System Safety

M7 Functional Hazard Analysis (FHA) V1.2

Matthew Squair

UNSW@Canberra

4 December 2015

Learning outcomes

To appropriately apply functional hazard analysis methods as part of a hazard analysis

To understand the strengths and weaknesses of the method

Overview

FHA is a known cause, unknown effect analysis that explores the effects of functional failures on the system

Various standards require or recommend it's use including ARP 4754[SAE 1994], DEF-STAN 0-56 and the JSSG Software Safety Handbook

FHA is a useful analytical tool to use during the concept design phase to populate the system level safety analysis with functional hazards, and during the preliminary desgn phase to populate the subsystem level safety analyses with functional hazards [SAE 1996]

Uses

An FHA can be used to help identify safety requirements, e.g. the acceptable rate of occurrence of hazardous functional failures

The FHA can be used to identify early in the implications of functional architecture decisions

The FHA does rely on some level of *functional analysis* having been done

Naturally dovetails with a *safety critical functions list*

Can be iterated across the design as it develops e.g conducted at both system and subsystem levels

Key definitions

Function. A very large atomic action operating (accepting inputs) over a defined interval of time, with the details of the transform and internal states hidden. Functions are specified by

- Inputs (data)
- Outputs (transform Invariant) - includes internal state updates
- Service provision (e.g start and stop (transition) criteria)
- Performance (efficiency, timing or accuracy of value metrics)

Functional timing requirement. A system must provide a response in accordance with a specified timing relationship between inputs (I) and outputs (O). Relationship may be min, max or both. There are four classes of timing relations (IO), (OO), (II), (OI)

Key definitions (cont'd)

Functional timing requirement (cont'd). Example functional timing requirements

Example

IO. The mission computer shall send an arm command to the engine no sooner than x seconds after receiving the umbilical not present signal

Example

OO. The WCS shall send a launch command no sooner than 5 seconds after it sends the activate battery command

Key definitions (cont'd)

Example

II. The missile supervisor must not depress the launch button sooner than 10 seconds after pressing the fire ready button

Example

OI. The pilot may not select a way-station until 5 seconds after the route page is displayed

The last two requirements express a required relationship with the environment, and therefore an *assumption* about the external world

Key definitions (cont'd)

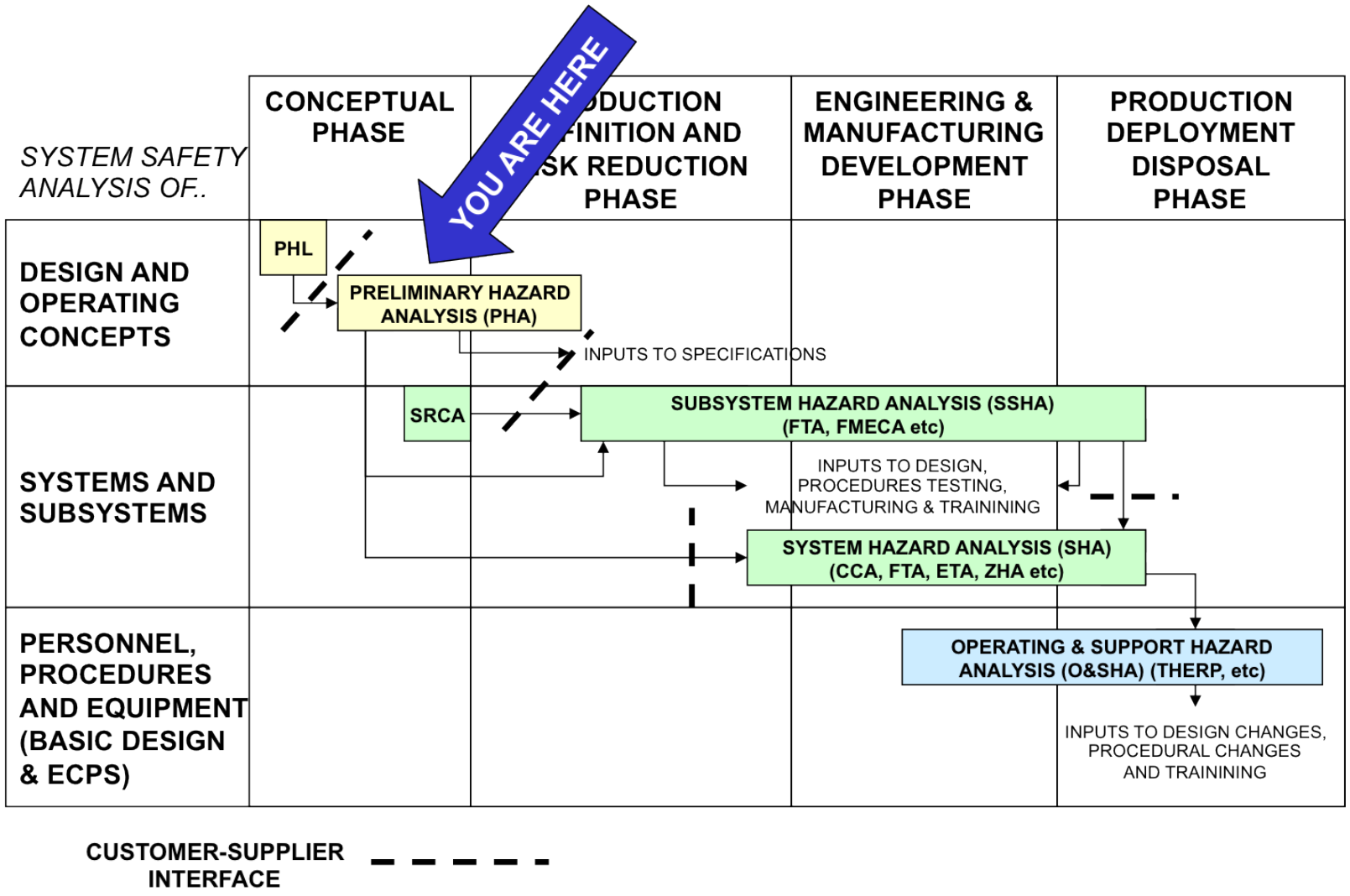
Functional failure. A functional failure is where the system fails to meet a *specified* functional requirement. Generally the model of failures used in FHA ([SAE 1994]) is the phenotype class of models, i.e. it is concerned with external effects or phenomena rather than underlying mechanisms

Functional failure mode. How a function fails to meet the specification [Pumfrey 1990]:

- Service provision (commision, ommision)
- Value (coarse, subtle)
- Timing (early, late)

Fault, Failure and Error. As per the definitions of module 1

Functional hazard analysis and the system lifecycle



Functional modelling

For complex systems the FHA must be supported by a functional analysis (modelling) methodology that

- Supports levels of abstraction
- Provides an implementation free view of the world
- Supports a stopping rule for the analysis

Should be able to model [Alford 1994]:

- Data & commands
- Sequence & concurrency
- Iteration and replication
- Functional interfaces
- Decomposition and allocation

Defining the scope

The hard work in a FHA is upfront in the scoping and problem familiarisation stage of the analysis

The analysis must be abstract enough so it does not bog down in detail, but not so abstract that meaningful insight is not obtained

Figuring out end effects when the function is part of a processing flow is not necessarily easy

Neither is determining when to stop

The stopping rule or 'don't boil the ocean'

To avoid nugatory work during the FHA it is essential to define the bounds of the analysis, it's objectives, and when to stop

Defining the scope

An initial scoping phase is necessary [Wilkinson, Kelly 1998]

1. Identify functions and their dependencies
2. Resolve ambiguities in description
3. Define the functional boundary of the analysis
4. Remove any avoidable dependencies *via design change*
5. Identify critical functions & interactions [SAE 1994]
6. Develop *functional model* to evaluate end effects
   - Develop organisational schema
   - Ensure consistent level of functional abstraction
7. Ensure traceability between levels of analysis [SAE 1994][1]
   - Higher level FHA results
   - Derived safety requirements from containing system[1]

Methodology (Based on FFA+)

1. Select a function for analysis
2. Identify single function failure modes
3. Assess the effects of this failure
   - End effects and severity
   - Contributing factors
   - Detection and recovery possibility
4. Determine associated severity
5. Document analysis in a worksheet or table

Complex or N+1 functional failures

Considering functions singularly will not identify functional interatcion type hazards

Example

Distributed control inconsistency In the Uberlingen disaster a distributed control system (TCAS) failed (in part) due to inconsistency in how resolution advisories were handled

Example

Shared control conflicts In a locomotive control system, an auto-engine start/stop function conflicted with remote controlled loco function through a shared system parameter, 'in-cab throttle position' = idle, leading to shutdown of remotely operated locomotives after five minutes

Methodology for N+1 failures (Based on FFA+)

1. Select an n-tuple of functions for analysis
2. Identify unique plausible *combinations* of N>1 failures
   - Identify symmetries and exclusivity
   - Apply other plausible criteria
     - Shared critical resources (data/physical)
     - Distributed rules about behaviour
3. Determine associated severity
4. Document analysis in a worksheet or table

FHA - Car cruise control

Example

Car cruise control system operates only when the engine is running. When the driver turns the system on, the speed at which the car is traveling at that instant is maintained. The system monitors the car's speed by sensing the rate at which the wheels are turning, and it maintains desired speed by controlling the throttle position. After the system has been turned on, the driver may tell it to start increasing speed, wait a period of time, and then tell it to stop increasing speed. Throughout the time period, the system will increase the speed at a fixed rate, and then will maintain the final speed reached.

The driver may turn off the system at any time. The system will turn off if it senses that the accelerator has been depressed far enough to override the throttle control. If the system is on and senses that the brake has been depressed, it will cease maintaining speed but will not turn off. The driver may tell the system to resume speed, whereupon it will return to the speed it was maintaining before braking and resume maintenance of that speed. (Source: Safeware PHA webpage)

FHA exercise - Car cruise control (cont'd)

Functional block diagram of car cruise control sub-functions

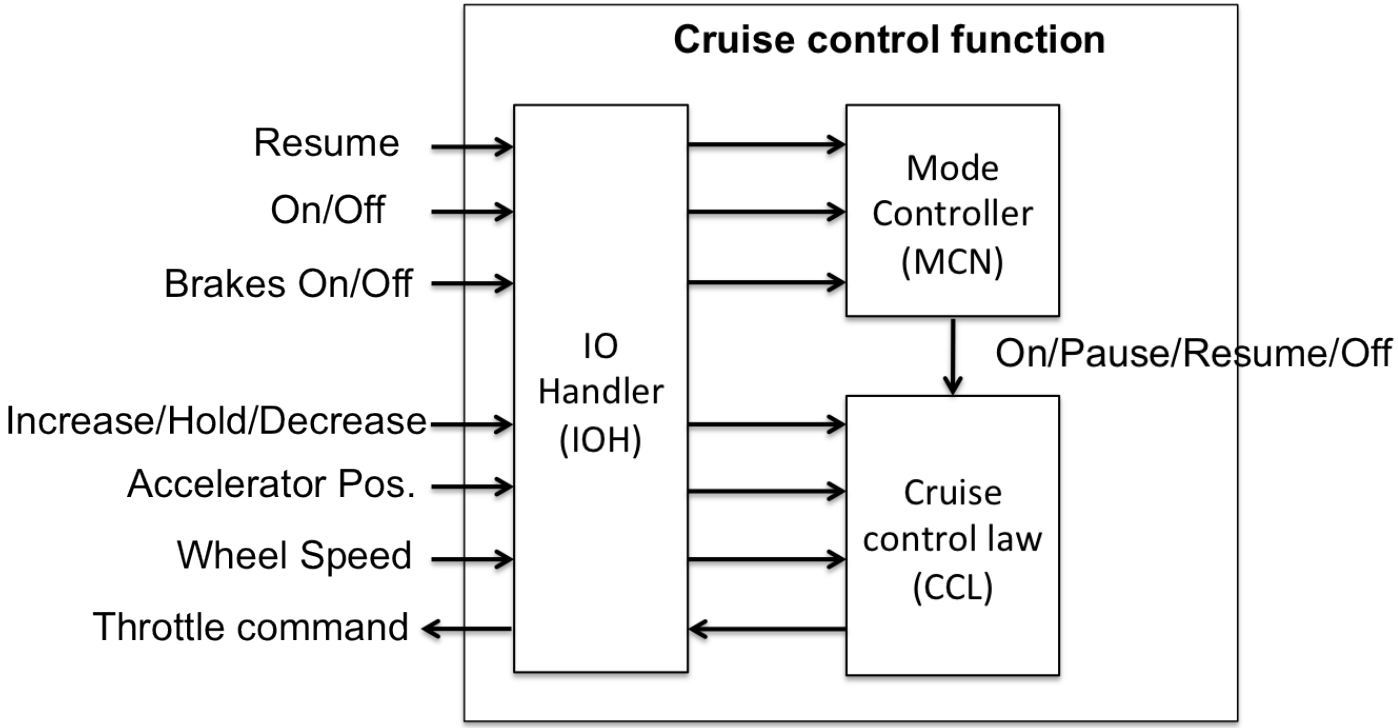


Figure: Cruise control functional block diagram

FHA exercise - Car cruise control (cont'd)

Partial functional hazard analysis worksheet for preceding example, safety requirements derived from ARP 4754 [SAE 1994]

ID	Phase/ Environ	F'n	Failure mode/ rate of development	End effect	S	DAL/ f/hr	Annunc., Detect. & recovery	Verified by
1	City (stop/go)	MC	Uncommanded resume	Tail end collision	II	C 1E-8	Low speed lockout	PSSA 110
2	Parking (low speed)	MC	Uncommanded resume	Low speed collision	III	B 1E-7	Low speed lockout	PSSA 111
3	Highway (cruise)	MC	Failure to turn off on braking	Loss of control & high speed crash	I	A 1E-9	Safety monitor	PSSA 112

Must consider the *context* of system functional failure

- *External factors*, such as weather, night-day etc
- *abnormal* system states and conditions
- *Rate* at which hazardous situation may evolve
- *Visibility* of the failure state

Consider the effects of failure propagation across interfaces

Functional hazards and the S-M-O model

Functional hazards identified in the FHA do not conform to the S-M-O model

While the hazardous source is identified (the functional failure) the mechanism by which that hazard occurs is not identified

To identify the mechanism we need to associate the functional hazard with a specific system element be it hardware or software whose failure would result in the functional hazard

Limitations of the method

Limitations

- This is a functional analysis, other sources of hazards are not considered
- Has difficulty in dealing with coupled functions
- Dealing with mode dependent behaviour adds significant scope
- Having performed the analysis we end up with *requirements*
- Higher level functions will generate more abstract hazards
- Difficult to apply at lower design levels due to increasing complexity of behaviour

An FHA establishes requirements for safety, it does not verify it

An FHA generates *functional* safety requirements, it *does not* prove that these are achieved, or achievable. Follow up safety analyses such as FTA or FMECA are required if you want to do this

Advantages and disadvantages of the method

Advantages

- Can derive functional safety requirements (SILs, DALs)
- Can identify functional hazards that SSHA should consider
- Allows functional hazards to be decomposed and allocated as part of the functional requirements analysis
- Clear focus on functional failures and effects

Disadvantages

- Can require significant design context to determine end *effects*
- the analysis can bloat if scope is not controlled
- inadvertent introduction of design detail can cloud the analysis

Conclusions

FHA relies on having a coherent and structured functional model of the system, without that the analyst is in trouble

A useful (and somtimes mandatory) technique to identify functional domain hazards as part of the concept design phase safety activities

The objective of the analysis is to obtain insight and define system safety requirements, *not* to generate reams of tabular data. When you have valid safety requirements, *stop*

Bibliography

[Alford 1994] Alford, M.W., *A Graph Model Based Approach to Specifications* in Lecture Notes Computer Science (Distributed Systems, Methods and Tools for Specifications, Advanced Course), Volume 190, Springer-Verlag 1985.

[1] EURO2015 EASA (2015) *Part 1: Functional Hazard Assessment* in EUROCONTROL, Air Navigation System Safety Assessment Methodology, Ed. 1.

[SAE 1994] SAE (1994), ARP 4754 *SAE Aerospace Recommended Practice, Guidelines For Development Of Civil Aircraft and Systems*, Society of Automotive Engineers (SAE), Inc.

[SAE 1996] SAE (1996) ARP 4761, *Aerospace Recommended Practice, Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*, Society of Automotive Engineers (SAE), Inc.

[Pumfrey 1990] Pumfrey, D.J. (1990) *The Principled Design of Computer System Safety Analyses*. York: Department of Computer Science, University of York, PhD Thesis.

[Wilkinson, Kelly 1998] Wilkinson, P.J., Kelly, T.P., (1998) *Functional hazard analysis for highly integrated aerospace systems*, Certification of Ground/Air Systems Seminar (Ref. No. 1998/255), IEE , vol., no., pp.4/1,4/6.