Safety Protocols: a New Safety Engineering Paradigm

Tony Cant and Brendan Mahony

Command, Control, Communications and Intelligence Division
Defence Science and Technology Organisation
PO Box 1500, Edinburgh, South Australia 5111
Email: Tony.Cant@dsto.defence.gov.au, Brendan.Mahony@dsto.defence.gov.au

Abstract

The field of *system safety* looks on the surface to be a mature discipline based on everyday intuitions about safety risk. System safety looks at potential accidents that could arise due to system behaviour. It is based on the notion of *system hazard*. In this paper, we look at the theory and practice of system safety. We propose a model of system safety behaviour suitable for describing and evauating the goals and processes of safety engineering. We argue that the notion of hazard is not appropriate as the central pillar of safety engineering and that it can actually be misleading. Instead, we propose that safety engineering is better served by a focus on *safety constraints*. To illustrate the benefits, we consider an approach to "hazard analysis" that begins by simply identifying all the dangerous physical flows in the systems intended environment and proposing a *safety policy* for managing them. Safety engineering then proceeds with the elucidation of *safety protocols* that coordinate the various systems in the environment in operating safely within the proposed policy constraints. We illustrate our approach using a case study.

Keywords: Safety case, safety assurance, rapid acquisition, urgent operational requirements.

1 Introduction

The field of *system safety* — as described, for example, in Leveson's well-known textbook entitled "Safeware" (Leveson 1995) and by safety standards such as MIL-STD 882C (Department of Defense 1993) — involves an approach to safety engineering that is very familiar (especially in the USA) and apparently well-understood. It looks primarily at potential accidents that could arise due to system behaviour and its most basic tool is the pervasive and widely-used concept of "hazard". System safety purports to be the technical expression of everyday intuitions about safety: co-opting every-day terms (such as "hazard") and imbuing them with elaborate technical interpretations.

Although the field of system safety looks on the surface to be a mature discipline, experience with a number of Defence projects suggests that Safety Programs are deficient in their safety arguments with uncomfortable frequency. Many safety programs strive to follow the processes required by (say) MIL-STD 882C (Department of Defense 1993): a great deal of analysis is done and diligently reported. Unfortunately, on close examination, the safety arguments can sometimes boil down to little more than "a great deal of analysis was done and diligently reported." As observed by the Nimrod Review, "... the task of drawing up the Safety Case became essentially a paperwork and tick-box exercise." It seems that the process of system safety can be easily abused. What is the reason for this?

One basic reason is that the field of system safety still lacks clear agreement on basic terminology. The well-known text by Leveson (Leveson 1995) does (for the most part successfully) attempt to provide clear definitions — but these are not introduced until Chapter 9. As forums such as the High Integrity Mailing List (Kelly 2011) demonstrate, the definitions of fundamental concepts in system safety are still the subject of much debate.

A more serious issue is discussed in this paper. We argue that the notion of hazard is not a useful one and that it can actually be misleading. In place of the unending hunt for the hazard, we propose that safety engineering should be carried out through the positive proposal of safety policies for dealing with dangerous physical flows and of safety protocols that coordinate interactions between systems so as to implement said safety policies.

This paper is structured as follows. In Section 2 we look at the terminology of system safety. In Section 3 we discuss a range of issues relating to hazards and hazard analysis. Section 4 gives a brief overview of Leveson's more recent approach to accident modelling and hazard analysis. In Section 5 we describe briefly the approach taken by the recently published DEF(AUST)5679 (Department of Defence 2008*b*). Then in Sections 6–8 we describe the notion of safety protocols. We illustrate our arguments in Sections 10–12 using a case study. Finally, Section 13 presents some concluding remarks.

2 System Safety

The primary driving concept in system safety is that of the *accident*. For a given system, the first key step in safety engineering is to consider the possible accidents to which system behaviour could contribute. As defined by Leveson (Leveson 1995):

> **Definition**. An *accident* is an undesired and unplanned (but not necessarily unexpected) event that results in (at least) a specified level of loss.

Succinctly put, an accident is an undesired loss event. One may debate whether or not loss of equipment or capability — as opposed to harm or loss of life — should be included in system safety engineering, but this is a minor consideration. We do not believe that the notion of accident is controversial: it has a meaning in everyday life but also makes sense as a technical concept.

Also familiar is the notion of *accident severity.* This characterises the damage that may be done by the loss event and is often used in safety engineering to rate accidents, possibly with a view to allocating more effort into protecting against the more severe possibilities.

We now turn to the notion of *hazard*, which is more problematic. In everyday life the notion of hazard is commonly used and seems to be well understood. We are familiar with the following examples:

1. "Smoking in the toilets is a fire hazard and smoke detectors have been fitted" (aircraft safety announcement);
2. "Confined space. Hazardous Atmosphere. Check oxygen level before and during entry" (warning sign);
3. "Ice on road. Hazardous driving conditions." (road sign); and
4. "Tripping Hazard", "Biological Hazard", "Electrical Hazard", "Overhead Hazard" (other warning signs).

Intuitively speaking, a hazard is a situation from which it is sufficiently likely that an accident could arise. To take the first example, it is easy to imagine that a cigarette butt that is carelessly disposed of in the wastepaper bin in an aircraft toilet could quickly lead to a fire that would threaten the safety of the aircraft. The announcement both warns of this hazard and also states that this hazard will be quickly detected (and thus dealt with by the cabin staffor automatic systems).

Thus in common parlance the notion of hazard is usually associated with some dangerous physical substance or release of energy (*dangerous flow*). In developing the field of system safety, it has been thought essential to retain the hazard as a central concept. However, this common notion of hazard has generally not been thought to be sufficiently powerful. Systems may be involved in accidents in many ways, even if they do exhibit such dangerous flows. Just consider an air traffic control system: the physical dangers intrinsic to the actual system equipment pale in comparison to its potential to do harm in the wider air traffic environment. Thus, considerable effort has been made to expand the definition of hazard to encompass all forms of dangerous interaction with the environment.

As defined by Leveson (Leveson 1995):

> **Definition**. A *hazard* is a state or set of conditions of a system (or object) that, together with other conditions in the environment of the system (or object), will inevitably lead to an accident (loss event).

As Leveson points out, implicit in this definition is that hazards must be determined with respect to the particular environment of the system. Hazards occur at or within the system boundary, which must be well defined, and may interact with other systems in the environment, which remain vague, in causing an accident. Leveson's definition is similar to most definitions of hazard, of which we quote just two:

> **Definition**. A *hazard* is a physical situation or state of a system, often following from some initiating event, that may lead to an accident (Ministry of Defence 2007)

> **Definition**. *System Hazards* are top-level states or events from which an accident, arising from a further chain of events external to the System, could plausibly result (Department of Defence 2008*b*).

This expanded notion of hazard takes a central place in modern system safety practice (as, for example, described in Leveson's book). Much of the effort applied in a safety program is devoted to the identification and assessment of hazards. This effort is called *hazard analysis* and involves techniques such as *Fault-Tree Analysis*(FTA) or *Failure Modes and Effects Analysis* (FMEA). On the one hand, FTA analyses the causes of hazards by reasoning backwards from a given top-level state (or event), using Boolean logic to describe how low-level events (which can be normal events or failure events) combine to bring about the hazard. On the other hand, FMEA reasons forward from low-level failures to determine how they may lead to system hazards.

Leveson (Leveson 1995) defines failure (a concept familiar in reliability) as follows:

> **Definition**. *Failure* is the non-performance or inability of the system or component to perform its intended function for a specified time under specified environmental conditions.

In fact, most of the techniques used in hazard analysis are borrowed and adapted from reliability engineering and depend on the concept of failures at least as strongly as on the concept of hazards.

Also striking is the degree of low-level information required by existing hazard analysis techniques. The design of the system must be quite well progressed for such techniques to be truly effective.

The system safety effort is usually directed through some form of probabilistic risk assessment in which a "hazard risk index" (HRI) is determined by a combination of hazard frequency and accident severity. If an HRI is too high, the risk may be regarded as unacceptable, or acceptable only with further measures designed to build in safety. Although this notion of hazard frequency is a natural one for simple physical hazards, it is harder to understand for the generalised notion of hazard adopted in system safety. It seems generally acknowledged that it is not sensible for hazards related to software behaviour and the Joint System Safety Handbook (Department of Defense 2010), for example, recommends that the notion of Software Control Category be used instead.

3 The hazards of "hazard"

In this paper we pose the specific question: *is this generalised notion of hazard suitable as the central pillar of modern safety engineering?* It is our contention that it is *not*, and that we need something better to guide our thinking.

The most important deficiencies are the following.

What we want vs what we have. Hazard analysis leads us to confound two quite different issues: what the system (imagined but not yet built) is required to do versus what the system actually does (as built). On the one hand, hazard analysis aims to determine system safety requirements, acknowledging that safety must be "designed" in to the system. On the other hand, hazard analysis uses techniques that, to be effective, require deep knowledge of how the system actually works.

Failures are not the whole story. The fixation that can be seen on "failure" of system components or items of equipment as potential root causes of hazards clouds the distinction between reliability and safety and leads to an emphasis on what Leveson (Leveson 2011) calls *component* *failure accidents*. However, an accident can also arise from "dysfunctional interactions between components" even if the components are working reliably. Such accidents are called *system accidents* or *component interaction accidents*.

Failures distort the story. Safety engineering is unusual in that prime focus is given to what can go wrong: that is, what is *not* required of the system. Usually, engineering concentrates on what *is* required of the system. Often "what can go wrong" is a much bigger and more imaginative world than "what we want." Failures thinking can lead to consideration of behaviours that are conceivable but disallowed by existing design constraints; it makes it hard to decide how much hazard analysis is enough; and it can even lead to extended consideration of failures that have no safety impact. Such an approach is a contributing factor to "laborious, discursive, document-heavy" safety cases — a key deficiency identified by the Nimrod Review (Haddon-Cave 2009).

Hazard analysis tends to be inward looking. The failures-focus of hazard analysis techniques makes them inward looking, concentrating on how problems within the system may lead to accidents in the environment. This makes it hard to describe safety functionality in terms of the system interface; contributing to the notorious non-compositionality of safety cases. Compositionality fundamentally relies on "plug and play" interface specifications.

Hazards are more complex than they seem. Superficially, the notion of hazard seems simple enough, but a little thought about its practicalities shows that it is hiding a great deal of complexity. Exactly what systems states may contribute to accidents is context sensitive and interacts with the notion of causality in subtle ways. Obviously, dangerous flows from the system are hazards. Equally, any system flow that may directly cause a dangerous flow in the environment is a hazard. Moreover, any system flow that directly causes an event in the environment that directly causes a dangerous flow is a hazard and so on and so forth. This kind of reasoning may be iterated to arbitrary numbers of intermediate events. In algorithmic terms the definition of hazard is quite complex indeed.

When do we stop? The iterative nature of the definition of hazard can potentially lead to the elaboration of very complex accident scenarios that can be hard to understand and may be considered "unlikely" on no better grounds than the number of intermediate events required. It also offers no guidance when to stop, making it very hard to be confident that all hazards have been enumerated.

How do we find hazards anyway? The determination of hazards involves a search for chains of events in the environment that may result in accidents, but says little about how to structure this search. In practice, the search becomes something of an imaginative process, usually structured only by guide words suggestive of possible ways in which things could go wrong. Again this can make it hard to be confident that all hazards have been enumerated.

Logical hazards complicate things. For certain kinds of systems, such as command-support systems, air traffic control systems etc, more subtle "logical" hazards relating to information flow tend to dominate. Logical hazards are difficult to understand or even to define, as there may be many levels of causal indirection between the "hazard" and the dangerous physical or material flows that it may eventually trigger. Such non-physical hazards may be problematic for existing hazard analysis techniques because they will not always be amenable to guide word analysis.

Software is not stochastic. Since we are ultimately concerned with assessing safety risk, there may be a tendency to assign probabilities to hazards (or to lower-level states or events). Such probabilities are dubious in the case of software-intensive systems. As Leveson (Leveson 1995) points out:

> Risk assessment is currently firmly rooted in the probabilistic analysis of failure events. Attempts to extend current probabilistic risk assessment techniques to software and other new technology, to management, and to cognitively complex human control activities have been disappointing.

Experience with a number of Defence projects suggests that the role of software in system safety is not always treated with sufficient care.

What do fault trees mean? The elucidation of lower-level hazards via techniques such as FTA is "handraulic" and not amenable to tool support. Attempts to provide a formal semantics for fault trees have met with limited success (Schellhorn et al. 2002).

For the above reasons, although we acknowledge that the concept of hazard is widely used in system safety and that many are comfortable with it, we believe that the notion of hazard is neither a useful nor helpful concept when we are looking for fundamental notions in system safety. At best we can say that hazards are only a means to an end, and play a role only as an auxiliary concept used to facilitate thinking in safety engineering.[1]

4 The STAMP Approach

As observed above, some of the issues and problems with the conventional approach to system safety — such as the heavy emphasis on failures — have already been pointed out by Leveson (Leveson 1995). Leveson has written a soon to be published new textbook, currently available on her web site in draft form (Leveson 2011). In this book, Leveson has proposed a new model for accidents and a new way of thinking about system safety. Her accident model is based on systems theory and is called *System Theory Accident Modelling and Processes* (STAMP).

In STAMP, accidents occur when external disturbances, component failures, and/or dysfunctional interactions among system components are not adequately controlled. Safety is viewed as a control problem for an adaptive socio-technical system. In such a framework, understanding why an accident occurred requires determining why the control structure was ineffective.

In systems theory, control is always associated with the imposition of *constraints*, which play a vital

[1] It is interesting to note that the *OHS Act (Com) 1991* does not make fundamental use of the concept at all (except for the use of the term "high-hazard" facilities).

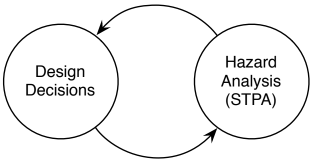

Figure 1: STPA-based design (Leveson 2011).

role in the STAMP approach. Accidents are considered to result from inadequate enforcement of constraints on behaviour at each level (e.g., technical, managerial, regulatory) of a socio-technical system. An example of a (physical) safety constraint is: “the power must never be on when the access door is open”. Preventing accidents now and in the future requires a control structure that will enforce the necessary constraints.

Leveson describes a new approach to hazard analysis called STPA — which originally stood for STAMP-Based Hazard Analysis but has now been changed to System Theoretic Process Analysis. STPA is meant to extend conventional hazard analysis to cover new factors such as design error, software flaws, component interaction accidents and social and human processes. There is still the notion of system hazard and component hazard. System safety requirements and design constraints are also important concepts. However, the role actually played by hazards within the STPA approach is not very clear.

What is clear is that Leveson envisages STPA reaching deep into the system design process in a tightly coupled feedback loop as shown in Figure 1. This is one of the most puzzling aspects of STAMP, given Leveson’s stated concerns (Leveson 2011, Ch. 6) over the tendency to isolate, misdirect, and delay safety efforts; what might be termed the too-much/too-late approach to safety. If essentially open-ended hazard-analysis efforts are required after every design decision — because hazard analysis eventually requires total knowledge of design detail — then it is little wonder that the tendency is to postpone safety efforts until the very end.

In later chapters, Leveson discusses the formulation of safety constraints using, as a worked example, the collision avoidance system TCAS II for aircraft (Leveson 2011). Leveson provides detailed specifications for the constraints (safety-related or not) and assumptions and limitations of the full socio-technical system in which TCAS II operates.

We are not going to discuss the STAMP and STPA approaches further. However, it is important to take away the following lessons: that Leveson thinks it desirable that the notion of hazard be de-emphasised and the notion of requirement or constraint be given a more prominent role.

5 DEF(AUST)5679

The “normal” approach to system safety — along with the heavy reliance on the notion of hazard — is reflected in most existing safety standards. DEF(AUST)5679 (now at Issue 2 (Department of Defence 2008*b*)) was written (at least in part) in an attempt to provide an approach to system safety driven more by safety requirements.

DEF(AUST)5679 provides requirements for the structure of the *safety case* (an evidence-based argument for safety). Safety case development is structured into three phases with associated reports (see Figure 2):

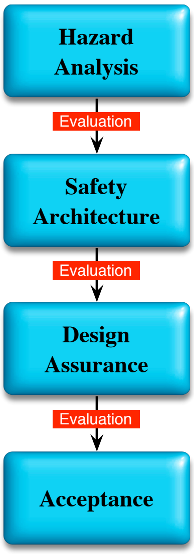

Figure 2: DEF(AUST)5679 Safety Case Development phases (Department of Defence 2008*b*).

Hazard Analysis – assess the danger (or threat to safety) that is potentially presented by the system;

Safety Architecture – demonstrate that the overall system is designed to be safe; and

Design Assurance – demonstrate that the components are designed to be safe.

In the hazard analysis phase, the system *interface* is defined in terms of inflows and outflows exchanged with the environment. Other systems present in the environment (the *operational context*) are described in appropriate detail. A hazard analysis then determines the ways in which the system, in its operational context, may contribute to an accident. The outputs of hazard analysis are as follows.

Accidents – external events that could directly result in death or injury.

Severities – a measure of the degree of seriousness of accidents in terms of the extent of injury or death that may result.

Hazards – states or events at the system interface from which an accident — arising from a further chain of events external to the system — could conceivably result.

Accident scenarios – a causally related mixture of system behaviours (*hazards*) and environment behaviours (*coeffectors*) that may culminate in an accident.

Thus, DEF(AUST)5679 adopts a fairly traditional notion of hazard with the conceptualisation of accident sequences taking a mandatory role in the hazard analysis phase. However, hazards turn out to play a limited role compared with much of current practice. Inward looking hazard analysis plays no role; hazard analysis is outward looking and is merely used to assist in determining what constitutes safety for the given system. Once this is achieved, the notion of hazard is no longer used.

The safety architecture phase begins with the development of a collection of *system safety requirements*. The system safety requirements are expressed

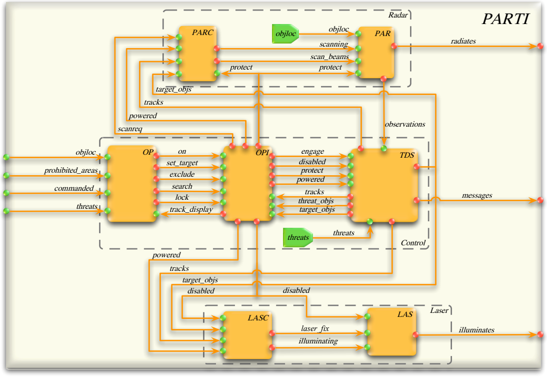

Figure 3: A safety architecture presented as a block diagram.

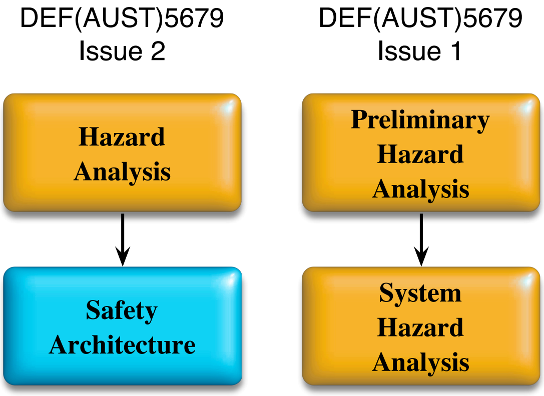

Figure 4: The hazard analysis phases of DEF(AUST)-5679-Issue 1 compared to Issue 2.

in terms of the system interface and collectively ensure that the system hazards do *not* occur. During subsequent system development, they are treated much like other system requirements.

So as to clarify the basic safety functionality of the system, the Safety Architecture decomposes the system into *components* (Figure 3 shows an example safety architecture (Mahony & Cant 2008)). The interaction between these components is described and they are assigned *component safety requirements* in order to discharge the system safety requirements. Finally, a *correctness* argument is made that shows how these component safety requirements ensure satisfaction of the system safety requirements (this is called *architecture verification*).

In the last phase of *design assurance*, the components are modelled to an appropriate level of detail and shown to satisfy their component safety requirements both through verification arguments on the models and through extensive testing.

In DEF(AUST)5679-Issue 2, *hazard analysis activities have no mandated role in either safety architecture or design assurance*. Instead, the safety requirements identified by hazard analysis are simply flowed down through subsequent phases. This is in sharp contrast to the approach adopted in Issue 1 of DEF(AUST)5679 (see Figure 4). Issue 1 mandated two hazard analysis phases: a Preliminary Hazard Analysis which looked from the system boundary out and a System Hazard Analysis which looked down into the system for dysfunctional interactions between components and for failures of individual components. Replacing System Hazard Analysis with Safety Architecture has the potential to make Issue 2 safety cases shorter, easier to understand and more convincing.

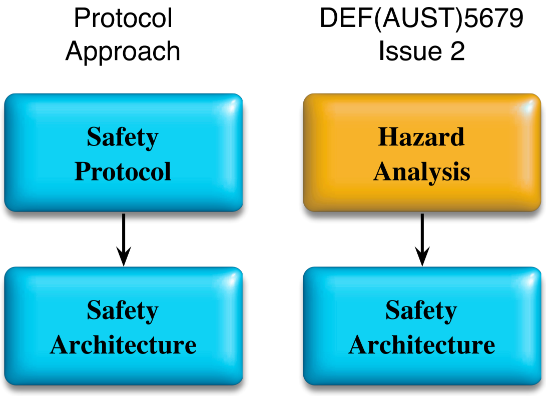

Figure 5: The protocol model compared to DEF-(AUST)5679-Issue 2.

The authors are currently carrying out a systematic application of DEF(AUST)5679-Issue 2 to the construction of the safety case for a real Defence system. In the course of this work, the essentially arbitrary nature of hazard analysis has become increasingly clear. This led to consideration of the potential feasibility of adopting DEF(AUST)5679's requirements flow-down model even earlier in the development cycle than safety architecture, replacing Hazard Analysis with Safety Protocol development as shown in Figure 5.

The goals of the Safety Protocol phase are essentially the same as those of Hazard Analysis, that is to identify potential accidents that may arise from the system operating in its intended environment and to propose system safety requirements that the system needs to satisfy to avert these accidents. However, instead of focussing on hazardous behaviours to be avoided, Safety Protocol development focuses on identifying safe behaviours to be adhered to.

In the following sections we briefly describe a model of system safety behaviour and then use it to describe the processes and outputs of Safety Protocol development.

6 A Simple Model of System Safety

We begin with a brief consideration of the setting in which safety engineering proceeds, describing a simple, generic model of system operation that is analogous to the system architecture model underlying DEF(AUST)5679 safety architecture. This model is used to structure the development of system safety requirements in much the same way as the architectural model structures the development of component safety requirements.

Suppose that we wish to engineer and operate a system S safely within a wider environment E. In general, the elements of E that will bear on safety include the following:

- the new system S;
- a collection of other systems (engineered elements) $\{S_1, \ldots, S_n\}$;
- a collection of humans (more generally protected elements) $\{H_1, \ldots, H_m\}$; and
- a physical *medium* (for example, the ocean or the atmosphere) M, in which these entities interact.

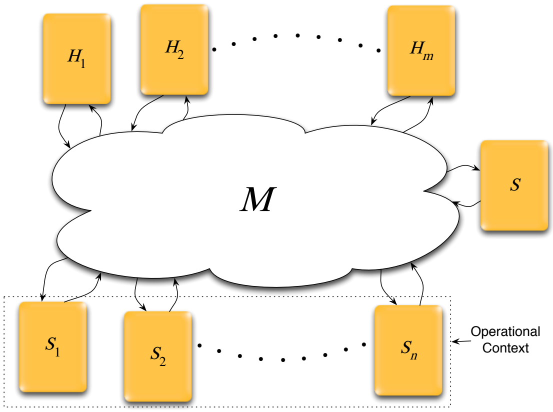

Figure 6: The structure of the environment.

Each element of E will have associated observable *inflows* and *outflows*: whether qualitative or quantitative; physical or logical; state or event based. We call these associated flows the *interface* to the element. This is a familiar concept for the engineered systems in the environment, but it is also readily applicable to the human elements and even the medium. The overall situation may be depicted in block diagram form as shown in Figure 6

Which humans should be included in the environment? Generally, the humans will comprise an undetermined number of potential bystanders. It may be convenient to aggregate such bystanders into a single representative block. However, some humans may have well defined roles for observing and/or controlling various systems in the environment. Such roles may be represented in the environment as specialised human blocks or separated out into system blocks that are implemented using human operators; depending on the nature of the safety functionality inherent to the role.

When considering the safety of human elements of E, the primary outflow of interest is the health status of the individual. The inflows of interest comprise the impact of *potentially* harmful energy on the individual. When these dangerous flows rise to *actually* harmful levels an accident can be said to occur, so we call the corresponding inflows accident flows and will generally represent them as event flows where the events represent the occurrence of accidents.

Which systems should be included in the environment? At a minimum, they should include all systems that S is intended to interact with, even indirectly, as all such interactions should be open to analysis for safety implications. In DEF(AUST)5679 parlance, this collection of systems is called the *operational context* of S. Ideally, the operational context would comprise some form of platform or system-of-systems which has an existing well-defined safety case and to which S is to be integrated.

When considering the safety of systems acting in E, the outflows of interest are those that may interact with the humans in the environment. The most obvious such flows are the dangerous flows that may emanate from the system, but control flows such as safety enclosures, interlocks, marked boundaries, warning signs, alarms, etc may be of significance.

If dangerous flows from one or more systems should be transmitted to a human element at a harmful level, then the corresponding accident event will be triggered. The precise mechanics of how dangerous flows from a number of systems are transmitted and aggregated through the environment is determined by the medium M. For example, there may be a number of radar sources in a given environment. The resultant radar intensity at each point in the environment is determined by the medium according to the power and direction of the signals from the source systems. An accident occurs when a human is positioned in the environment where the intensity of signal exceeds safe levels.

While the list of environmental flows that are known to be harmful is quite long: heat energy, kinetic energy, gravitational potential, poisons, explosive substances, etc; it is certainly finite and guidance can be found from many sources (Comcare 2007, Royal Australian Navy 2006, Department of Primary Industries 2007). Determining the dangerous flows for a given system is little more complex than running down a checklist. Once the dangerous flows have been identified, the “hazard analysis” part of our approach is over. Instead, we move our attention to the question of how safely to operate the various systems in the environment.

7 The Safety Policy

Since accidents are always associated with the presence of dangerous flows, achieving a safe environment is a matter of controlling the way in which humans interact with the dangerous flows in the environment. Typically, this is done by eliminating, containing or isolating the dangerous flows, thus protecting the humans from their harmful effects. We call the approach taken to controlling the dangerous flows in an environment the *safety policy*. It may be described by a collection of *safety constraints* (expressed in terms of the system and human interfaces) that, if adhered to within the given medium, will ensure a safe environment (no accidents).

Guidance on safety policy development can be found from many sources (Royal Australian Navy 2006, Department of Primary Industries 2007) . In general, a safety policy will take one of the following approaches to controlling the dangerous flows.

A dangerous flow may be eliminated or constrained below harmful levels. Many standards exist that offer guidance as to safe tolerances for exposure to potentially harmful substances and energies.

A dangerous flow may be isolated from the humans in the environment. This may involve active control, monitoring for human presence and directing dangerous flows away from them; or passive control, building barriers around the dangerous flow or removing it to a remote location.

Finally, the humans in the environment may be made resistant to the dangerous flow by restricting working hours, requiring the use of protective equipment, etc.

Policy development bears some similarity to hazard analysis. It requires an understanding of system interfaces and investigates the potential effects of dangerous flows in the environment. Both involve an outward search through the environment to find dangerous flows that may be influenced by the system S to cause harm. However, policy development is a more contained and positive activity than traditional hazard analysis. In particular, because it focuses on the direct interactions between humans and dangerous flows, it does not require nor promote the kind of analysis of complex causal chains of hazards and co-effectors that requires a deep understanding of the system and environment, both in nominal and failure modes.

Indeed, failure analysis cannot occupy its traditional central place in safety policy development as (quite deliberately) too little is known of the inner workings of the systems operating in the environment.

Only the interface flows of systems (and primarily the dangerous flows) are considered and the focus is on determining how these should be constrained to promote a safe environment. This is not to say that danger mitigation (reacting to policy breaches) and harm minimisation (reacting to accidents) should not feature in a safety policy, only that safety policy development can (and must) be addressed from the very earliest stages of system development.

As system development proceeds, as equipment choices are made and unmade, the collection of dangerous flows may change and perhaps even the medium itself. Any such changes will force a redevelopment of the safety policy. However, at each point, the scope of the safety policy remains the same: it does not creep inexorably into the deepest nooks and crannies of system design as does the traditional failures-oriented hazard analysis process.

In some cases there will be an existing safety policy applying the operation context and analysis may be concentrated on the ways that the new system may perturb the existing policy. For example, the addition of a new radar source may require further constraints on existing radar sources, perhaps reducing their maximum intensity, perhaps restricting their direction of signal.

In any case, the desired endpoint is a convincing, positive argument that the proposed safety policy will ensure a safe environment (when operated in the proposed medium). The purpose of subsequent safety engineering, is to enforce adherence to the chosen safety policy.

8 The Safety Protocol

In developing the safety policy, the focus is on the interaction between systems and humans, determining what constraints systems must satisfy in order to operate safely. The obvious next step is to move our focus onto the interactions between the systems themselves, placing a structure on those interactions that will enable adherence to the chosen safety policy. During this design process, safety constraints may be decomposed and/or strengthened and new flows introduced to serve as communications channels between systems. The eventual aim is to assign to each individual system a collection of safety requirements expressed solely in terms of the given system's inflows and outflows — in such a manner as to ensure that the aggregation of all the system safety requirements implements the safety policy. We call such an assignment of safety functionality across the various systems a *safety protocol*.

The safety protocol constraints on S must be expressed in terms of the various system interfaces so that they can then be adopted as system safety requirements as development moves into the system architecture phase. Safety protocol constraints on systems in the operational context should also act as safety requirements to their respective systems and may be needed as assumptions during safety architecture verification on S. In any case, it seems appropriate to treat them on a par with the constraints on S and express them solely in terms of their system's interface.

The safety protocol should also describe any safety mitigations present in the operational context. Such factors include redundant safety functionality, system isolation, safety monitoring and protective barriers: anything that may impact on the degree of reliance placed on S for ensuring the overall safety the environment.

The development of a safety protocol is a creative process, concentrating on the desired interactions between the systems in the environment. It may include many aspects of traditional hazard analysis, but without the usual focus on failures. It may involve a certain amount of trial and error, proposing safety constraints and challenging them with accident scenarios that show them to be inadequate. Equally, it may involve the adoption of standardised approaches to controlling specific dangerous flows or mathematical calculation of "worst-case" propagation of dangerous flows in the given medium. Many existing standards provide useful guidance on developing hazard control protocols (Royal Australian Navy 2006, Department of Defense 1993, Department of Primary Industries 2007).

In any case the desired endpoint is a convincing argument that the protocol actually implements the safety policy.

9 Methodological Matters

While we have spoken of the safety protocol *approach* in the preceding, what we have described is perhaps better considered a modelling framework from which to hang considerations about the fundamental nature and purpose of system safety engineering. We do not suggest that this framework provides even a significant portion of a viable methodology for safety engineering, but we do believe that it has shown its value in immediately clarifying some of the murky waters surrounding the foundations of system safety. This can be used to assist in evaluating and utilising existing methodologies. We do not go into any details of existing methodologies here, but raise some relevant points in the following.

Recognising the safety constraint as the primary focus of safety engineering effort has the potential to clarify the concept of hazard, as used in existing safety techniques. Current definitions of hazard are unsatisfying in that they conflate the notion of constraint violation with those of dangerous flow and equipment failure. This is particularly problematic when considering (or justifying the exclusion of) accident scenarios involving dangerous flows or equipment failures that are managed by protocol measures in the operating context. Both dangerous flows and equipment failures may legitimately occur within a system operating correctly within its safety constraints – this is after all the purpose of the safety constraints. It is important that practitioners clearly distinguish these three concepts and that safety methodologies should encourage them to do so.

The authors' primary interest in this system safety model lies in its potential to enhance the presentation and evaluation of safety cases. We have identified a simple three-level taxonomy of safety constraints. Policy constraints directly address the safe management of dangerous flows within the operating context. Protocol constraints address the safe interaction of systems in the operating context. Architecture constraints address the safe interaction of components within a system. This separation of concerns clarifies the structure of the safety engineering process and opens the potential for highly formal requirements tracing in support of high assurance safe cases. Again it is likely that practitioners will benefit by clearly distinguishing these three levels of safety constraint and, even if they do not require such structure, safety methodologies should, at least, be able to accommodate it. Moreover, a constraint focus allows safety engineering to be better integrated with the general engineering process, which is also requirements focused.

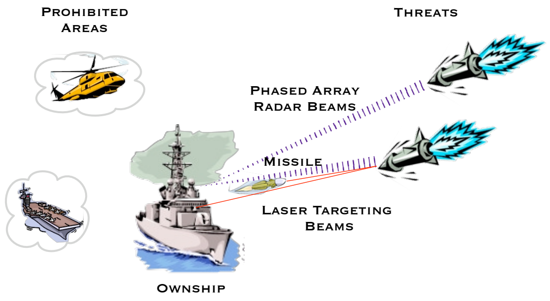

Figure 7: PARTI System Overview

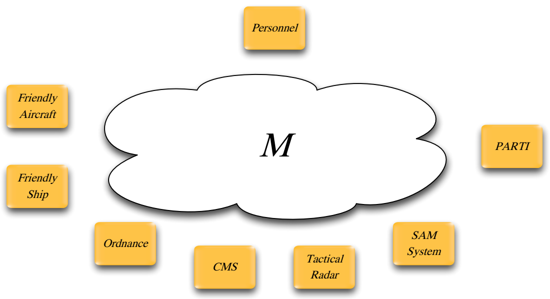

Figure 8: PARTI Environment

10 The PARTI Environment

We illustrate the safety protocol approach using a case study from DEF(AUST)10679 (Department of Defence 2008*a*, Mahony & Cant 2008). All of the information presented here is taken from that case study and the purpose is to contrast safety protocol development with the hazard analysis approach taken there.

The PARTI (Phased Array Radar and Target Illumination) System is a ship-borne Surface to Air Missile (SAM) targeting support system. It uses a Phased Array Radar (PAR) to direct laser illumination of hostile missiles and aircraft. The laser illumination provides targeting information to an existing ownship SAM capability. The main elements of interest in the PARTI and environment are depicted in Figure 7.

Our approach begins by describing the environment in which the PARTI must operate in sufficient detail to allow the development of a safety policy governing the operation of the environment with the PARTI. Ideally there would be an existing fleet-level or theatre-level safety policy which we would be updating to allow the inclusion of one or more PARTI systems on ship platforms. Unfortunately, no such policy is likely to exist and developing such a policy is likely beyond the scope of the PARTI development. Instead we strive to describe only those aspects of the environment, safety policy and safety protocol relevant to the safe operation of the PARTI. Additionally, for the sake of brevity, we adopt a narrow focus on “system safety” issues — ignoring “OHS” issues — that would not be appropriate in a real safety case.

The PARTI system is to be installed on a class of frigate operating in a naval theatre of operations. The elements of safety concern in this environment are as follows.

CMS — The Combat Management System (CMS) provides command and control over all ship-based systems. The CMS has no relevant dangerous flows.

Search Radar — The CMS supports a conventional search radar for maintaining situational awareness and supporting an Identify Friend and Foe (IFF) functionality. The radar emits a HF radio signal.

SAM — The SAM System launches missiles against targets identified by the CMS. The missiles have target illumination home-all-the-way capability that is to be enabled by the PARTI. Missiles also have self-destruct functionality that will activate on order from the SAM system, on loss of target illumination and at mission expiry. The SAM system has no other ability to influence missile flight post-launch. The SAM system can deliver dangerous kinetic and explosive flows through its missiles.

PARTI — The PARTI provides precision tracking and target illumination in support of the SAM system. The PARTI can deliver dangerous HF radiation and laser energies.

Ordnance — Other systems include the harpoon missile, the 5-inch gun, the Nulka Anti Missile Defence system and a torpedo system. These ordnance can deliver dangerous explosive energies.

Friendly Aircraft — Helicopters may land or take-off from the frigate and/or from nearby ships. Other friendly aircraft may also be present. These aircraft can deliver dangerous kinetic, chemical and fire energies.

Friendly Ships — The frigate may be accompanied by other non-hostile surface vessels. These ships can deliver dangerous kinetic, chemical and fire energies.

The structure of the environment is depicted in Figure 8.

The medium M resolves the physical interactions between system outflows to determine the resultant system inflows. For the most part this is a straightforward resolution of positional interactions, in particular determining when humans actually come in contact with dangerous flows present in the environment: is the human present when a collision between aircraft and missile or terrain causes dangerous acceleration; is the human close enough to an explosion to be harmed; is the human in the path of a laser beam. Interactions with the (at least) two radars are also complicated by the need to resolve the superposition of the interacting wave forms.

11 The PARTI Safety Policy

The purpose of the safety policy is to describe safe interaction between the systems and the humans in the environment. The most desirable approach to ensure safe interaction is to restrict the release of energies to safe levels: prevent collisions and dangerous accelerations; prevent fires; etc. However, the PARTI environment contains a number of dangerous flows that might reasonably be termed “mission critical”. The SAMs must fly energetically to their target and explode effectively. The PARTI must emit radar and laser signals at intensities suitable for the purpose of guiding the SAMs to destroy incoming threats. Instead of preventing the release of these energies, our policy is ensure that they are never released in the presence of the humans in the environment.

To achieve this, a region of the environment is set apart for the enacting of PARTI functionality, the SAM system and the PARTI agreeing to operate solely in this space and the humans in the environment agreeing not to enter it. This *tactical region* may vary according to the threat situation but will always exclude inhabited areas of ownship and other friendly surface vessels. The DEF(AUST)10679 case study (Department of Defence 2008*a*), adopts a policy of always setting the tactical region so as to exclude all friendly manned traffic in the environment's airspace. This protects friendly manned aircraft at the possible expense of making it impossible to effectively respond to an incoming threat. Thus, since the safety onus is placed on the PARTI, friendly aircraft and ships effectively have no safety responsibilities related to the PARTI system.

Overall, the safety policy constraints are as follows.

CMS — The tactical region is set and promulgated by the CMS, according to situational awareness and in response to command input. The tactical region must always exclude a suitable buffer zone around all manned friendly aircraft and surface vessels. The tactical region should always include only the essential volume(s) of space required to respond effectively to any identified threat(s).

Search Radar — The search radar shall always emit HF radiation within safety standards set for naval operations.

SAM — The SAM system shall always operate its missiles within the tactical region.

PARTI — The PARTI shall always emit its HF radiation and laser beams only within the tactical region.

Ordnance — All ordnance may only detonate within the tactical region.

12 The PARTI Safety Protocol

We now proceed to consider the potential interactions between systems in the environment so as to develop a protocol for their safe operation in the environment.

An obvious matter of concern in regard of ordnance and friendly vessels lies in the potential for radar and laser beams to damage these systems with consequent loss of safety control and release of dangerous flows. To avoid this threat, it is sufficient to ensure that radar and laser energies are never directed at any of these systems. This is essentially the purpose of the tactical region safety constraints and they serve to protect the systems as well as their human operators.

The primary matter of concern is the PARTI mission of providing target guidance for the SAM system. Since the SAM system has little control over missiles once launched, the PARTI must have primary responsibility in ensuring that missiles fly within the tactical region and therefore do not interfere with friendly traffic. Clearly, these systems must communicate effectively if they are to operate safely and this communication will be enacted through the CMS.

The CMS determines (in response to operator input) when the SAM and PARTI are operational.

The CMS develops situational awareness of the threat environment through an array of sensors, including the search radar and IFF. This situational awareness is transmitted to the PARTI as a list of tracks, some of which are tagged as threats.

The CMS determines (in light of its situational awareness and in response to operator input) the tactical region and factors this information to the PARTI.

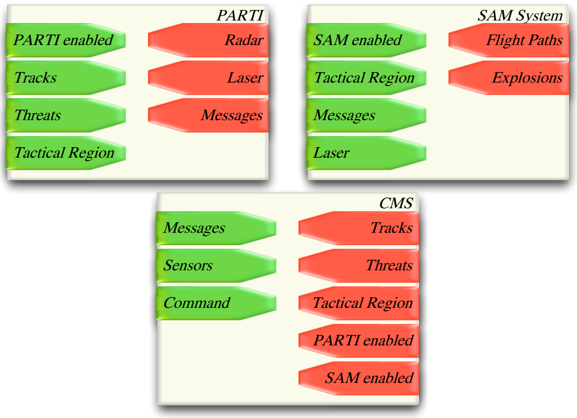

Figure 9: Protocol Interfaces

Based on this information from the CMS, the PARTI uses its PAR to acquire precision tracks on identified threats and then commences illuminating them with its targetting lasers. When target illumination is established it sends a (target) *acquired* message to the SAM system, along with the current position of the threat. The identified threat is then referred as a *target* until a (target) *released* message is sent or it is destroyed.

The SAM system configures a missile to acquire target lock on the identified threat position and launches it. The missile briefly flies a preprogrammed path ρ, within the tactical region, during which it must either attain target lock or self-destruct. Once target lock is attained it flies a line of sight path to the threat.

The PARTI maintains target illumination until the threat is destroyed or it becomes unsafe to maintain illumination. It is safe to maintain illumination provided the laser and the missile have safe *line of sight* to the threat. Line of sight is essentially the straight line path between objects, modulo the missile's flight navigation tolerances. The line of sight is safe provided it is entirely within the tactical region and there is no third object in line of sight. If line of sight becomes unsafe, the PARTI informs the SAM system and ceases illumination. The SAM system then transmits a self-destruct command to the missile, which will self-destruct, either because it has detected the loss of target illumination or because it has received the self-destruct command.

The resulting safety interface to the PARTI safety protocol is shown in Figure 9 (outflows are shown in red and inflows in green).

The protocol described above is summarised by the following safety requirements.

CMS_A – at all times the tracks presented form an accurate model of the objects moving through the environment to within allowed tolerances.

CMS_B – at all times the tactical region plus an allowed tolerance contains no friendly tracks.

SAM_A – if a missile is launched, the SAM system is enabled and there is a valid target.

SAM_B – when a missile is launched, it initially flies along an initial path ρ safely within the tactical region.

SAM_C – if a missile departs from its initial path ρ, it has either acquired target lock or self-destructed.

SAM_D – while a missile has target lock, it navigates a line of sight path to its target to within allowed tolerances.

SAM_E – if an in-flight missile does not acquire target lock promptly or loses target lock or flies beyond its mission deadline or the SAM receives a target release message, the missile self-destructs promptly.

PARTI_A – only tactical regions are irradiated.

PARTI_B – if an object is being illuminated then it is a target.

PARTI_C – each target acquired message gives the correct current position of a threat within the tactical region.

PARTI_D – if an object is a target then it is a threat, the PARTI is enabled and the object is being illuminated.

PARTI_E – if line of sight to an object is not safe then it is not a target.

PARTI_F – if the PARTI is not enabled then it is not radiating or illuminating.

The protocol constraints CMS_A and CMS_B clearly ensure satisfaction of the CMS safety policy constraint. Similarly, the PARTI_A, PARTI_B, PARTI_E and PARTI_F ensure satisfaction of the PARTI policy constraint.

Modulo the determination of correct tolerances, the protocol also ensures satisfaction of the SAM policy constraint. To see this, consider the path of a missile from launch to detonation or self-destruct. SAM_B ensures that some initial segment of the missile's path is safely in the tactical region. Now suppose that the missile has travelled safely in the tactical zone up until some point x on its flight path. Then the missile is safely within some fixed tolerance, say δ, of the tactical region boundary and regardless of the missile's current speed and direction, it will remain within the tactical region until it has moved a distance $\frac{\delta}{2}$ to a point y. Either the threat remains a target while the missiles moves to y or it does not. If the target remains a threat, then by PARTI_E there is safe line of sight to the target at all times and in particular the point y is safely within the tactical region. If the threat ceases to be a target while the missile moves, then by PARTI_B the threat is no longer being illuminated and by PARTI_E a target released message has been sent. Thus, by SAM_E the missile will self-destruct before it can leave the tactical region.

Note that SAM_C and SAM_D are not used in this argument as they are in fact redundant safety functionality. If either protocol constraint is violated, the PARTI will detect loss of line-of-sight, release the target and the missile will self-destruct.

13 Final Remarks

In this paper, we have discussed the traditional hazard-based approach to developing system safety requirements, highlighting a number of its properties that we see as short comings. In its place, we suggest a more contained investigation of the dangerous flows associated with a system and its environment, together with a more direct determination of a policy for safely constraining these dangerous flows. A safety protocol is then developed that describes a particular way of coordinating the interactions between systems in the operational context so as to implement the safety policy and therefore ensure a safe environment.

Why have we used the word "protocol" instead of "requirement"? This is a most interesting question. In the computer science field the term protocol is used to mean "a set of rules governing the exchange or transmission of data between devices". There are many familiar examples: communications protocols, such as TCP/IP, but also security protocols, such as Internet Key Exchange. In the field of safety, the term "protocol" has so far mostly been used in a narrow sense to denote procedural rules designed to promote safety, such as rules for food handling or procedures for operator behaviour in factory operations. We like the term in our situation because we wish to think that averting a dangerous flow is going to involve a kind of "agreement" or "handshake" between the various systems in the operational context. To use another familiar analogy, it is going to be a "contract" that represents agreement between the components. The notion of protocol is broad enough to cover both system design constraints and procedural obligations on humans.

References

Comcare (2007), Identifying hazards in the workplace, Guide booklet, Australian Government, `http://www.comcare.gov.au/`.

Department of Defence (2008*a*), Guidance Material for DEF(AUST)5679/Issue 2, Australian Defence Handbook DEF(AUST)10679/Issue 1, Australian Government.

Department of Defence (2008*b*), Safety Engineering for Defence Systems, Australian Defence Standard DEF(AUST)5679/Issue 2, Australian Government.

Department of Defense (1993), System Safety Program Requirements, Military Standard MIL-STD-882C, United States of America.

Department of Defense (2010), Joint Software Systems Safety Handbook (SSSH), DoD publication, United States of America.

Department of Primary Industries (2007), Guideline for hazardous energy control (isolation or treatment), MDG 40, New South Wales, `http://www.dpi.nsw.gov.au/minerals/safety`.

Haddon-Cave, C. (2009), *The Nimrod Review*, The Stationery Office Limited, UK.

Kelly, T. (2011), 'Safety critical mailing list', `http://www.cs.york.ac.uk/hise/sc_list_arc.php`.

Leveson, N. G. (1995), *Safeware: System Safety And Computers*, Addison-Wesley.

Leveson, N. G. (2011), 'Engineering a safer world', `http://sunnyday.mit.edu/safer-world/index.html`.

Mahony, B. & Cant, A. (2008), The PARTI architecture assurance, *in* 'Proceedings of the 13th Australian Conference on Safety-Related Programmable Systems', Australian Computer Society.

Ministry of Defence (2007), Safety management requirements for defence systems, part 1 requirements, Defence Standard 00-56, British Government.

Royal Australian Navy (2006), Navy Safety Systems Manual, ABR 6303, Australian Government.

Schellhorn, G., Thums, A., Reif, W. & Augsburg, U. (2002), Formal fault tree semantics, *in* 'Proceedings of The Sixth World Conference on Integrated Design & Process Technology'.