CHAPTER 7

An Example in Which System Integrity is Critical

The launching of a US intercontinental ballistic missile (ICBM), or a series of ICBMs, is a complex process, due to the distributed nature of the system and its necessary communications infrastructure, as well as its dependability requirements and the ensuing sociotechnical-algorithmic complexity to assure that dependability. As befits a technology which is critical to the future of humankind, the launch process and its reliability and integrity issues have been discussed extensively in the unclassified literature. A selection follows. Blair conducted a seminal study of the reliability, resiliency and integrity of the entire US launch process and its criteria, as it then was, in 1985 [1]. Mackenzie wrote a prize-winning sociotechnical study of the evolution of ICBM guidance systems [13]. Sagan studied accidents and incidents in US nuclear-weaponry operations [14]. Schlosser investigated in a journalistic style some incidents surrounding ICBM installations in the US MidWest [16]. Recently, concerns about system integrity have been raised in articles by Blair, considering the possibility of unauthorised intrusion into the digital systems of ICBM control [2], and Shatz, who considered the dependability of the human command structure [17].

Rather than describe such a complex system by means of its minutiae, I suggest it is helpful for cybersecurity analysis to decompose it into components and their causal relations.

I will consider here the launch decision and action, and the integrity of these processes. A launch decision by POTUS is communicated to the physical command centres by an Emergency Action Message (EAM) [15, 20]. An EAM is a command sent by the US President (POTUS) to commence system action, including launch,

and is roughly comparable in length with a tweet[1]. An EAM is encrypted and cryptographically authenticated. It is a system requirement that a valid launch EAM results inevitably in a launch.

7.1 Causal Control Flow Diagrams (CCFDs)

A Causal Control Flow Diagram (CCFD) is a diagram which Causalis uses to illustrate the causation inside an engineered system [18], [9, Chapter 5][2]. The notion of "necessary causal factor" (NCF) used in a CCFD is a philosophical-logically rigorous notion developed by the philosopher David Lewis [10] based on his logic of counterfactual assertions [11]. The Lewis notion of NCF was used in the engineering causal-analysis method Why-Because Analysis (WBA) [9, Chapter 1 and Chapter 4], [19]. CCFDs were developed using a similar conceptual apparatus to that of WBA.

CCFDs were developed for use in engineered systems. However, they can be used to analyse sociotechnical systems such as an ICBM launch system with certain semantic adaptations. Sociotechnical systems involve human and human-organisational agents. It is a complex philosophical problem to speak of "cause" when considering human agency. For my purposes here, it suffices to identify the notion of a sociological "cause" of an executed action with a human or organisational intention to execute the action as defined or implied by standard system procedures. (If the action is not executed, it of course cannot have a cause.)

7.2 Functional Integrity and Information Integrity

I recall the definitions of functional integrity and information integrity from Chapter 5, and repeat the CCFD illustrating the definition of information integrity.

- *Functional integrity* is the property of a system or component that its system-relevant behaviour remains the same.

1 A *tweet* is a message sent on the Internet broadcast-messaging service Twitter, and is up to 140 alphabetical characters in length.

2 Also see [8, Chapter 9], where they were called "Causal Influence Diagrams", a name we then discovered was used by many others for various different notions.

where

- *system-relevant behaviour* is behaviour of a system or a component of a system which contributes causally to the fulfilment of some part of the system requirements specification

Information integrity is defined as follows:

- *Information integrity* is the property that the meaning of the information held at any state St of the system Sys is conformant with
  - either the real world (that is, the information corresponding to real-world parameters is veridical) or
  - veridical information held at other states St_1 of the system, transformed by the functionally-correct transformations applied by Sys to St_1 which result in St.

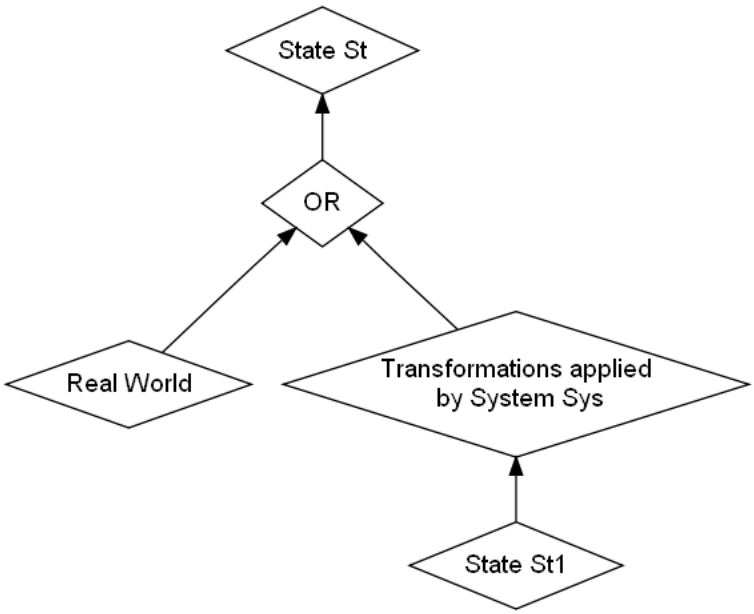

Figure 7.1: Information Integrity Causal Control Flow Diagram

7.3 The Launch Function Analysed

A general not-quite-CCFD of the launch process is shown in Figure 7.2. I write "not-quite" because the diagram is not an actual CCFD. This is because the lower nodes, "causally" feeding in to the launch decision, do not necessarily satisfy the Counterfactual Test[1]: the decider, POTUS, is not in fact bound by procedure to take these factors into account [17]. We might call it a quasi-CCFD.

The quasi-CCFD shows various phenomena such as *Phenomenology* and *Checklist and Procedures* which are causally or quasi-causally input into *Launch Decision and Action*, which then causally results in *EAM commands launch* via intermediate causal apparatus denoted *Syst2*. The EAM issued causally results in *Missile Launch* via intermediate causal apparatus denoted *Syst1*.

There is a "dual phenomenology" used to aid launch decisions. This phenomenology consists of real-time information about possible missile launches from adversaries, and comprises

(a) infrared data coming from satellites, and

(b) dynamic data coming from multiple radar sites.

These two data streams are generally assumed to be independent. The dual phenomenology is intended to be an important causal input – better said, two important causal inputs which should cohere – in a launch decision. Other important causal inputs are

- the applicable checklist, and
- the applicable procedures, and
- possibly other environmental parameters, for example information communicated by military aircraft in flight, which are here not further specified.

All this information is intended to form causal factors for decisions on activation of the system. The information is causally intermediated on its way to the launch decision by systems designated *Syst3*, *Syst4* and *Syst5*. It is intended by the system designers that there is causal influence from these inputs on the decision, but as noted

1 The Counterfactual Test asks: given a decision to launch, would the decision to launch not have been taken had the phenomenology not indicated what it indicated? And the answer is: not necessarily, as Shatz [17] says.

above it is possible that the influence is absent [17]. The arrows are thus not causal per se; they are "desired-to-be-causal" in some sense, but there is no mechanism to ensure that these factors are indeed causal – – and they thus do not satisfy the Counterfactual Test. They are shown as dashed in Figure 7.2 to distinguish them from those nodes whose causal relations are established through the Counterfactual Test.

It may well be that the dashed arrows between *(6) Phenomenology* and *Syst4* are in fact causal rather than just desired-to-be-causal; similarly the arrows between *(5) Checklist and Procedures* and *Syst5*, and *(4) other Environmental situations* and *Syst3*. This can be established (or contraindicated) by more detailed inquiry into the nature of these subsystems and their causal connections.

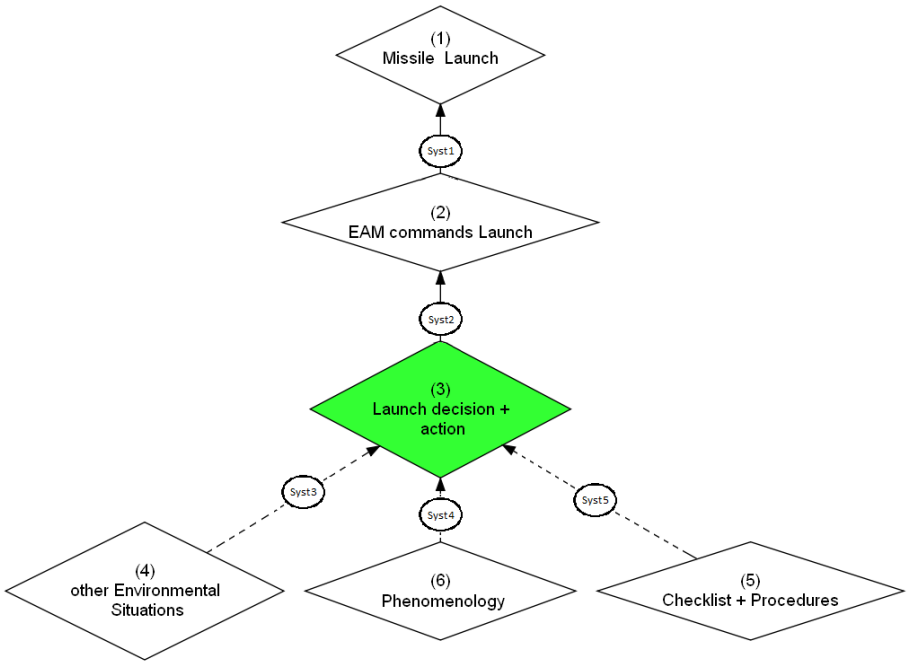

Figure 7.2: A General Quasi-CCFD of a Launch Action of a US ICBM

In between each of the major labelled nodes are the nodes labelled *Syst1*, ... *Syst5*, which represent causally-intermediary system apparatus. To illustrate, let us consider *Syst1*, the causal system intermediating between the production of an EAM commanding a launch and the actual launch of a missile. When a launch-EAM is

produced,

- communications systems transmit this EAM to the the site of a missile to be launched;
- at the launch site, the authenticity of the message is validated by a sociotechnical subsystem; and,
- if the authentication validates, action to launch the missile is then taken by that sociotechnical subsystem.

These are three separate system functions which serially combine to connect causally the production of an EAM commanding a missile launch with an actual missile launch. This subsystem, a combination of geographically-separated communications and the on-site sociotechnical subsystem, is denoted by *Syst1* in Figure 7.2.

For system-analytical reasons, one may wish to decompose *Syst1* into two subsystems, the second of which itself decomposes serially into two components:

- the communications subsystem *Syst1.1* conveys a POTUS launch decision encoded in an EAM to the launch site, and
- the on-site sociotechnical subsystem *Syst1.2* validates the EAM and acts to launch. *Syst1.2* itself decomposes into the serially-executed subsystems
  - on-launch-site reception, decoding and validation of the authenticity of the EAM; followed by
  - if a launch-EAM validates, action to launch the missile.

Such a decomposition helps to localise the various vulnerabilities which may be manifest through the impact of emerging technologies, as follows:

- It has been suggested that the on-site sociotechnical subsystem *Syst1.2* is fairly robust against cybersecurity threats posed by emerging technologies [3][1]. This is largely because the procedures are human, static, and validation is largely physical, not digital-electronic.
- On general grounds, systems scientists may well be concerned about the cybersecurity of the communications subsystem *Syst1.1* and the possible means of inhibiting or "spoofing" an EAM. (Inhibition was considered in depth in [1].)

The caveat "fairly robust" for the cybersecurity of *Syst1.2* is apt. For example, a *Syst1.2*

1 The referenced Workshop was conducted under the Chatham House Rule [4].

common-cause electronic fault has indeed occurred on-site, reported by Blair [2][1]. In this incident, using the terminology just elaborated, the fault was present, but a failure was only potential. The fault would have inhibited a launch on a launch-EAM, had such an EAM been issued. Since no such EAM was issued[2], the fault did not manifest as behaviour.

Applying the terminology of Section 7.2 to the incident recounted by Blair, the on-site sociotechnical subsystem *Syst1.2* did not retain its functional integrity. The cause of the loss of functional integrity was an implementation error, a faulty circuit board. There are many ways of rendering circuit boards faulty. Some of them are spontaneous. Some are inadvertent design, manufacturing, installation or maintenance errors. We may assume one of these occurred in the incident recounted by Blair. But such actions which may be inadvertent can also be deliberate, initiated by a malfeasant intervenor. Then they become cybersecurity issues. Let us consider them individually.

- Circuit-board or chip design is a process usually involving a team. Design errors, inadvertent or deliberate, may be avoided by
  - keeping the design of the chip simple, and
  - using formal methods to prove mathematically that the design fulfils the functional requirements.

  Any attempt to introduce a deliberate design error must somehow circumvent the formal verification – the formal verification must come up with the result that the design fulfils its requirements, although in fact the design does not do so. Introducing an error, but allowing such "proof" to be falsely generated, is a situation which can be controlled for using well-exercised human-organisational techniques: separate, independent verification teams and processes, for ex-

1 In general engineering terminology, a fault is a system state which would causally engender erroneous behaviour. The erroneous behaviour itself is called a failure. The definition of failure in the basic (non-military) electronic/programmable electronic functional-safety standard IEC 61508-4:2010 [10] varies from this, though, as follows. IEC 61508-4:2010 Subclause 3.6.1 *fault*: abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function. IEC 61508-4:2010 Subclause 3.6.4 *failure*: termination of the ability of a functional unit to provide a required function or operation of a functional unit in any way other than as required.

2 And we may well hope that no such EAM ever is issued!

ample. The human-organisational problem of infiltrating each independent verification team and successfully causing a spurious verification in each team may well be a much harder problem than deliberately introducing a certain kind of error into the design.

- Introducing deliberate errors during manufacture of a chip would similarly be fraught with organisational problems. If errors are introduced randomly, then it is very likely that such errors would be caught during chip validation – very likely, but not impossible, for it is not possible to test the behaviour of a chip on all possible inputs. However, chip manufacturers have good records on validation.
- An error during installation of a circuit board could be
  - physical damage to the board resulting in partially different functionality [5], or
  - connecting the board incorrectly to peripherals.

  Such phenomena are well-controlled through independent validation processes, as with design, for they must guard against inadvertent errors.
- An error in a circuit board introduced during maintenance, whether inadvertent or deliberate, is controlled for by similar procedures to those for installation.

In summary, the processes which control for inadvertent error in design, manufacture, installation or maintenance of a circuit board arguably suffice to control for deliberate fault introduction. It seems appropriate to suppose that the mechanisms already in place to control for faults in the circuit-board lifecycle suffice also to control for deliberate as well as inadvertent faults, maybe with some strengthening. In particular, it seems as if there is limited scope for achieving such results using so-called "new technology", such as deep-learning neural-network (DLNN) technology[1].

This discussion goes some way towards substantiating the suggestion that *Syst1.2* is "fairly robust" in the face of cyberattack. Such an attack would have to focus on specific phases or components of *Syst1.2*, and, as in the case of a circuit board exhibiting variant functionality, a strengthening of the controls already in place in

1 Now popularly called "artificial intelligence", appropriating this half-century-old concept for a subpart of its subject matter, or "algorithms", appropriating an even older and more venerable concept from computer science. We may welcome technical progress and at the same time regret venerable technical terms losing their specificity.

those phases could well suffice to inhibit the introduction of deliberate faults as well as the inadvertent faults which they already largely suffice to inhibit.

Other parts of the launch-decision-and-action system appear to be less robust against "new technology" cyberattack using DLNN technology. There are two broad ways a launch decision could be inappropriate:

- a decision is made not to launch during actual attack. There may be two reasons for this.
  - One reason is discussed in [1], that, given an attack is in progress, a retaliation would not lead to the best possible outcome. The reasoning involved in determining the best possible outcome may itself be dependent on information supplied externally to the decision-maker, say through *Syst3* in transforming the information from *(4) other Environmental Situations*. Such reasoning may be susceptible to loss of information integrity in *Syst3* as well as loss of both functional and information integrity in *(4) other Environmental Situations*. Mutatis mutandis for *Phenomenology/Syst4*. However, since *Checklist + Procedures/Syst5* is largely static, it is harder to see how functional and information integrity could here be lost. It should also be noted that a decision not to launch can be made appropriately, based on information that has retained its integrity.
  - Another reason is that the attack is not recognised as an attack. This would involve loss of integrity (functional and/or informational) in *Syst4*. Another possibility is coordinated loss of integrity in the dual systems comprising *(6) Phenomenology*. This would be a case of common-cause failure, as in Figure 7.3. However, during the more than half-century these systems have been in place, one can well imagine that the possibilities for common-cause failure of both parts of the dual phenomenology have been well-studied and appropriate prophylactic measures introduced. There may be good grounds for constantly reviewing the independence of both channels of the dual phenomenology, but these grounds are independent of how a common-cause failure might occur. If common-cause failures are indeed appropriately inhibited, "new technology" cyberattacks on the phenomenological channels will by hypothesis not succeed in causing a fail-negative. The major worry here is surely a loss of integrity in *Syst4* through cyberattack.

- a decision is made to launch based on "recognition" of an attack that is in fact not taking place. Assuming *(6) Phenomenology* is causal in the decision, a false "recognition" of an phantom attack would involve compromising the information integrity of both channels of the dual phenomenology in a coordinated fashion. Such coordinated compromises of information integrity in both channels, which are based on independent physics and technology, is widely regarded as unrealistic. However, a loss of integrity in *Syst4*, the causal intermediary system between the facts recognised by the phenomenology and the contribution to a decision, could theoretically result in faulty "recognition".

Examples of phenomenological input misleading the military to perceive an impending attack have occurred in both the US command [14, Chapter 5] and in the Russian command [6, 21]. (As far as I know, there is no incident yet, thankfully, in which valid warning information has been inhibited.) The situation in such a common–cause failure of information integrity is indicated by the CCFD in Figure 7.3. Note that such a common cause would have to affect the subsystems *Syst3* and *Syst4* in Figure 7.2 in a coordinated fashion. For the reasons of independence adduced above, this would be a very tall order.

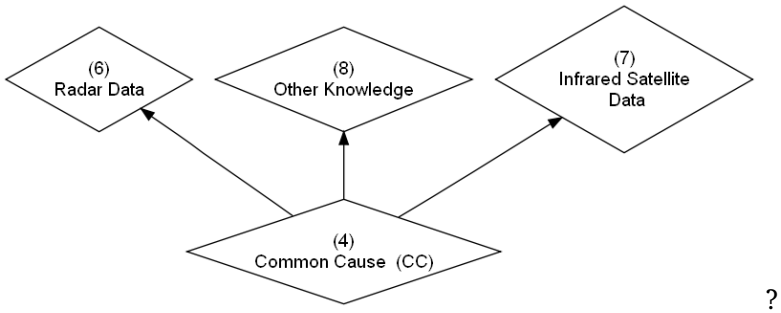

Figure 7.3: CCFD illustrating a common cause

A final example of the analytical localisation of loss of integrity comes from considering in more detail the supporting information flow to a launch decision, as in Figure 7.4. I have already noted that the connections between the informational factors, *(2)*, *(3)*, *(4)* and *(1) Launch Decision* is that of desired-to-be-causal rather than truly causal as determined by the Counterfactual Test, hence this diagram is a Quasi-CCFD rather than a CCFD. Although dashed lines are not used here, the connections are

causal-or-desired-causal and not causal simpliciter.

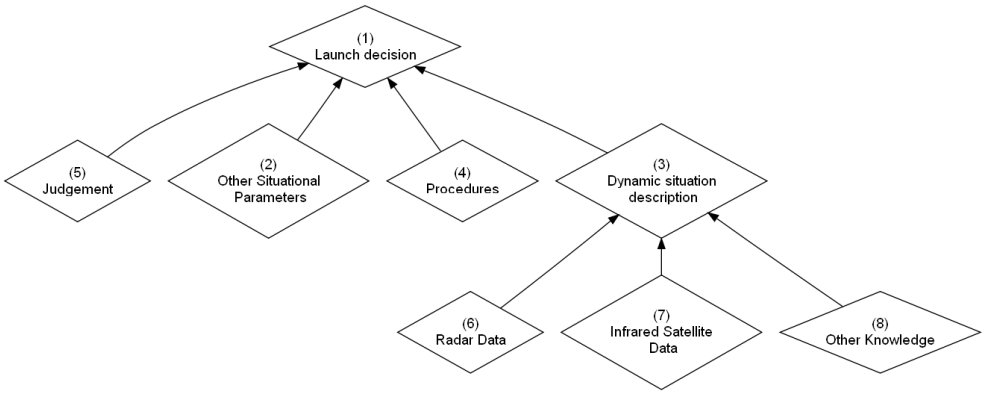

Figure 7.4: A Quasi-CCFD of Information Supporting a Launch Decision

I have observed that the launch decision is not required to take available information into account, but it is reasonable to suppose, indeed expected and anticipated of the decision maker POTUS, that such information from the dual-phenomenological systems, as well as applicable defined procedures, will play a causal role, along with the decision-maker's judgement, in a launch decision. Let us consider these factors one by one.

- Figure 7.4 shows the dual phenomenology, along with other information (there may be other observers of a potential launch in immediate contact with the Situation Room, for example, reconnaissance aircraft gathering telemetry; such observations are collected under *(8) Other Knowledge*) assembled under the rubric of *(3) Dynamic situation description*. It may be theoretically possible for deliberate intervention to cause a failure of information integrity in the dynamic situation description, but only under the condition that *Syst3* and *Syst4* are compromised in coordinated fashion. It should be possible to inhibit such a coordinated compromise by ensuring that *Syst3* is sufficiently independent of *Syst4*, both physically (sensors and communications) and in terms of personnel, and that common causes of loss of integrity of both *Syst3* and *Syst4* are hard or impossible to devise. Such measures would ensure the information integrity of the *(3) Dynamic situation description*.
- The *(5) Judgement* of the decision maker is presumably not influenced by any

emerging technology.

- *(4) Procedures* are defined largely statically, as well as independently of the technologies used to implement them. What is required here is to ensure the functional integrity of those procedures, in particular under technology change. This is a matter of defining the functional behaviour of each subsystem, and ensuring under technology change that this functional behaviour is invariant. In other words, ensuring the functional integrity of the procedures. It is surely relatively easy to devise ways of doing this which are not prone to cyberattack.
- *(2) Other Situational Parameters* is the factor potentially most in need of care and attention. One could envisage new technology – big-data analytics, say; DLNN technology applied to varied presumed-independent sources of data not derived from the traditional sensing technology involved in the dual phenomenology – being used to attempt to enhance the information from the dual phenomenology. If the dual phenomenology maintains information integrity, then such systems are superfluous. So one way of reducing the risk of vulnerabilities in new technology is to enhance the assurance of the information integrity of the dual phenomenology. This is maybe difficult, but surely desirable in any case.

7.4 Summary

I have briefly illustrated the use of CCFDs and Quasi-CCFDs to describe the causal and desired-causal flows of information and control through the ICBM launch system. Although the (Quasi-)CCFDs were general, the integrity properties of specific subsystems and their effects on the integrity of the overall system could nevertheless be considered at this level of granularity. We might call this process "*decomposing integrity requirements*". The notion of integrity used was that from Chapter 5, because existing conceptions did not appear adequate. Possibilities for loss of integrity, intentionally or inadvertently, can be enumerated using the notion of Chapter 5 more finely, on finer-grained (quasi-)CCFDs derived from more detailed system description.

7.5 Further Work

Further work would derive more detailed CCFDs/Quasi-CCFDs of the various subsystems, in particular *Syst1–Syst5*, and perform a threat analysis, in particular concerning

the possibilities of CCF given emerging technology such as the use of deep-learning neural networks and the building-block style of composing malware to exploit known vulnerabilities in COTS infrastructure.

Bibliography

[1] Bruce G. Blair, *Strategic Command and Control: Redefining the Nuclear Threat*, Brookings Institution, 1985.

[2] Bruce G. Blair, *Why Our Nuclear Weapons Can Be Hacked*, New York Times, March 14, 2017. Available at https://www.nytimes.com/2017/03/14/opinion/why-our-nuclear-weapons-can-be-hacked.html, accessed 2017-11-19.

[3] Chatham House and the Stanley Foundation, *Workshop on Mapping the Relative Risks Emerging Technologies Pose to Nuclear Weapons Systems*, Chatham House, July 18-19, 2017 .

[4] Chatham House, The Chatham House Rule, no date. Available at https://www.chathamhouse.org/about/chatham-house-rule, accessed 2017-11-19.

[5] Kevin R. Driscoll, *Murphy Was An Optimist*, ongoing lecture/seminar on the occurrence of Byzantine faults in electronic/programmable electronic equipment. Version 19 is available at https://rvs-bi.de/publications/DriscollMurphyv19.pdf, accessed 2017-11-30.

[6] The Economist, *Obituary: Stanislav Petrov was declared to have died on September 18th*, September 30th, 2017. Available at https://www.economist.com/news/obituary/21729727-man-who-saved-world-was-77-obituary-stanislav-petrov-was-declared-have-died, accessed 2017-11-19.

[7] International Electrotechnical Commission, *IEC 61508-3, Functional safety of electrical/electronic/programmable electronic safety-related systems Part 4 – Definitions and abbreviations*, 2nd Edition, 2010.

[8] Peter Bernard Ladkin, *Causal System Analysis*, on-line textbook, RVS Group, 2001. Available at https://rvs-bi.de/publications/books/CausalSystemAnalysis/index.html, accessed 2017-11-19.

[9] Peter Bernard Ladkin, *Digital System Safety: Mostly Qualitative Aspects*, preprint of textbook, 2017. Available from the author upon request.

[10] David Lewis, *Causation*, Journal of Philosophy 70:556-67, 1973. Reprinted in [12].

[11] David Lewis, *Counterfactuals*, Basil Blackwell Ltd, 1973, reissued 2001.

[12] David Lewis, *Philosophical Papers, Volume II*, Oxford University Press, 1986.

[13] Donald Mackenzie, *Inventing Accuracy: A Historical Sociology of Nuclear Missile Guidance*, MIT Press, 1990.

[14] Scott D. Sagan, *The Limits of Safety*, Princeton University Press, 1993.

[15] Eric Schaum and Marcel H., *EAMs and HF-GCS*, http://www.numbers-stations.com/media/articles/EAMs.pdf, June 1st, 2016. Accessed 2017-11-19.

[16] Eric Schlosser, *Command and Control*, Penguin Books, 2013.

[17] Adam Shatz, *The President and the Bomb*, London Review of Books 39(22):3–6, 16 November 2017. Available at https://www.lrb.co.uk/v39/n22/adam-shatz/the-president-and-the-bomb, accessed 2017-11-19.

[18] Bernd Sieker, *Examples of Reverse Engineering*, Causalis Limited, 2012. Available at https://causalis.com/90-publications/99-downloads/ReverseEngineeringExamples.pdf, accessed 2017-11-19.

[19] RVS Group, *The Why-Because Analysis Home Page*, no date. Available at https://rvs-bi.de/research/WBA/, accessed 2017-11-19.

[20] Wikipedia, *Emergency Action Message*, no date. Available at https://en.wikipedia.org/wiki/Emergency_Action_Message, accessed 2017-11-19.

[21] Wikipedia, *1983 Soviet nuclear false alarm incident*, no date. Available at https://en.wikipedia.org/wiki/1983_Soviet_nuclear_false_alarm_incident, accessed 2017-11-19.