

Leveson's "Safer World" Interlock Example with OHA

Jan Sanders
Uni-Bielefeld-CITEC and Causalis Limited

22 October 2010

Preliminary Version 2 in Note Form

On pages 181-2 and Figure 8.1 of the draft of her new book, *Engineering a Safer World*, Nancy Leveson introduces an example of an interlock system, which she proposes to analyse for hazards. Daniel Jackson has used this example in <http://people.csail.mit.edu/dnj/tmp/interlock-example.pdf> for which he also provides a working in Alloy at <http://people.csail.mit.edu/dnj/tmp/alloy-model.zip> to illustrate a point he wishes to make concerning hazard analysis for software, namely that the hazard analysis and safety requirements derivation can be performed "up front", and does not need to be carried through the development of the software. We rework the example using Ontological Hazard Analysis (OHA).

We abide by IEC 61508-conformant usage of the terms "requirement" and "specification". We anticipate that this use of these terms can be reconciled with Michael Jackson's more precise use of these terms.

We begin with an informal circumscription of the domain, as would be usual in a client interview. It is important to elicit the client's view of what the system should do, and then perform an analysis, using logic, of this view, amongst other things in order to elucidate possible conflicts (which would appear as contradictions in a formal analysis) and resolve them.

Informal Domain Description (IDD)

There is a power source.

There is at least one human operator.

If an operator touches the power source when it is live/online then this operator will suffer harm.

The power source does not inflict harm, when touched, if it is offline.

During maintenance, it is necessary for an operator to touch the power source.

It is not necessary that the power source be online during maintenance.

First Observations During a Hypothetical Client Interview

We offer here our idea of the kinds of issues that arise during client interview on requirements, given the above client presentation of the safety concerns.

Question: *do other humans come into contact with this equipment other than operators?* Answer: *no.*

Question: *are there procedures in place which ensure this?* Answer: *yes (with appropriate citation) and we are assured of this.* (Note: one might be sceptical of this answer, but separation of concerns

is a legitimate position.)

Question: *in what other ways could the operator suffer harm while operating this equipment, other than by touching a live power source?* Answer: *we are not aware of any.*

Question: *in what other ways could the operator come into contact with a live power source other than during maintenance activities?* Answer: *no other ways.* (Note: one might be sceptical of this answer, but separation of concerns is a legitimate position, and it is open to the operator to train and expressly forbid proximity except during designated maintenance operations, in the same way in which ground personnel are prohibited from approaching aircraft on the ground when engines are not shut down.)

Observations Following from the Interview

Given the client answers to the questions above, one could conclude the following:

If maintenance is never necessary, no harm can ensue through the mechanisms mentioned in the system description.

It follows that *a safety requirement upon these mechanisms is only necessary if maintenance is necessary.*

Note: this is a «closed-world»-type condition. It is probably good practice for a safety analyst to maintain a sceptical attitude towards closed-world conditions. For example, what if the operator violates procedures? (Which operators often do, indeed which may be expected: see Jens Rasmussen's observations about work practices which he called "*migration to the boundary*", and their contribution to the accidents he has surveyed.)

Second, although informal reasoning about system requirements and safety requirements may seem easy and legitimate, formal checks on the reasoning are worthwhile, as Daniel Jackson emphasises. Our original formulation of one observation was that *If maintenance is never necessary, no harm can ensue*. Michael Jackson pointed out that this cannot be right: harm may always ensue through other mechanisms not (yet) handled. A formal reasoning check, even a relatively trivial one, could have been expected to have caught this. One might well consider formal checks on reasoning, even on simple reasoning, to be a matter of professional *duty of care* for a safety analyst.

Further Client Contribution

Question: *given that, as indicated here, harm can only ensue during maintenance, is it possible to forego maintenance?* Answer: *no; maintenance is from time to time necessary.*

Axiomatic Safety Requirement

Harm shall not occur.

For justification of this axiomatic safety requirement, if it needs justification, see IEC 61508.

Informal Safety Requirement Derived from Client Interview

It follows from the above that, in order that harm shall not be inflicted through the mechanisms considered, the power source must be offline when the operator touches it. Which, combined with the axiomatic safety requirement, leads to the safety requirement:

The power source must be offline when the operator touches it.

Given that the client maintains that there are (unspecified) mechanisms in place to ensure that the operator is only in the proximity of the power supply during designated periods of maintenance, we could also consider the safety requirement:

The power source must be offline during designated periods of maintenance.

This is a stronger requirement than the previous, as follows. It follows from what has been discussed, if the answers may be taken to be rigorous, that the operator can only touch the power source during periods of designated maintenance. If the power source is offline during designated maintenance, and the operator may only then be in proximity, it follows that the operator can only touch the power source when it is offline, and the requirement as formulated is thereby fulfilled.

All the above follows, we hope obviously, in logic (simple propositional logic, not alethic or deontic logic, if appropriately formulated). We leave the formulation as an exercise for the reader. This is a routine example of informal-requirements analysis based on trivial formality. However, it is based on taking client answers to safety questions literally and rigorously, and this is not necessarily good practice for a safety analyst, because we know that actual day-to-day practice can deviate (as in "migration to the boundary").

OHA proceeds differently. The informal discussion is used as a guide to the initial (Level 0) vocabulary with which the OHA starts. Generally, observations in the client interviews may be used as guides for the line of development of the OHA Level 0, but the safety requirements specification at Level 0 proceeds purely from formal considerations while performing the OHA, and does not logically depend on the analysis of suggestions in the client interview.

Ontological Hazard Analysis of the Interlock Example

We proceed with the OHA at Level 0. We propose that this analysis is more intuitive than Daniel Jackson's analysis, with fewer "Eureka" steps. However, it presumes a certain level of skill at "bookkeeping", in this case in logic. Here, the Jackson analysis is superior because it has been checked using the logical toolset Alloy, whereas ours remains formalised "by hand".

We have found while teaching hazard analysis that logical "bookkeeping" is the single most valuable skill which students can acquire, and indeed communicating this skill is a major component of successfully training hazard-analysts. However, this skill is often assumed to be present, although it may not be. We show, *inter alia*, that some properties intuited by Daniel Jackson and Leveson are explicitly derived during OHA.

We make use in this analysis of *Safety Assumptions*. These are explicit assumptions made about the domain, which may be assumed (as a conjunct in the antecedent) while proving the Safety Requirements during refinement. Let the Level- n Safety Requirements, k of them, be $SRn.1 \dots SRn.k$. Let the (translation of) the Level-0 Safety Requirement (in Level n , using Meaning Postulates, if need be) be SR . Suppose SA is a Safety Assumption. Then the *Refinement Safety Proof Obligation* (RSPO) at Level n is

$$IF SRn.1 \ \&\& \ \dots \ \&\& \ SRn.k \ \&\& \ SA \ THEN \ SR$$

It is most often the case that SR will be expressed as the conjunction of the (translation under the Meaning Postulates of the) safety requirements at the previous level, Level $(n-1)$, say there are j of

them, thus

$$IF SR_{n.1} \ \&\& \dots \ \&\& SR_{n.k} \ \&\& SA \ THEN \ SR_{(n-1).1} \ \&\& \dots \ \&\& SR_{(n-1).j}$$

or, equivalently,

$$\begin{aligned} &IF SR_{n.1} \ \&\& \dots \ \&\& SR_{n.k} \ \&\& SA \ THEN \ SR_{(n-1).1} \\ &\quad \&\& \\ &IF SR_{n.1} \ \&\& \dots \ \&\& SR_{n.k} \ \&\& SA \ THEN \ SR_{(n-1).2} \\ &\quad \&\& \\ &\quad \dots\dots \\ &\quad \&\& \\ &IF SR_{n.1} \ \&\& \dots \ \&\& SR_{n.k} \ \&\& SA \ THEN \ SR_{(n-1).j} \end{aligned}$$

The point of the RSPOs is as follows. It follows by simple propositional logic that if all RSPOs at Levels 0 to n hold, the Level 0 Safety Requirement, which is complete, as we shall see, holds at the Level n refinement step.

It is a requirement of OHA that the Safety Assumption SA is *discharged* at some point. *Discharged* means that:

EITHER
SA is proved by the Level k SRs for some k
OR
A risk analysis must be undertaken for the condition NOT SA, and the risk must be explicitly deemed acceptable

The point of the requirement that any SA be discharged is as follows. We have seen that the notional client has made some assumptions with regard to the operation of his/her system, of which we may be sceptical. We are in effect doing the same if we make an SA. We must allay our professional scepticism in one of two ways: either by formally proving that our safety assumption is no assumption, but is in fact derivable at some level during refinement, or by permitting it to continue as an assumption of the analysis, in which case we have only safety-analysed the cases in which the assumption holds; we have not yet analysed the cases in which the safety assumption does not hold. We must therefore perform this risk analysis for the cases in which the safety assumption does not hold.

We now proceed with the OHA.

OHA Level 0 Ontology

The first step in an OHA is fixing the very-high-level ontology. The client's expression of the system is used here as a guide; however, the ontology is derived by the OH-Analyst using his/her judgement. The purpose, as always, is to keep the Level-0 ontology as sparse as possible, while allowing at least one safety issue to be expressed. The language is preferably that of quantifier-free predicate logic. It should be borne in mind that a major purpose of the Level-0 development is to formulate *and prove complete* a safety requirement in the high-level Level-0 vocabulary.

Objects:

Operator

PowerSource

Relations and Properties, with their negations after the symbol "/" (we believe this convention enhances readability over using a generic NOT statement).

PowerOn(PowerSource) / PowerOff(PowerSource)

Touch(Operator, PowerSource)

Harmed(Operator) / Unharmed(Operator)

Formal Level 0 Safety Requirements:

S0) Unharmed(Operator)

Justification: this safety requirement is axiomatic.

Level 0 Safety Assumption:

During the client interview, we have seen the circumstance envisaged by the client in which harm could occur:

SA0) IF Touch(Operator, PowerSource) && PowerOn(PowerSource) THEN Harmed(Operator)

If the operator is to remain unharmed, the power to the power source must be off when the power source is touched by the operator. This condition is the logical contraposition of SA0, therefore logically equivalent. We may thus take SA0 as an SA. The justification for this SA would be that it is the formal translation of an indubitable assertion from the informal analysis.

We will shortly see, though, through formal analysis, that SA0 is not strong enough. One advantage of OHA over an informal analysis, such as occurred in the hypothetical client interview above, is that such weaknesses are not overlooked.

Level 0 Logical Analysis for Completeness of Safety Requirements

There are precisely three Boolean variables generated by the basic vocabulary:

PowerOn(PowerSource), abbreviated POn

Touch(Operator, Power Source), abbreviated Touch

Harmed(Operator), abbreviated H

In addition, we have *Unharmed(Operator)* IFF NOT H; and *PowerOff(Source)* IFF NOT POn, but these are purely linguistic extensions which do not affect the logic.

Three Boolean variables yield 8 non-equivalent Boolean combinations of the variables, as follows. These are of course simply lines of the truth table (we use the usual convention that "True" is 1; "False" is 0):

POn	Touch	H
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

The Safety Assumption SA0 yields that the seventh line cannot be a reachable state. However, there are three more lines in which H is 1! The complete safety requirement consists of ruling these states out (as well as including the negation of the domain assumption). The states are

(IF NOT POn && NOT Touch THEN NOT H)

(IF NOT POn && Touch THEN NOT H)

(IF POn && NOT Touch THEN NOT H)

The conjunction of these three reduces to

IF (NOT POn || NOT Touch) THEN NOT H

The question is whether this must be a safety requirement. One cannot rule out as a practical matter that the operator remains unharmed if heshe does not Touch the PowerSource. Maybe heshe has a pacemaker, which is adversely affected by the strong electric field surrounding the PowerSource when it is on, and even though heshe is not touching the PowerSource, suffers thereby cardiac arhythmia. This cannot reasonably be a design requirement of the system at Level 0, for Level 0 does not have the vocabulary to express it. It could well be a hazard identified "lower down" in the system development; indeed, we hope it would be so identified. To us, this shows the necessity of continuing to perform hazard analysis during refinement.

So it is possible for the operator to become harmed by the system, but not in vocabulary which we can express at this stage. It follows that the safety condition above is properly considered a Safety Assumption, because it must be carried through the analysis and discharged when the appropriate vocabulary becomes available. We must modify SA0 to

SA0.1)

(IF Touch && POn THEN H) && (IF (NOT POn || NOT Touch) THEN NOT H)

Completeness of the Safety Analysis at Level 0

As indicated above, this is an important step in the Level 0 analysis.

We have enumerated all the non-equivalent assertions that may be made at Level 0 using the vocabulary of Level 0. We have exhaustively ruled out the "unsafe" states, by means this time of a

domain assumption SA0.1. This analysis is obviously complete relative to the Level 0 vocabulary.

Our Level-0 analysis is hereby finished. We proceed to the Level 1 refinement.

OHA Level 1

At Level 1, we attempt to convert the Level 0 safety requirement, which is general (indeed, axiomatic!) into requirements which are connected with the system components and operation as already conceived, that is, with the power source, the operator, and the action of touching. We don't attempt to do more, in line with the OHA practice of making small refinement steps.

Objects, Relations, Properties

As in Level 0.

Level 1 Safety Requirements

S1) PowerOn(PowerSource) && NOT Touch(Operator, PowerSource)

S2) PowerOff(PowerSource) && (Touch(Operator, PowerSource) || NOT Touch(Operator, PowerSource))

Level 1 Proof Obligations

It needs to be checked here that *IF S1 && S2 && SA0.1 THEN S0*. We leave this as an trivial exercise.

Refinement Level 2

The intuitive conception of the safety requirement is that, all other things at this level of expression being equal, the power source must not be touchable by the operator in the online mode. One way of achieving that is to install a safety barrier. The barrier must be removable, so that it may be removed during maintenance. Level 2 introduces a refinement corresponding to this design choice.

Additional Object:

SafetyBarrier

Additional Relations:

Touchable(Operator, PowerSource)

Open(SafetyBarrier) / Close(SafetyBarrier)

Safety Requirements for installing the Safety Barrier

S3) Open(SafetyBarrier) IFF Touchable(Operator, PowerSource)

since we want

S4) Touchable(Operator, PowerSource) IFF PowerOff(PowerSource)

S5) IF Open(SafetyBarrier) THEN PowerOff(PowerSource)

where

(Partial) Meaning Postulate

MP1) IF Touch(Operator, PowerSource) THEN Touchable(Operator, PowerSource)

Level 2 Proof Obligations

It must be shown that *IF S3 && S4 && S5 && SA0.1 && MP1 THEN S0*.

We leave this as an exercise for the reader. Note that S3-S5 are essentially meaning postulates – they postulate truth-equivalence between differently-typed terms.

We note further that the proof may proceed by showing that the antecedent implies *S1*, and separately that the antecedent implies *S2*, and using transitivity of implication and the *Level 1 Proof Obligation*. We leave it as an exercise.

Refinement Level 3

We introduce the Interlock at this level. Again, this is a design choice. It is attached to the safety barrier and shuts the power source off if the barrier is opened.

Level 3 Additional Object:

Interlock

Level 3 Additional Relations and Properties:

Locked(Interlock) / Unlocked(Interlock)

Level 3 Additional Safety Requirements:

we need the Interlock to comply with:

S6) IF Open(SafetyBarrier) THEN Unlocked(Interlock)

S7) IF Unlocked(Interlock) THEN PowerOff(PowerSource)

Note that it is OK for the Interlock to be locked and power source to be off. Also the safety barrier may be closed and the interlock open/power source off. Neither violates the safety requirements.

Level 3 Proof Obligations:

It must be shown that *IF S6 && S7 && SA0.1 && MP1 THEN S0*.

Note this proof can proceed, as already above, by transitivity via the safety requirements at preceding levels.

Refinement Level 4

Let's say that the power source does not immediately become harmless once it is powered off. There is a transition from completely harmful to completely harmless. The transition is governed by physical processes and is deterministic.

To accommodate this latency, we need to refine the expression of power supply modes.

We supplement and substitute:

PowerOn(PowerSource) / PowerOff(PowerSource)

with a "transitional" element:

PowerOn(PowerSource) / PowerTransition(PowerSource) / PowerOff(PowerSource)

This changes the negation of the power-source properties.

We consider the transition state to be harmful and thereby need to rewrite one of our safety requirements.

S1) PowerOn(PowerSource) && NOT Touch(Operator, PowerSource)

will be superseded by

S 1.1) NOT PowerOff(PowerSource) && NOT Touch(Operator, PowerSource)

We omit stating (and demonstrating) the proof obligations from now on. We exhibit merely the way in which we expect the refinement to proceed. It may be that some logical catch turns up in checking the proof obligations, but we assume here that it will be routine to handle it.

Refinement Level 5

If there is a transition phase between a power off command and the time it is safe to touch the power source, then we need to introduce a delay.

Commanding a power-off on removing the barrier is not an option here: we already asserted in S5 that the power supply must be off if the barrier can be opened, so any power-off must precede an open-barrier. To hold the barrier in place we need a lock. To allow the operator to command power-on/off we need a switch. To implement the delay we need a timer.

Level 5 Additional Objects:

BarrierLock

OperatorCommandSwitch

Timer

DelayTime

Level 5 Additional Relations:

Locked(BarrierLock) / Unlocked(BarrierLock)

CommandOn(OperatorCommandSwitch) / CommandOff(OperatorCommandSwitch)

CommandOn(PowerSource) / CommandOff(PowerSource) // note that we have not stated where

this comes from.

Set(Timer, DelayTime)
Start(Timer)

Finished(Timer) // can only be reached by uninterrupted counting

Level 5 Domain Assumption:

There is a process intended here for opening the barrier lock:

- 1) CommandOff(OperatorCommandSwitch)
- 2) CommandOff(PowerSource); PowerTransition(PowerSource) holds from here on
- 3) Set(Timer, DelayTime)
- 4) Start(Timer)
- 5) Finished(Timer)
- 6) PowerOff(PowerSource) holds from here on
- 7) Unlocked(BarrierLock)

Note that this sequence will work from all safe states. The power supply need not be shut off. Except for the timer the sequence does not assume timing constraints. No assumption has been made on the use of the command switch after it was set to off to initiate the process.

Refinement Level 6

Once the power supply is off and the barrier lock is open, the operator can open the barrier and perform all the maintenance heshe wants. To bring the power supply back online the barrier must be closed and locked. Only after that will the power supply come back online.

Level 6 Domain Assumption:

The start procedure of the power supply looks like this:

- 1) Locked(BarrierLock)
- 2) CommandPowerOn(OperatorCommandSwitch)
- 3) PowerOn(PowerSource)

Level 6 Proof Obligation:

Both procedures satisfy the safety requirements stated above.

The procedures are not safety requirements in themselves, but any system that will implement these procedures will also implement the safety requirements.

Commentary on Level 6:

In this example the power on/off procedures are the first procedural descriptions that have resemblance to software. A computer system governing the power on/off and lock/unlock procedures is not far away from the point we are at. We have yet to describe how the state information is passed within the system. For that, see the following refinement.

Refinement Level 7

We may introduce here a E/E/PES to govern the barrier-open/close and power-on/off processes. We will need

Level 7 Additional Objects:

PES

OpCommSwitchSensor

BarrierLockSensor

PowerSourceCommandON

PowerSourceCommandOFF

Commentary on Level 7:

There are 24 states the PES must govern:

Power On/Off (2) x Timer Set/Start/Finish (3) x CommandSwitch On/Off (2) x BarrierLock LockUnlock (2)

These states and their pre- and post-conditions are specified at this level.

Conclusions of the OHA

Our main claims are: (a) we have developed the system through seven refinement levels, many of them trivial modifications of the preceding level, using OHA, to the point at which it becomes routine to implement; and (b) these levels are routine to adherents of OHA with developed logical "bookkeeping" skills.

We introduced the device of "Safety Assumption", here SA0.1. A safety assumption is carried through the safety requirements analysis (the proof of refinement at each level) as an assumption, specifically as a conjunct in the antecedent of the RSPOs. At some point in the system development, this assumption SA0.1 must be discharged. For example, for SA0.1, the effect of the electrical field strength of the PowerSource on pacemakers, and other potential health effects, must be considered. SA0.1 acts thus as a placeholder for a future Hazan step.

The analysis of Daniel Jackson did not explicitly identify the safety assumption SA0.1; neither did Leveson's SPTA. OHA identified it through trivial propositional-logical bookkeeping at Level 0. Daniel Jackson has informed us (private communication) that his analysis has no apparent explicit need for SA0.1. We introduced SA0.1 in order, inter alia, to be able to make the Level 0 assertion of completeness of the safety requirements. Daniel Jackson does not assert completeness of his formulation of safety requirements, so it is plausible that he does not need such a device. We consider the completeness assertion at Level 0, though, to be a key element of OHA which distinguishes it (we would say: advantageously) from other Hazan methods.

We assert that the Safety Assumption SA0.1 would be identified routinely by any competent analyst trained in OHA, just as it "fell out" here during the required propositional-logical "bookkeeping".

It should also be apparent that this development, of an interlock system for a possible-harm-

inducing system state, could occur as a system component in a more complex software-based system. The Hazan as we have performed it here is necessary, and would then occur during development of architecture of the system, which might well occur later than the initial system-definition stage.

We believe we have exhibited the benefits of an OHA over Leveson's and Daniel Jackson's analyses. Assumptions and requirements are routinely identified through OHA that may be missed in other analyses. Additionally, other analysis methods do not insist on, nor necessarily achieve, a demonstrable level of completeness in the safety requirements at any given level of expression, whereas it is a key component of OHA that OHA does so at the high level. We acknowledge that Daniel Jackson's analysis has been formally checked using Alloy, and has in this respect an advantage over our purely "hand"-executed analysis presented here.

The OHA here does not support the claim of Daniel Jackson that Hazan may stop at the very high level during development of a software-based safety-critical system. On the contrary, we formulated a safety assumption which is key to showing completeness of the high-level safety analysis, as well as to proving refinement proof obligations further in the development, and must ultimately be discharged through further Hazan, or through logic using appropriate safety requirements, later in the system development. Making a safety assumption thus commits the analyst to further Hazan.

Acknowledgements

The author gratefully thanks John Spriggs and Michael Jackson for identifying mistakes in the original version of this manuscript, as well as Daniel Jackson for discussion and insightful commentary which has highlighted what we needed to more carefully formulate, and to make clearer about OHA, and which has been essential to writing the current version. The author particularly thanks Peter Bernard Ladkin and Bernd Sieker for very helpful review and comments. The current version has been largely modified by Ladkin from the previous version, to reflect the outcome of discussion between the author, Sieker and Ladkin on 21.10.2010.