Safecomp99, 27-29 Sept 1999, Toulouse, France

The Integration of Safety and Security Requirements

David Peter Eames[1] and Jonathan Moffett[2]

[1]ASACS Safety and Standards Unit, RAF, UK
deames@assu.org.uk
[2]Department of Computer Science, University of York, UK
jdm@cs.york.ac.uk

Abstract. This paper investigates safety and security requirements specification methods, and proposed techniques for the integration of contrasting methodologies. The nature of interaction between safety and security requirements, and problems relating to their independent development, are discussed. The requirements specifications of an Air Traffic Control system are used to highlight the problems inherent in the independent approach to requirements development. From investigation of the literature and the case study, we identify several areas that can cause problems when we attempt to harmonize safety and security requirements techniques. The most important of these are: different system models used for safety and security; different documentation structures for the analyses and their results; the interaction of safety and security requirements; isolation of safety and security requirements processes.

1 Background

Computer systems are increasingly used in areas where their failure could have serious consequences. There are many opinions as to the properties such critical systems should possess, and the techniques that should be used to develop them. Two such properties are safety and security. Within their domains, specialised methods have been developed to investigate and generate requirements specifications.

However, systems that are now being built are frequently required to satisfy these properties simultaneously. Due in part to the evolutionary growth of the approaches to safety and security specification techniques, they have largely been developed in isolation. As a result, there is growing interest in the degree to which techniques from one domain complement or conflict with those from the other.

Although there has been some work in the area of integrating techniques, this has concentrated on the techniques themselves, identifying similarities and differences, or presenting ways in which they can be brought together. The aim of this work is to investigate the nature of integration, and its influence on requirements specification.

Section 2 discusses methodologies for risk assessment in the safety and security domains. Section 3 presents a case study of an air traffic control system that is to be modernised in the near future, and has both safety and security requirements. Section

4 discusses the case study in the light of the integration framework presented, and section 5 draws conclusions from the analysis and provides pointers for the future.

2 Survey of Safety and Security Risk Analysis Techniques

2.1 Introduction

In this section we present a discussion of accepted methodologies for risk assessment within the safety (2.2) and security (2.3) domains. We then investigate work carried out to integrate security and safety risk analysis (2.4). Section 2.5 identifies the common structure of safety and security analysis. The final part of this section (2.6) is a summary and discussion of these analyses.

2.2 The Safety Risk Analysis Process

The safety risk analysis process has the aim of specifying the safety requirements of the system. There are four distinct stages in this process:

1. *Functional and Technical Analysis.* The first stage is to gather data on the system; its functional and technical characteristics. The aim is to develop a picture of how the system works. Characteristics to investigate include: the system functions or missions, the system structure, how the system is operated, the environment within which the system functions, and the system boundaries.

2. *Qualitative Analysis.* The purpose of qualitative analysis is to investigate the failure causes and hazards that could affect the system. Many hazard analysis techniques are available to produce a thorough and consistent investigation. Choosing the most appropriate technique must take into account the overall goals of the analysis, the system being analysed, and the assets available to support the analysis. This analysis develops an understanding of the failure mechanisms of the system and the combinations of failures that could lead to hazardous situations.

3. *Quantitative Analysis.* Quantitative analysis involves putting numbers to the findings of stage 2. The data used to quantify the identified hazards will always have a degree of uncertainty; it will be probabilistic in nature, taken from sources such as test results, operational records, etc. Quantitative analysis can, however, provide developers with a measure of the relative threat of the hazards, thus allowing attention to be focussed in critical areas, and may be useful in comparing the reliability of alternative features of designs [1].

4. *Synthesis and Conclusions*. Combining qualitative and quantitative analysis identifies critical components and important functions. It allows developers to identify the measures and requirements of the system which must be put in place if it is to

be acceptably safe. It is this work that generates the system requirements specification.

2.3 The Security Risk Analysis Process

The methodology of security risk analysis also comprises a number of basic steps. These differ between authors, but in general include:

1. Asset Identification. The asset identification phase should identify the resources that require protection. These will include: hardware, software, data, documentation, and computer services and processes. Financial values can be readily applied to some of these assets, but others are more difficult to price.

2. Vulnerability Analysis. Having listed the assets of a computer system, the next stage is to determine their vulnerabilities. This stage is more difficult than the first, as it requires a degree of imagination to predict what damage might occur to the assets and from what sources [2]. The general aims of computer security are to ensure data secrecy, data integrity and availability. System vulnerabilities are situations that could cause the loss of any of these qualities. A thorough understanding of the threats to the system is required if all the vulnerabilities are to be identified. Methodical and structured approaches are required if threat identification and vulnerability analysis is to be successful.

3. Likelihood Analysis. The aim of likelihood analysis is to ascertain how often the system will be exposed to each of the vulnerabilities identified. Likelihood relates to the current security safeguards and the environment in which they are applied. Estimating the probability of exposure to a threat can be difficult. Sources of data for this estimation include: operations logs, local crime statistics and user complaints.

4. Countermeasure Evaluation. All the analysis so far reflects the current situation. If, from this analysis, it is determined that the projected loss will be unacceptable, new or alternative countermeasures will have to be investigated. New controls will have to be identified, and their effectiveness evaluated.

2.4 Integration of Safety and Security

The term "integrate" can mean many things. A thesaurus will provide a number of synonyms, such as: consolidate, combine, synthesise, unify, and harmonise. In the main, the meaning of these is epitomised by the word *unify* (the dictionary meaning of which is ‘make or become one’). This meaning has also been adopted by a number of authors who have investigated methods for the integration of safety and security requirements specification techniques. The result of this approach is a single set of requirements describing the safety and security functions of a system.

An alternative meaning is encapsulated by the word *harmonise* (the dictionary meaning being ‘cause to agree, reconcile’). Rather than combining techniques to produce a single, unified methodology, here we are considering adjusting or modi-

fying techniques to bring them into alignment with each other. The aim of this would be to produce individual sets of requirements for safety and security that can be compared and analysed for conflicts without having to utilise intermediate techniques.

The Unification Approach. The idea of unifying safety, security and other criteria together under the heading of 'dependability' has been investigated by a number of authors. The notion of dependability has been around for a long time. One definition is "that property of a computing system that allows reliance to be justifiably placed on the service it delivers" [3]. Within this definition, dependability thus includes attributes such as reliability, availability, *safety* and *security*. Work by McDermid [4], and Sanders and Meyer [5] adopt this approach.

Harmonising Approaches. A number of general papers have been written discussing the differences and/or similarities between safety and security properties. Cullyer presents a general paper [6] on the nature of safety and security, in which he recognises the differences between the two, but also states that both groups subscribe to similar development techniques. Rushby [7] also presents an extremely pertinent paper in this context, in which he draws a conclusion (among many) that safety and security techniques could be applicable to each other's domains. Brewer [8] also presents an interesting slant on this theme, by considering the application of security techniques to the development of safety systems and looking at the relationship between safety and security. He concludes that security could benefit from fault tolerant approaches typically found in safety techniques, and that security system developers might benefit from a greater understanding of the hazard analysis methods used by safety engineers. However, he also notes that safety might be improved by the use of fault preventative methods often used in the security domain.

2.5 Canonical Risk Analysis Process

An observation from the early work was that although texts talk about 'safety hazard analysis' and 'security risk analysis', the processes involved in each of the analyses are not actually dissimilar. If sections 2.2 and 2.3 are compared, we can see that they each have the same general stages:

1. An investigation of the system, its components, its functions, its connections to other systems, its environment, the users, etc. The information gained from this investigation is used to develop a model of the system. Although these investigations go by different names and vary slightly they are essentially **system modeling**.

2. A **qualitative analysis** of the weaknesses in the system, and dangers to its continued correct operation. In security these are called vulnerabilities and threats, in safety they are called failure mechanisms and hazards, but again, they can be considered to be alike.

3. A **quantitative analysis** of the weaknesses and dangers. Safety and security domains both attempt to quantify the risks using probabilistic approaches. In security this often results in estimated expected financial losses, while in safety this is

expressed as a likelihood of system failure resulting in an accident. Once more, in principle they are similar.

4. The final stage involves combining the qualitative and quantitative analyses to allow developers to make judgements about the measures that need to be put in place to counter the risks. Examples in safety are redundancy, protective equipment, monitoring devices, etc., while in security examples are access controls, firewalls, etc. In both cases, this last stage involves **defining the requirements** of the system that will ensure that the risks are reduced to acceptable levels.

While accepting that this is a simplification, we believe that that meaningful inferences can be drawn from it. Significantly, this initial observation supports the idea that safety and security *can* be integrated. The next important step is to determine exactly how best the two domains could be brought together.

2.6 Initial Discussion

The amount of published work on integration of safety and security analysis techniques is not large, but it covers a broad spectrum of approaches. These papers generally adopt one of two standpoints, either aiming to unify techniques into a single methodology, or investigating the similarities of techniques from each.

An observation made while reviewing these papers, in light of the earlier investigation of risk analysis, is that safety and security are closely related. As mentioned before, both deal with risks. Also, both safety and security risk analyses result in constraints (which may be regarded as negative requirements) that can conflict with functional and other performance-related system requirements [9]. Both involve protective measures, and both produce requirements that are considered to be of the greatest importance. These similarities indicate that some of the techniques applicable to one field could also be applicable to the other.

However, the conclusion that we draw from the analysis of the papers discussing 'integration' is that, while the definition of safety or security could be extended to include both concepts, in the majority of situations it is inappropriate to attempt to *unify* safety and security risk analysis techniques. Such unification would have the benefit of producing a single set of requirements, and conflict resolution could form an inherent part of the resultant approach, but this has to be weighed against the disadvantages. We believe that consolidation of safety and security could reduce developers' understanding of the system being analysed, and prevent a thorough analysis of either property. Specialised techniques in each domain have evolved with the aim of producing a thorough and complete analysis. Attempts to unify two such techniques would involve compromises in each, which in turn could lead to an *in*complete analysis, with subsequent safety and security risks going unobserved and being incorporated in the final system. An additional danger is that a unified approach might actually hide the requirements conflicts that it aims to resolve. Also, the process of resolving conflicts itself can actually be worthwhile, as it engenders better understanding of the system and its domain. This value could be lost in a unified approach. Finally, trade-offs between qualities could be hidden by a global abstraction, with unnoticed detrimental effects. For example, it is possible to increase reliability while decreasing safety without it being apparent that an increase in risk has occurred [1].

Our analysis leads us to believe that the value in integrating safety and security lies with harmonising techniques from each domain. Such an approach would provide numerous benefits without the disadvantages associated with unification approaches:

- The specialised techniques developed in each domain would not have to be compromised.

- Conflicts could become more apparent than if the techniques were applied in isolation, as comparisons between the two sets of requirements could be simpler.

- The cross-fertilisation of ideas from one domain to the other could promote better understanding of the system and its environment, and might lead to the recognition of risks that could otherwise be overlooked.

- Separation of properties would permit recognition of conflicts and trade-offs, and allow judgement-based decisions to be made, rather than have an 'automated' method make choices, and perhaps screen them from the system developers.

From our initial analysis of integrating safety and security requirements we conclude that safety, security and their associated risk analysis techniques are closely related and have sufficient similarity to make integration a reasonable and achievable goal. Further analysis shows that *unification* of methods has disadvantages that outweigh the benefits they may provide. Finally, *harmonising* safety and security techniques could provide advantages over independently producing requirements in each field, and would not present developers with the drawbacks of the unification approach.

3 The Case Study

3.1 Introduction

This section presents the case study that has been analysed as part of this work. The case study is a military air traffic control system (MATCS) that is due to undergo major modernisation in the near future. The purpose of this work is to replace obsolete hardware and software with current generation equipment. The system safety and security requirements for the programme have been produced via a lengthy analysis process.

In sections 3.2 and 3.3 respectively, we describe the development of the safety and security requirements. These requirements have been developed in isolation from each other, using established approaches from their respective disciplines:

- The safety requirements have been produced within a safety case framework. Preliminary Hazard Identification and systematic hazard analysis has been used to produce high-level safety requirements documented in a safety case.

- The security requirements have been produced via the development of system security policies. Security risk/threat assessment has been carried out using a qualitative approach incorporating evaluation criteria cont-ained in the UK Communications-Electronics Security Group (CESG) guidelines [10, 11].

3.2 Development of MATCS Safety Requirements

A four-part safety case, as defined in UK MoD Defence Standard 00-56/2 [12], has been adopted for the MATCS modernisation project. The main objective of the safety case is to produce arguments and evidence that the system is suitably safe for its intended purpose. The work that has been carried out to date has resulted in the production of the Safety Case Part 1, which contains the user-defined safety targets and high-level safety requirements. This section of the paper describes the process used to produce the safety requirements that are detailed in the Safety Case Part 1.

Safety Requirements Determination. The purpose of the safety analysis was to define the high level safety requirements for the system, which would in turn form the basis for the contractual safety requirements to be met by the implementation contractor. During the process of identifying the safety requirements, the following activities were performed:

Definition of the system boundaries and functions. The first stage of the safety analysis was to define the system boundaries, functions and interfaces. Much of the system functional model could be developed from the existing system; the replacement is required to provide the same functionality, with minor exceptions. From the required high level functionality, a Safety Context Diagram (SCD) was produced, showing the system and the interfaces with external agencies. From the SCD, more detailed Functional Interface Diagrams (FIDs) were developed, whose purpose was to detail the information that would be passed between the system sub-functions.

Preliminary Hazard Identification (PHI). Once the system functions had been identified, a PHI could be performed to record hazards or hazardous failures identified by the project team and operational experts. Known hazards from similar systems were also taken into account, and available documentation on the current system was examined to identify potential future hazards. Although a systematic Preliminary Hazard Analysis (PHA) was expected to identify functional failures of the system, the 'brainstorming' aspect of the PHI was carried out to ensure that known hazards in the existing system were taken into account. As well as specific discussions to identify hazards, based on the system functions, ad-hoc questions were raised and potential hazards identified.

Preliminary Hazard Analysis (PHA). Following the PHI, a PHA was undertaken. The PHA was expected to identify the functional failures that could lead to hazardous states. Once this analysis had been carried out, a process of rationalisation took place to relate the PHA-identified hazards to those from the PHI, removing any duplicates and producing a set of system hazards that could be used to generate the safety

requirements. The primary tool used to perform the PHA was Functional Failure Analysis (FFA). The FFA used a systematic approach to determine the impact of failures of each identified function. The basis for discussion involved in the FFA was the SCD and FIDs. Each of the potential outputs shown on a diagram is investigated to determine the consequences of failures of the output, the effect on the function, and the influence on the system as a whole. The ways in which outputs could fail were: data not provided, erroneous data provided, and data output being delayed. For every data output of a sub-function, each failure type was considered in turn to determine whether the subsequent functional failure modes could affect the safety of the system.

Generating the MATCS Safety Requirements. The final safety requirements presented in the Safety Case Part 1 were derived from a number of sources, including the analysis described above, consideration of the system as a whole (including expert knowledge of the existing system), good engineering practice and guidance provided by the Independent Safety Advisor (ISA). As well as defining requirements to avoid the functional failure modes highlighted by the FFA, the general safety requirements (such as Safety Integrity Levels for software) were defined, as were human factors requirements (for which a separate analysis was carried out). In this way, the Safety Case Part 1 provided the basis from which further, implementation-specific, analysis could be carried out, and against which the implementation contractor could demonstrate the safety of the proposed solution.

3.3 Development of MATCS Security Requirements

While work was being carried out to produce the high-level safety requirements for MATCS, a separate team was developing the security requirements for the project. These requirements were generated and documented via a System Security Policy (SSP). As for the safety requirements, this policy was produced according to government guidelines; in this case, those of the CESG [10]. In accordance with these guidelines, electronic communications and computer systems handling protectively marked information should be accredited, to ensure that their use does not present an unacceptable risk to national security. The basis for this accreditation is the SSP, which represents an agreement between the project manager and the accreditor with respect to system security.

Security Requirements Determination. The SSP describes the scope of the system, the nature of the security requirement and the specific security measures that are to be implemented. The elements of an SSP include:

Definition of the MATCS for Security. This was a concise definition of the system, encompassing security-relevant information, accounting for the role of the system, the data to be handled, the number and security clearance of the users and the system configuration. The system is defined using free text and diagrams. The role and physical distribution of elements and sub-systems is exactly the same as for the

generation of the safety requirements. However, the numerous diagrams and text are much less readily understood and intuitive than the SCD.

Security Threat Assessment for MATCS. The threat assessment was carried out following the CESG guidelines. This process is similar to that mandated by the US Department of Defence 'Orange Book' [13] but takes into account the later work of the European ITSEC standard [14]. The first stage was to identify the threat sources, such as hostile intelligence, disaffected users, erroneous system operation, etc. Each of these were then investigated in turn to determine the possible origins of the threat and methods of attack that the system may have to face. From this, the rationale for a security system was formed. In the case of the MATCS, the most significant threat was determined to be authorised users who, for whatever reason, disrupt the system or compromise information. Having considered the threats, it was necessary to carry out an evaluation to determine the level of assurance that was required to protect the system. This evaluation took into account the threats, the system vulnerabilities (numbers of authorised users, modes of operation, clearances and technical characteristics of the system), and marking and/or sensitivity of the information. With the system information at hand, tables in [11] could be referenced to determine the required assurance level. This took a value in the range E0 (low) to E6 (high).

Definition of the MATCS Security Measures. The final stage was to describe the measures necessary to achieve the required level of security, as determined by the threat assessment. This defined what would traditionally be called the system security requirements. The measures were organised by stating the security principles and risks which had been associated with each of the eight ITSEC-identified general security principles (access control, authentication, accounting, audit, object reuse, accuracy, reliability of service and data exchange). Following this, the SSP specified how the risks would be addressed, either by means of assertions, or by measures to be applied (technical or procedural).

4 Case Study Analysis and Discussion

4.1 Introduction

Having described the case, this section documents the analysis carried out in the context of the concepts and ideas presented earlier in the paper. The analysis begins by presenting the authors' views on the concept of integration of safety and security requirements (4.2): the idea that the requirements themselves can be conflicting and/or inconsistent; but also that the requirements are inter-related. This relationship needs to be understood if successful integration is to be achieved. Following this, the safety and security requirements generation processes from the case study are analysed and documented (4.3, 4.4). The final section (4.5) presents a number of recommendations, with respect to the case study, that could allow the safety and security specifications to be better integrated.

4.2 Interaction between Safety and Security

From the initial literature review and investigation of the case study, two kinds of interaction between safety and security requirements have been identified.

The first form of interaction is the obvious case where requirements from each of the two domains are incompatible. As an example, we can consider a computer-controlled door locking system in a modern building. A requirement from the safety domain could be that the system must 'fail safe'. In this case, this would mean that on failure, the doors should be unlocked, thereby allowing personnel to vacate rooms in the event of an emergency. On the other hand, the security developers may identify a requirement for the system to 'fail secure', that is; all the doors should be locked to prevent unauthorised access. Although there are technical solutions to this particular example, it serves to demonstrate how two groups of developers working in isolation can produce incompatible requirements.

The second form of interaction is subtler, and generally not as easily recognised. Here we are considering the interaction that produces what we can call 'primary' and 'derived' requirements. Primary requirements are those that have their foundations in their own domain. For example, primary security requirements have their foundations in the security domain. From investigation of security threats, requirements engineers identify those security measures needed to counter the threats. We might say 'somebody may wish to steal or corrupt this data, so we must control access, limiting it to those personnel that we trust'.

We can think of derived requirements as those that are brought about (their essential rationale) as the result of analysis undertaken *outside* their own domain. An example of this could be *security* requirements identified as the result of *safety* analysis. In this case, these requirements are not specified to protect the system from traditional security threats per se, but rather have been identified as part of the process of reducing the threats to the system's safety performance. We might say 'if this particular data is corrupted in any way, the overall safety of the system could be compromised. Therefore, we must make sure that the equipment cannot fail in a way that causes data corruption, but we must also control access so that people who should not be tampering with it cannot cause a corruption'. In these examples, both primary and derived requirements result in access controls, but there are important differences in the rationales.

These requirements interactions need to be fully understood for two main reasons:

Conflict Resolution. If conflicts are to be properly dealt with, developers need to appreciate how they have arisen. Only then can they be sure that the processes they use to resolve them do not introduce new problems. With the first type of interaction, this should not be a problem. However, the subtle nature of the second form of interaction means that it needs more careful handling; changes to derived requirements could affect not only the parent document, but also the analysis and requirements in the associated opposite domain.

Integration of Requirements. Not until requirements interactions are understood can processes to *integrate* them effectively be developed.

4.3 Safety Case Development

The process used to develop the MATCS safety requirements has followed established and well-documented procedures. A preliminary hazard identification was followed by systematic analysis, all documented via a Safety Case Part 1.

In the Safety Case Part 1, among the General Safety Requirements there is the following conflict resolution policy: "Wherever there is a conflict between the safety requirements and other requirements for the system, the safety requirements shall always have precedence". However, in the System Attribute Safety Requirements, we find a subsection titled 'security requirements' which calls for:

- Access controls to be applied to the operational and system management interfaces.
- Appropriate mechanisms for maintaining secure copies of system configuration and software versions, and for verifying that they have not been altered.
- Sub-contracted systems and commercial-of-the-shelf (COTS) equipment to be free from unapproved or hazardous features including software viruses.
- Security measures or devices implemented in the system not to affect the safety performance of the operational system.

These security requirements and the placement of them in the safety documentation raise two issues:

- First, from where were these specific security requirements derived and why specify these requirements in particular? Traceability within the safety case documentation is incomplete, making it difficult to determine the rationale for specific requirements.
- If a conflict were to occur between one of the derived security requirements and a requirement contained elsewhere in the Safety Case, which would have precedence? This highlights a weakness of a simplistic approach to conflict resolution.

4.4 Security Policy Development

The security policy document states: "In the event of a conflict between security and safety requirements it shall be presumed that safety has precedence until a ruling has been obtained from the System Manager". This differs from the safety documentation's approach to conflict resolution, in which safety requirements *shall always* have the highest precedence.

4.5 Discussion

From investigation of the literature and the case study, we can identify a number of areas that can cause problems when we attempt to harmonise safety and security requirements techniques. The most important ones are:

- Different system models developed for safety and security.
- Different documentation structures for the analyses and their results.
- The interaction of safety and security requirements.
- Isolation of safety and security requirements processes.

System Models. We have found that security system models and safety system models can differ greatly. The problem with having broadly differing abstract system models is that they lead developers to take different views of the system. However, the physical system for which the case study safety and security requirements were produced was the same for each development team. Production of different models hindered communication as well as being a duplication of effort. Although it may be essential to have different models to reflect the different concerns of safety and security, it must be possible to map between them.

Documentation Structure. Documentation of analyses can differ greatly between safety and security. The problem with these differences is that they can make it difficult to find and compare requirements. We believe that it would be relatively straightforward to harmonise safety and security documentation, with few, if any, problems. Having similar documentation structures would allow easier identification of conflicts and subtle differences in things like the wording of concepts and the levels of detail.

Interaction of Requirements. In section 4.2, we described two ways in which safety and security requirements can interact. The first of these was a simple conflict of requirements, while the second was a more subtle form of interaction due to the way in which security could support safety, and vice versa. We have been unable to identify problems of the first type in the case study. As might be expected of a project that has undergone such extensive work, cursory reading of the documentation has revealed no obvious requirements conflicts. There is, however, a clear example of the second type of interaction as described in section 4.2. The safety requirements contain a number of derived security requirements. There might also be cases where safety could support security, although we could find no examples of this in the case study. In order to identify interaction of safety and security requirements, there needs to be improved traceability in the documentation, particularly where primary and derived requirements are involved. Requirements in safety documentation must be cross-referenced to security analysis, and vice versa, so that the effects of changes can be fully evaluated.

Interaction of Requirements Processes. The final topic we wish to discuss concerns the interaction of the requirements generation processes themselves. It is now widely accepted that in generating requirements an iterative lifecycle is required, rather than the long-standing 'waterfall' lifecycle. But what about the effects that changes to the safety requirements can have on the security requirements, and vice versa? As well as the processes being iterative within their own domain, we believe that they also need to be 'cross-iterative'. By this we mean that changes in one set of requirements may need to be investigated in *both*. Requirements engineers need to be aware of this interaction if costly errors are to be avoided in system development, and analysis is not to be undermined.

5 Conclusions and Future Work

We can summarise the findings of this paper as follows. If safety and security requirements are defined in isolation from each other, there is the danger that unrecognised, and therefore unresolved, conflicts or inconsistencies between them will arise, as demonstrated in the case study. Some form of integration is therefore required; but it is neither practical nor desirable to unify the two kinds of requirement into a single process. The most appropriate form of integration is the harmonisation of the two processes, enabling any conflicts to be recognised and resolved.

Our suggested approach to harmonisation is, first, to identify appropriate relationships between the stages and documentation of the safety and security requirements processes. We identified in section 2.5 a close correspondence between the processes, which should make this possible. It will then be necessary to introduce, into the over-all systems requirements process, additional steps for the identification and resolution of conflicts and inconsistencies. In this way the aim of integration can be achieved.

In this work we have only begun the task of enabling the integration of safety and security. Further work is required:

- To verify, by examination of other systems, that the concerns which we have raised here are generally applicable.
- To make concrete proposals for the alteration of the general systems requirements process to enable integration of safety and security.
- To verify, on new systems developments, that the proposals are practicable and do indeed provide the benefits that we envisage.

We believe that progress in this work will make a contribution to the production of genuinely dependable systems for the future.

6 References

1. Leveson, N., G.: Software Safety: Why, What and How. In: ACM Computing Surveys, Vol. 18, No. 2 (1986).
2. Pfleeger, C., P.: Security in Computing. Prentice Hall Inc (1997).
3. Avizienis, A., Laprie, J. C. (eds.): Dependable Computing for Critical Applications. Springer-Verlag/Wien (1991).
4. McDermid, J., A.: On Dependibility, its Measurement and its Management. In: High Integrity Systems, Vol. 1, No. 1 (1994).
5. Sanders, W., E., Meyer, J., F.: A Unified Approach to Specifying Measures of Performance, Dependability and Performability. In Dependable Computing for Critical Systems. Springer-Verlag/Wien (1991).
6. Cullyer, J.: The Technology of Safety and Security. In: The Computer Bulletin, Vol. 5, No. 5 (1993).
7. Rushby, J.: Critical Properties; Survey and Taxonomy. In: Reliability Engineering and System Safety, Vol. 43, (1994).
8. Brewer, D. F. C.: Applying Security Techniques to Achieve Safety. In: Directions in Safety-Critical Systems, Proceedings of the Safety-Critical Systems Symposium, Bristol 1993. Springer-Verlag London Ltd (1993).
9. Leveson, N., G.: Safeware, System Safety and Computers. Addison-Wesley Publishing Company Inc (1996).
10. CESG.: CESG INFOSEC Memorandum Number 5 - System Security Policies, Issue 3.0 (July 1994).
11. CESG.: CESG COMPUSEC Memorandum No 10 - Minimum Computer Security Standards for HMG Information Handled by Information Technology Systems, Issue 2.2, (October 1996).
12. UK Ministry of Defence: Defence Standard 00-56/Issue 2 (DS 00-56/2), Safety Management Requirements for Defence Systems, dated 13 December 1996 (1996).
13. Department of Defense Trusted Computer System Evaluation Criteria. US Department of Defense (1985).
14. Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model, Common Criteria Implementation Board. CCIB (96/011) (1996).