficient conditions (MSCs) which represent the separate failure modes for the controlling and controlled system. These MSCs incorporate the notion of 'standing condition' which we introduced earlier. For each (or perhaps only some) of these MSC the failure of the computer system is necessary, but not sufficient – the connection of the computer equipment to the plant, the absence of suitable protection mechanisms, etc., are also needed before the event of concern can occur. Thus a failure of a safety critical (computer) system will, in general, either be an INUS condition for the controlled system or not contribute to a harmful situation.

It is useful to introduce some simple notation to deal with such causal structures. In general we will have a set of MSCs and at least one of these conditions must be true for the incident to arise. Further, for each MSC, some conditions must be present for a harmful failure to
occur and some must be absent. We can therefore represent the possible causes of a harmful incident as:

\[ \text{MSC}_1 \lor \text{MSC}_2 \lor \text{MSC}_3 \ldots \]

where we can regard each MSC as having three components:

\[ \text{MSC}_N = \langle C_N | F_N | D_N \rangle \]

(In this notation, the bra (\(\langle\)) and ket (\(\rangle\)) symbols and the vertical bars are to be taken as part of the syntax of the first and third components respectively.)

In this formula, component \(\langle C \rangle\) (for contributory) represents those conditions which are part of the normal operation of the system for the failure \(F\) to make the MSC true and \(\langle D \rangle\) (for defensive) represents those things which are part of the abnormal operation, these being either ordinary components in an error state is detected.*

For example \(\langle C \rangle\) might be the presence of flammable material, \(F\) a failure which generates a spark, and \(\langle D \rangle\) the absence, or or inoperative state, of a suitable sprinkler system. It is indeed often the case that a partial condition for a failure to occur has to be expressed negatively, as the absence or inoperative state of a particular component. However, we shall assume that the implied negative is included in the statement of the \(\langle D \rangle\) condition in order to avoid a possibly confusing use of the negation symbol.

In the above terminology the failure, \(F\), is the INUS condition. For the INUS condition to be a ‘sufficient cause’, or to have immediate effect in the causal sense, it must do for our definitions to be appropriate, then both the contributory and defensive causes (\(\langle C \rangle\) and \(\langle D \rangle\) respectively) must hold for the appropriate MSC.†

The above notation represents causal immediacy, but it says nothing about temporal issues. If we view \(F\) as the condition caused by the failure event then we can say that the harmful event actually occurs if the MSC is never true – thus this deals with the situation where there are delayed hazards which nonetheless are caused (in the INUS sense) by a failure.‡ In principle with a delayed cause the failure might never lead to a hazard.

There are some modelling assumptions behind this discussion of causality. The most significant is that we assume, in effect, that \(\langle C \rangle\) represents those aspects of the ‘real world’ which are beyond our control, or at least outside the sphere of control of the system or enterprise we are considering. In contrast \(\langle D \rangle\) represents issues which we may be able to control, or influence, such as engineered protection systems, and the like. Thus \(\langle D \rangle\) need not be static as we may modify our defences as time passes (clearly the ‘real world’ changes too but many causal relations, e.g. to do with the flammability of gases, are invariant). However, it is typically the case that \(\langle D \rangle\) changes slowly by comparison with the operation of the system.

Also we note that changes to \(\langle D \rangle\) may be planned, or unplanned. It is clear that, in many safety critical situations, operators find ways of managing the equipment which were not intended by the plant designers. Whilst, in some cases, this may contribute to the incident becoming an accident often they will find ways of preventing an accident by going outside the operating procedures for the equipment. Our causal structures can cope with this creativity, but it gives us some difficulty in defining what the attribute ‘safety critical’ means in causal terms. However, since \(\langle D \rangle\) would normally change only slowly, we can say that \(F\) is a critical failure if it would lead to harm \textit{ceteris paribus} – all other things being equal. Thus a system would not cease to be safety critical just because an operator had found a way of dealing with one particular incident if he had done so by modifying a \(\langle D \rangle\) condition (or equally, but less likely) a \(\langle C \rangle\) condition, in a way that had not previously been considered. Perhaps another, more constructive, way of summarising this would be to say that we make the judgements of criticality in terms of the causal dependencies that at design time are believed will later hold. To put it more strongly, the distinction between \(\langle C \rangle\) and \(\langle D \rangle\) is judgemental, but the judgements are of practical benefit as a way of structuring an analysis of critical systems during their design and analysis. These observations have ramifications for the way in which we build specific models of safety- or security-critical systems.

We have just introduced another interesting dimension to this discussion – the epochs at which the conditions and events are determined (and evaluated). All the events and conditions are, we assume, defined (identified) at design time. However, events in the class \(F\) arise at execution time for the computer system. Thus although the \(\langle C \rangle\) and \(\langle D \rangle\) conditions can be described at design time, the evaluation takes place at the time of event \(F\). We need therefore to be able to describe the changes that take place to \(\langle C \rangle\) and \(\langle D \rangle\). The (values of) conditions \(\langle C \rangle\) are established by physics, the plant design and its operation, and \(\langle D \rangle\) is evaluated in a similar way to \(\langle C \rangle\), except that unplanned events are more likely to modify \(\langle D \rangle\) than \(\langle C \rangle\) during plant operation. Also maintenance can change any of the event/condition definitions and change actual values. Thus maintenance activity can be analysed using the same type of causal structures, provided we can show how failures in maintaining \(F\) modify \(\langle C \rangle\) and \(\langle D \rangle\) conditions associated with other MSCs.

### 3.2 Types of entity in the model

It is sometimes convenient to think of object in and surrounding a computer system as falling into three classes: agents, activities and resources. For our purposes here we need to extend the model with the notions of:

- **objectives** – a specific target of the some enterprise; in this context we will be concerned with safety and security objectives for the enterprise using the system;
- **services** – a grouping of activities which are all intended to satisfy some objective; in this context we will be concerned with those services intended to achieve safety or security objectives;
- **events** – an action associated with a service or a resource; we are concerned with such events as safety...
and security relevant occurrences (including failures) and their causal consequences.

Further we need to classify the enterprises and systems involved in order to have a suitable basis for articulating the model. We distinguish the computer system (CS) from its operational environment (OE) which contains all the equipment which has direct interactions with the computer system. In our discussion we will assume that the computer system only provides one service so we will refer to computer failures, not service failures. The computer system and the operational environment are classes of resource, as is the critical resource (CR) the ‘protection’ of which is the concern of some objective (it may be a component of the computer system or environment). We use the term user enterprise (UE) for the organisation concerned with operating the computer system to achieve some specified objective or objectives. We use the term operators (O) for those agents who legitimate interact with the computer system or the equipment in its operational environment; operators are members of the user enterprise. We use the term free agents (FA) for other agents who may be involved in the causal chain resulting in harm to the critical resource. We will use the one or two letter mnemonics as subscripts to label events, and the like, in order to be clear about the interpretation of the causal definitions we will present.

Some comments are in order before we present our definitions. We note that agents are essentially human, and they are characterised by holding responsibility and authority. Agents will typically be identified by a combination of role and objective – an agent will hold some role in respect to the achievement of a defined objective. Authority may be delegated to machines, but responsibility may not – this is essentially a human (organisational) notion. Thus, even where automation is used for critical functions, e.g. automated stock trading, the machine has the authority to buy and sell, but the responsibility for the actions of the programs lie elsewhere (with some agent). This notion will be central to some of our later analyses.

We could think of systems having multiple services and satisfying several objectives (with a many-to-many relationship between services and objectives). For simplicity we assume a single objective for the computer system and a single service intended to satisfy that objective, again without loss of generality. We could add a further agent, the judge, to cater for the fact that notions of safety and security (and the identification of events, for that matter) are judgemental. Again we make simplifying assumptions and the definition presented can be thought of as representing the judgement made by the designers with respect to events, causal chains, and the like. Thus we believe our definitions are simpler than is necessary for a full analysis of safety or security critical systems, but that the essence of the approach is fully covered by the definition, or definitions.

We present two distinct definitions and we treat safety before security.

3.3 A definition of safety

We can now give a definition of safety which, at the top level, is surprisingly simple:

\[ H_{cr} \leftarrow \neg C_{OE} F_{cs} \]

A failure of the computer system \( F_{cs} \) combined with the standing conditions in the environment \( C_{OE} \) (i.e. the plant design and physics) is sufficient to cause the harmful state for the critical resource \( H_{cr} \) (where we use the arrow \( \leftarrow \) to denote ‘causes’). Implicitly there is no defence in the operational environment from the failures (no \( D_{OE} \)) so the failure leads immediately (in the causal, not necessarily the temporal, sense) to the harmful state. If such a relation holds then the CS is safety critical. Where there is more than one possible failure mode we can extend the right-hand side of the above formula by disjunction:

\[ H_{cr} \leftarrow \neg C_{OE} F_{1cs} \lor \neg C_{OE} F_{2cs} \lor \ldots \]

This definition, of itself, adds little to the earlier discussion although the causal immediacy is clear (as mentioned above, there is no defence in the operational environment). However, we can extend the idea to model the causes of states such as \( F_{cs} \) in the illustration above and to consider defence conditions. We might have:

\[ F_{cs} \leftarrow \neg C_{CS} F_{o} \lor \neg C_{CS} A_{F} D_{OE} \]

Here we have a step in the causal chain for representing a failure on the part of the operator \( F_{o} \) (an erroneous action) which the computer system does not prevent or render nugatory, or a legal, but inappropriate, action by the operator \( A_{F} \) which is not defended against due to (say) a computer design failure \( D_{cs} \), leading to the computer system failure event \( F_{cs} \).

At an abstract level the causal definition is quite clear – it is obviously strongly related to the notion of fault-trees. Clearly, in practice, the difficulty would come in defining the states, events and causal relations. We will sketch ways of carrying out this modelling in Section 4.

3.4 A definition of security

The definition of security is similar to that for safety, except that there is an ‘indirection’ in the causal chain:

\[ H_{cr} \leftarrow \neg C_{OE} A_{FA} D_{OE} \]

where the defence conditions depend on the free agents:

\[ D_{OE} \leftarrow \neg C_{CS} F_{cs} \lor \neg C_{OE} F_{o} \]

Here \( D_{OE} \) represents the ‘defence condition’ which can be negated by the failure of the computerised (security-critical) system, and \( A_{FA} \) represents an attack which can succeed when \( D_{OE} \) is false (perhaps a password is leaked). Thus a failure of the security critical computer system produces a vulnerability \( D_{OE} \) which can be exploited by an attack. This two-level causal definition seems to capture directly the notion that a vulnerability is caused by a failure and that further acts (by the free agents) are required to cause harm. If such a relation holds then the CS is security-critical. Again, both right-hand sides can be extended by disjunction as before.

Note, of course, that the cause of the computer system failure may be in the development process or operator malfeasance. This sort of situation is illustrated in the above by the second disjunct representing a negation of a defence condition by the malfeasance \( F_{o} \) of an operator.

* Of course the resource to be protected may be in the computer system and, in this case, the condition should refer to the CS not the OE, but the principle remains the same.
Thus the above causal structures are general enough to permit the modelling of Trojan Horses and other forms of deliberately introduced flaws so long as the model is extended to cover all the relevant enterprises, e.g. the developers.

This definition shows clearly the distinction between safety critical and security critical systems and hopefully clarifies the notion of causal sufficiency. There are, however, some interesting characteristics of our definitions which we have not yet explored and there are some improvements we can make to the definitions, for example, to reflect the fact that a computer system may support multiple services. We provide a brief commentary first, then present the final form of the definitions.

4. COMMENTARY

We discuss here a number of important characteristics and ramifications of the definitions. There is no notion of priority implied by the ordering.

4.1 Simplicity of the formal definitions

Our formal definitions are very compact and may appear trite or almost vacuous. Indeed the reader might be inclined to wonder what the fuss is about. However their simplicity is a greater virtue. The distinction between safety and security and their relationship has been a troublesome matter for some time. We hope and believe that the very simplicity of the distinction speaks for its validity. In any case the simple and sharp distinction gives us a powerful analytical tool as we shall show later.

4.2 The scope of the harmful behaviour

The definitions consider harm without being specific about what set of resources, facilities, and the like should be considered (the military examples make it clear that we have to make some distinctions). We need to consider in respect of all resources, and so on, whether there is a legitimate reliance on protection or preservation of that resource by the computer system. Some examples should clarify the point.

Safety-critical and security-critical systems can harm 'individuals' other than the system owners or operators – for example the clients of a bank, or the environment of a chemical plant – and this often the primary reason for being concerned about such systems. As before it is easiest to see the issues in the context of safety. Consider a chemical process plant. It is clear that the owners and operators of the plant are responsible for the resources directly controlled by the plant, e.g. the chemicals themselves and the pipes, valves, and so on which make up the plant. However, the owners and operators also have responsibilities for their workforce and the environment through legal and moral frameworks. In the safety case we often speak of a 'duty of care' and within the context of an enterprise model we would expect to see explicit responsibilities representing this duty.

It may seem simply that we have to take into account the set of authorities and responsibilities of the owners and operators, but we also need to take into account the consequences of failure from different points of view. Thus, for example, there will be responsibilities and authorities associated with a computerised banking system from the points of view of both the owners and operators, and from the point of view of the users. In setting the system to work the bank is (effectively) delegating authority (for certain aspects of security) to the computerised system for itself and for its clients. Thus in this case the computer system is acting as a surrogate for the bank. Perhaps more precisely when an individual becomes a client of the bank he (or he and the bank jointly) are delegating authority to the computerised system. However, in delegating authority the bank is taking on a responsibility for the safe and secure operation of the system from the point of view of the customers. Thus a definition of safety and security needs to take into account responsibilities from wherever they can (legitimately) arise. Recall our earlier comments on agency. Responsibility cannot be delegated to a machine and the responsibility for system safety or security remains with the operators, e.g. the bank, or perhaps the developers depending on the nature of the development contract.

In analysing particular situations we need to introduce an enterprise model covering all those organisations and individuals who have legitimate dependence on the enterprise operating the system (and perhaps those the operators depend upon) in order to evaluate responsibilities and to delineate the resources which should be considered in a safety or security analysis.

4.3 The relationship between safety and security

We noted above that security problems (relative harm) can often lead to safety problems (absolute harm). We have already mentioned an obvious military example relating to information about a proposed offensive reaching the opposing force in time for improved defences to be prepared. In the commercial sphere we can imagine similar information regarding future products.

However, safety incidents can also lead to, or imply, security incidents when there is an adversarial situation. For example destruction of a physical protection mechanism, e.g. a fence, may well be viewed as a safety incident but it might also have security connotations (it increases the likelihood of certain sorts of 'attacks' succeeding). Perhaps more interestingly the death of a worker in a factory may adversely affect customer confidence and thus give other companies a competitive advantage (we note that, in practice, security is often occasion concerned with covering up safety problems).

In an adversarial situation, or where there are competitors, a safety problem almost invariably implies a security problem. Security problems are, by their very nature, in the causal chain of those classes of safety incident where the absolute harm arises from the action of some agent who gained useful knowledge from the failure of some other system. In this sense security is subordinate to safety, in a single context, but, in general, security incidents can lead to safety incidents, and vice versa, where there are causal dependencies between actions in different contexts. In this sense it is not meaningful to talk about one concept being subordinate to, or representing a subset of, the other.
4.4 Recovery and judgement

Our definitions are based on the notion of causal consequences. However, as we mentioned above, if some form of recovery action is taken then the anticipated causal consequences may not come to pass. This is the reason we introduced the notion of \textit{ceteris paribus} in our earlier discussions (although the definitions do not explicitly introduce this notion). This raises the question of when the judgement about causal consequences is made.

The obvious answer to the above question is that the judgement is made at design time – when decisions are being made about how to protect against predicted failure modes. This is true in the sense that such judgements are made at this time, but this is not the only time such judgements are made. They may also be made in accident investigations (when it may be realised that the causal chains are different to those which the designer anticipated), in maintenance, and even during operation (the operator taking some recovery action).

Thus the definitions have to be interpreted as being judgements made on a \textit{ceteris paribus} basis at a specific time, recognising that different judgements may be made at different times by that individual (and perhaps by different judges). Consequently recovery is viewed as that which the judge (perhaps the system designer) would view as abnormal, i.e. falling outside normal procedures, at the time of making the judgement. We do not clutter the definitions with such qualifiers, but they should be interpreted in this light.

4.5 Safety, security and integrity

Given our domain of discourse there is inevitably a question of the level of abstraction at which our definitions apply – after all systems are composed of other systems, and so on. We motivated the definitions by consideration of computerised systems, but clearly the definitions could be applied to other classes of system (this is one of the appeals of the causal definition). The simple answer is that they can be \textit{tried} at any level of abstraction which is of interest, but that they cannot be \textit{applied validly} at all levels. The test we need to apply is whether or not the causal relations set out above apply to the system when modelled consistently at that level of abstraction. With this interpretation we see that it is \textit{not} appropriate to define all subsystems in a security critical or safety critical systems as being themselves security critical or safety critical. Of course this is nothing new – but it is a criterion for judging the validity of our definitions. Also there are some interesting corollaries of this observation.

First, little software inside a safety critical system can legitimately be called safety critical. In many cases it will only be device drivers controlling actuators which have this property. This is not to say that other software in the system does not contribute to safety, but that its role in the causal chain means that it does not deserve the attribute ‘safety critical’ according to our definitions. When considering ‘real world’ issues we assumed that \(\text{C} \) and \(\text{D} \) in our causal definitions were largely static (although \(\text{D} \) is more likely to change than \(\text{C} \)). This is not the case \textit{within} a safety critical system. A failure of a software component could modify the \(\text{C} \) condition for a safety critical software component, e.g. corrupt a table used by a device driver, and thus lead to a failure. We would term the corrupting component \textit{integrity critical}. We will set out a definition for this concept when we give our final definitions of safety and security.

Second, when considering security critical systems we observe that, in a typical situation, rather more of the software can be considered to be security critical than in a similar safety critical system. This is in part due to the indirect causal links between failure and harm but also due to the fact that the resources of concern are typically ‘inside’ the computer system. However, in a similar manner, we may introduce the notion of integrity critical software components (indeed the definition is identical).

Third, the application of the attributes security critical or safety critical can be erroneous. Sometimes, this is just a simple error of judgement, but there are occasions of what we term failures of systematicity, as the error arises from considering an inappropriate system or system boundary. Note that this applies to complete systems, not just system components, as for example a computerised information system which has no capability to control physical plant but which advises operators on what to do. The computer system plus operators to safety critical. The computer system only is safety critical if the operators have no way of independently validating the computer’s advice (no \(\text{D} \) condition).

We accept that, in common practice, the terms security critical and safety critical are used for components of larger systems which are legitimately viewed as security critical or safety critical even though the components fail our definitions. Whilst many of these components will best be thought of as having integrity requirements we believe that the terms \textit{safety related} and \textit{security related} are quite helpful as descriptive terms for those components which are potentially in a causal chain which can lead to a catastrophic failure – although at least one of these terms already is used with other connotations.

5. FINAL DEFINITIONS

The definitions we gave in Section 2 were rather limited in a number of respects. We have improved the definitions, but there several issues that we need to address including the fact that safety and security are more properly thought of as properties of services (of a computer system in a context) rather than of computer systems, \textit{per se}. Following Laprie\textsuperscript{4} we exploit the definitional convenience of treating services first, then considering how to classify systems based on the properties of the services which they provide.

Thus we first provide definitions for safety- and security-critical services:

- a service is judged to be safety-critical in a given context if its behaviour could be sufficient to cause absolute harm to resources for which the enterprise operating the service has responsibility;
- a service is judged to be security-critical in a given context if its behaviour could be sufficient to cause relative harm, but never sufficient to cause absolute harm, to resources for which the enterprise operating the service has responsibility.

We have used behaviour here rather than failure because, as we identified above, normal behaviour as...
well as failure behaviour may have safety or security connotations. Now we can deal with systems:

- a computer system is judged to be safety-critical in a given context if it has at least one component or service which is judged to be safety-critical;
- a computer system is judged to be security-critical in a given context if it has at least one component or service which is judged to be security-critical.

This simple reworking enables us to make the appropriate distinctions between systems, but also to recognise that a given system may be both safety- and security-critical (and the same applies to services). We can now complete our definitions by considering integrity:

- a service is judged to be integrity-critical in a given context if it is in the causal chain of service provision for a safety-critical or security-critical service but whose behaviour is not sufficient to cause relative or absolute harm with respect to resources for which the enterprise operating the service has responsibility;
- a computer system (component) is judged to be integrity-critical in a given context if it has at least one service which is judged to be integrity-critical.

These definitions summarise the results of much of the above discussion.

### 6. EXAMPLES

In order to illustrate the analytical utility of our definitions we briefly reconsider the three example situations outlined in the introduction. In order to present a full analysis of these situations we would need to build extensive models of the enterprises and systems involved. However for our present purposes such detail is not necessary and we simply add adequate information to our earlier descriptions to enable us to present an informal analysis of each situation.

#### 6.1 Automatic braking system

Our first example was based on a hypothetical incident involving an automatic braking system (ABS), namely:

- unauthorised modification to the contents of an ROM in a car ABS leading to a fatal accident.

In order to analyse the possibilities we need to consider system boundaries. For our discussion we will assume that the fatality involved the driver.

Taking the system as the car and its occupant then the situation is quite simple:

- the service is the provision of optimal braking;
- the incident is an example of absolute harm;
- the failure of the service (computerised ABS) was the immediate cause of the harmful incident (this is clear in our example although the engineering detail is not supplied);
- the corruption of the ROM is an integrity flaw which led to an (unspecified) safety failure of the computer system and service.

The ABS service and the control computer are safety-critical and the ROM is integrity-critical.

We can now consider how the fault arose. Here we assume that the modification to the ABS system arose when a 'specialist' tried to optimise the braking performance and failed. Here the system is the car plus the technician providing the service;

- the service is optimisation of braking performance;
- the harmful incident was the corruption of the ROM.

In this case we would say that there was an integrity failure of the service and this in turn was perhaps caused in turn by inadequate education of the technician. Note that if the technician had been in the pay of a rival manufacturer and had deliberately introduced the fault, then this would have been a security issue. Thus our definitions allow us to discriminate between different contexts. The importance of this is that, seen from the point of view of providing appropriate countermeasures, the difference is important - countermeasures against well-meaning but incompetent mechanic include training, whereas countermeasures against wicked competitors infiltrating Trojan mechanics involve much harder issues of personnel selection.

#### 6.2 Programmed trading system

The second example was of a programmed trading system which malfunctioned with disastrous results:

- a software design fault in a programmed trading system which causes a bank to buy many times its assets, hence bankrupting the company.

Again we make the analysis in stages. Here the system being considered is the bank including its computer system. Again the situation is quite simple to analyse, although we need to add some detail in order to clarify the example:

- the service is buying and selling shares, exploiting the difference between the prices in two different markets, and the ability of computers to react faster than humans;
- the incident is an example of absolute harm;
- the failure of the service (programmed trading) was the immediate cause of the harmful incident (this is clear in our example although details of the financial collapse were not supplied);
- the software design fault is an integrity flaw which led to an (unspecified) safety failure of the computer system and service.

Perhaps the only slightly surprising aspect of the analysis is the treatment of the bank failure as absolute harm, but hopefully the reason for this assessment should be clear from the definitions and earlier discussion. Clearly the programmed trading service and computer must be considered safety critical (although perhaps a bank would not realise this until after the system had failed).

If we consider the development environment for the software (where we presume that the software design fault arose) then, if the fault were introduced accidently, we would have a case of an integrity failure, as above. However, if we considered rival banks and the possibility of a maliciously introduced flaw then clearly we have a

* We are aware of organisations which carry out such 'optimisation' on engine management systems; a similar service for brakes seems somewhat less likely, but not beyond the bounds of possibility.
security problem – in the software development environment. Thus one fault could be caused in two different ways – but this should not be surprising and it indicates that we may need to provide protection against more than one minimal sufficient condition for causing a harmful condition to arise.

6.3 Syringe pump

The third example relates to a commonly used medical instrument, a syringe pump, which can be set up to pump drugs or anaesthetics from a syringe into a patient at a predetermined rate. In our example we only have a hypothetical fault, but we assume that the incident suggested actually occurred to help us lay out the analysis. The example is:

- a syringe pump whose setting could be altered by an unauthorised (and untrained) individual to give a fatal dose of drugs.

At first sight, it might appear that this is equivalent to the case of the ABS that we have already discussed. However, there are some interesting differences which we shall draw out when we consider the system boundaries, since there is no equivalent in the ABS case of the hospital organisation as a whole being considered as the system in which the fault or failure occurred.

To start with, we consider the system as being merely the syringe pump and the patient. The basic judgements are:

- the service is the provision of a particular drug at a predetermined rate;
- the incident is an example of absolute harm;
- the failure of the service (drug provision) was the immediate cause of the harmful event (presumably too high a rate of delivery);
- the corruption of the delivery rate setting is an integrity flaw which led to the 'over-rate' safety failure of the computer system and service.

Clearly the service and the computer control system in the pump are safety critical.

Analysis of the source of the failure is more interesting and we consider two possibilities. First we assume a situation where there is free access to the hospital and there are no mechanisms for challenging individuals and preventing them from entering wards. The system is the syringe pump. We assume that the individual who mis-set the rate had a grudge against the patient (to provide the opposing objectives). Here we have:

- the service is the provision of a facility for accurately setting a predetermined drug delivery rate.
- the incident is an example of an integrity failure;
- the failure of the service (setting delivery rate) was the immediate cause of the harmful state (too high a rate of delivery).

The service and the computer system providing the service are integrity-critical. It might be expected that this would be viewed as a security problem. However, within the context discussed there is no prospect of competitive advantage and no notion of relative harm. This is still rather a surprising result and we return to the point below.

Secondly, we consider an alternative situation where the hospital authorities are responsible for access to the premises. Here we have to consider two levels of system – the pump as before as the ‘inner’ system and the hospital and its staff, including the syringe pump, as the encompassing system. The analysis for the inner system is the same as before and we do not repeat it here. For the encompassing system we have:

- the service is the prevention of access by individuals who have no legitimate business on the premises (more specifically ensuring that individuals with legitimate access only visit those areas compatible with their legitimate interests);
- the incident is an example of relative harm (the death is not entailed by the failure);
- the failure of the service (admitting an unauthorised individual) was only a partial cause of the harmful incident (setting too high a rate of delivery and hence causing the fatality).

This is clearly a security not a safety incident (although death ensued as the causal consequence of the security failure). Interestingly this is a good example of a situation where an organisation (the hospital) has responsibility for ‘resources’ which it does not own, i.e. which do not constitute part of it as an enterprise.

A few further comments are in order. The two scenarios which we analysed above are really rather different. In the first case we (implicitly) assumed that the hospital had no responsibility in regard to the setting of the delivery rate on the pump and clearly this is not true in the second case. It is interesting to assume instead that the hospital does have the responsibility (specifically for the appropriate setting to maximise the chance of recovery), but we now need to design the mechanisms of the hospital and to consider the options.

First we assume that the ‘hospital design’ is intended to ensure that only qualified staff, e.g. nurses, can have access to modify the pump settings and that everyone else is to be prevented from doing by physical means; this is then simply the second case we analysed above.

Specifically the hospital has the authority and responsibility with regard to the integrity of the pump setting, but it delegates the authority (but probably not responsibility) to the nursing staff for actual changes, and to the security staff (and to a lesser extent medical staff) for prevention of unauthorised (and unqualified) modification.

However, if we assume open access the situation is quite different. Wittingly or not, the hospital has effectively given away any notion of authority (specifically, access capability) for the pump settings to anyone who enters the hospital. This is actually a security fault in the design of the hospital – relative harm is done in regard of the future patients of the hospital, or at least those who have ‘enemies’. Clearly the security fault could be remedied by adding access control mechanisms to the pump – and this would perhaps have satisfied our intuitive expectations that the incorrect setting of the pump would be viewed as a security failure. We believe that the above analysis is more realistic, but it does point out the need to consider the operational context when considering this type of requirement.

Indeed, it is often the case that introducing a new computerised system into an organisation should lead to a modification of that organisation in terms of respon-
sibilities and authorities. Thus, in many cases, we need also to consider the design of both the computer system and the organisational system at the same time. It seems to us that one of the major sources of failure in developing many critical systems is the failure to realise the intrinsic relationships between the human organisation and the computer system. This is an issue which is being investigated by one of us (JD) under an ESPRIT project.

6.4 Formalisation of the examples
In principle, formalisation of the examples requires a causal logic in order to represent the dependencies between the different events and states (conditions), although we could have produced a semi-formal definition of the examples using the notation introduced earlier. However, this would have yielded little more insight into the examples than the above analysis and considerable effort would have been needed to build suitable models of the systems of interest in order to express the predicates defining the conditions. For this reason we chose to present informal arguments.

Nonetheless we are firmly convinced of the value of the formal frameworks as they enable us to provide more rigorous, and potentially automatable, analysis techniques. We are investigating the formalisation of such structures both by extending existing formalisms and developing causal logics.

7. CONCLUSIONS
We believe that our treatment of safety and security yields good definitions both in the sense that they are close to the common intuitions and they can be applied systematically. In essence the value of our definitions is that they make clear 'what we are looking for' when we are trying to find out if something is safety or security critical. Obviously this is crucial to requirements and design analysis techniques. It is hoped that the discussion and the examples above substantiate this point.

The discussion also throws further light on the definitions of safety and security presented in Ref. 4. It is clear that the definitions do not need to make safety and security disjoint, so that our initial concerns about those definitions is no longer valid. However, we believe that our treatment of the issues shows that intentionality is properly associated with the consequences of a security failure, not simply with the causes of the failure (although, of course, they can be caused deliberately). Perhaps more generally we have shown that security is characterised by the occasioning of relative harm (in an adversarial situation), and this is certainly not apparent from the definitions in Ref. 4. However, the spirit of the definitions is not substantially different, and we can perhaps best view our treatment here as clarifying the concepts underlying the earlier definitions.

Readers may have been struck by the similarities between our causal definitions and fault-trees. The MSCs internally have the structure of and-gates in fault-trees, the disjunction of MSCs have the structure of or-gates, and the defence conditions play a similar role to inhibit-gates. This observation is perhaps unsurprising in safety, but is perhaps less obvious for security. Of course the difference in the security case is that our causal definitions (fault-trees) have to range over the actions of free agents as well the system of interest and its legitimate operators. Nonetheless it indicates that we might profitably use techniques such as fault-trees in the analysis of security.

We also hope that the examples make clear the importance of identifying system boundaries and fault assumptions. Given different boundaries and assumptions quite different conclusions may be drawn about the criticality or otherwise of particular systems or components. We believe that much of the weakness in many practical safety and security analyses comes from paying inadequate attention to these two issues.

Finally we believe that our definitions help in making trade-offs. There is little firm evidence of this point in the paper, but the example in Section 4.3 shows how different assumptions about the structure of organisations and where mechanisms are placed can affect the requirements for equipment. This is a topic which would bear further study.

We are carrying out further research on the specification and analysis of safety and security properties, particularly in the context of requirements analysis, and believe that the approach should lead to more effective techniques. In particular we wish to deal with situations where authority is delegated to automated systems (such as in programmed trading systems) and/or where consideration of the constraints of the organisation using the system are paramount. Some of our work on formal approaches to specification is reported in Ref. 8, and earlier works, e.g. Ref. 7, give a basis for the definitions needed to articulate the specifications. In addition we are considering ways of supporting generalised safety arguments which would include recording in an inspectable way 'arguments' such as those presented in Section 4 (see for example Ref. 9). We believe that the insight provided by the above definitions and analysis, together with these specification techniques and models, will give us an effective basis for reasoning about safety and security and for making reasoned trade-offs in the development process. We hope to be able to publish examples in the near future to substantiate this belief.

Acknowledgements
This work has been done as part of ESPRIT Basic Research Action 3092, Predictably Dependable Computing Systems (PDCS), and we are grateful to our many colleagues in PDCS who have commented on earlier versions of this paper. Particular mention must be made of Lorenzo Strigini, who has the rare gift of writing comments that are simultaneously perceptive, critical and helpful.

Finally, our thinking in this area was stimulated by a short contract from the Admiralty Research Establishment (ARE), and a slightly different version of this paper formed part of our report to ARE. We are grateful to ARE for their permission to rewrite and republish.
REFERENCES


Book Review

K. V. RUSSELL (Ed.)
Yearbook of Law Computers and Technology, Volume 5
Butterworths, Sevenoaks, 1991

This international yearbook's general aim is to publish work arising from a variety of disciplines and perspectives which reflect on the law's concern with new technologies. Has it succeeded? Each issue now concentrates on a major theme. Here it is 'Technology and the Courts'. Clearly this ties up the disciplines. In the two principal sections, the theme section 'Technology and the Courts' and 'Current Developments', certainly the perspectives of the judiciary, academics and I.T. specialists are well represented. The experience of practising solicitors who, after all, are in direct touch with the courts' consumers, would also have been relevant.

The introduction reminds us that effective use of technology enhances the quality of justice, making the system more accessible and more accurate, and providing more benefits. Yet this is not the rationale employed in marketing the new technologies. Efficiency and economy, particularly the latter, are the keywords, so that the vested interests in the politics of achieving justice can safely hold their own.

The theme section provides interesting descriptions of what is happening with the use of computers in courts, both in the UK and abroad, and glimpses of the exciting developments to look forward to, heralding the transformation of our legal institutions in what Ronald Madden epitomizes as the 'post-Gutenberg' era. The content is rooted in the practicalities of what has already been achieved.

Mark Tantam's approach is thought-provoking, not only because of the analogy between international conflict and adversarial conduct at court, but on the ways of presenting cases effectively to jurors who are not used to concentrating for long.

In a fascinating account of promoting the LEXIS service, Kyle Bosworth points out that, contrary to conventional wisdom, lawyers do not in fact typically spend their time looking things up. This has meant the consideration of new ways of encouraging the use of computer-assisted legal information retrieval.

In contrast to the first section, there is a lack of balance in the 'Current Developments' section. Two articles about data protection, two on education, one straightforward analysis of the EC Telecommunications Services Directive, and one article which could have been included in the theme section, comprise a miscellany without a coherent focus on selected topical trends and their implications. In this section, Roy Freed's controversial article, on teaching and practising computer law effectively, stresses the nature and scope of the subject in an idiosyncratic way. His extended definition of 'computer law', going far beyond his initial quotation of the 'substantive legal aspects of the availability of computer technology', is so wide as not to be entirely meaningful. The authoritative reviews of significant books published and the discussions of major cases during the year are useful.

It is unfortunate that a book with such informed content, which is well produced, on good-quality paper, backed by a distinguished Advisory Board, should suffer from so many misspellings, particularly such a book, where knowledge of electronic remedial techniques should have been taken for granted.

This book is refreshing in bringing innovative insights and ideas to the fore, as well as in the actual reports of what is taking place in the courts.

RACHEL BURNETT
London