Prof. Chris Johnson,
School of Computing Science, University of Glasgow.
johnson@dcs.gla.ac.uk
http://www.dcs.gla.ac.uk/~johnson

University of Glasgow

Introduction

- Why is software different?
- Software requirements.
- Leveson's completeness criteria.
-
- Dublin Airport Case Study
- .

University of Glasgow

Why is Software Different?

Software is an abstract concept in that it is a set of instructions on a piece of paper or in computer memory. It can be torn apart and analysed in piece parts like hardware, yet unlike hardware it is not a physical entity with physical characteristics which must comply with the laws of nature (i.e., physics and chemistry).

Since software is not a physical entity it does not wear out or degrade over time. This means that software does not have any failure modes per se. Once developed it always works the same without variation. Unlike hardware, once a software program is developed it can be duplicated or manufactured into many copies without any manufacturing variations.

Software is much easier to change than is hardware. For this reason many system fixes are made by modifying the software rather than the hardware.

There are no standard parts in software as there are with hardware. Therefore there are no high reliability software modules, and no industry alerts on poor quality software items.

University of Glasgow

Why is Software Different?

If software has anything which even resembles a failure mode, it is in the area of hardware induced failures.

Hardware reliability prediction is based upon random failures, whereas software reliability prediction is based upon the theory that predestined errors exist in the software program.

Hardware reliability modelling is well established, however, there is no uniform, accurate or practical approach to predicting and measuring software reliability. Since software does not have any failure modes, a software problem is referred to as a software error.

A software error is defined as a situation when the software does not perform to specifications or as reasonably expected, that is when it performs unintended functions. This definition is fairly consistent with that of a hardware failure, except that the mechanisms or causes of failure are very different.

University of Glasgow

Why is Software Different?

Hardware primarily fails due to physical or chemical mechanisms and seldom fails due to human failure mechanisms (e.g., documentation errors, coding errors, specification oversights), whereas just the opposite is true with software.

Software has many more failure paths than hardware, making it difficult to test all paths.

By itself software can do nothing and is not hazardous. Software must be combined with hardware in order to do anything.

Clif Ericson, Boeing.

University of Glasgow

Types of Software Defects

A software defect is either a fault or discrepancy between code and documentation that compromises testing or produces adverse effects in installation, modification, maintenance, or testing.

Requirements Defects: Failure of software requirements to specify the environment in which the software will be used, or requirements documentation that does not reflect the design of the system in which the software will be employed.

Design Defects: Failure of designs to satisfy requirements, or failure of design documentation to correctly describe the design.

Code Defects: Failure of code to conform to software designs.

Robert Dunn, Software Defect Removal, McGraw-Hill,

University of Glasgow

Software Hazard Analysis

- Already seen software fault trees.

1. Trace identified software hazards to the software-hardware interface. Translate the identified software related hazards into requirements and constraints on software behaviour.

2. Show the consistency of the software safety constraints with the software requirements specification. Demonstrate the completeness of the software requirements, including the human-computer interface requirements, with respect to system safety properties.

Acknowledgement: Nancy Leveson, Safeware: System Safety and Computers, Addison Wesley, Reading Massachusetts, 1995.

- Point 2 links to safety case slides?

University of Glasgow

Software Requirements Analysis

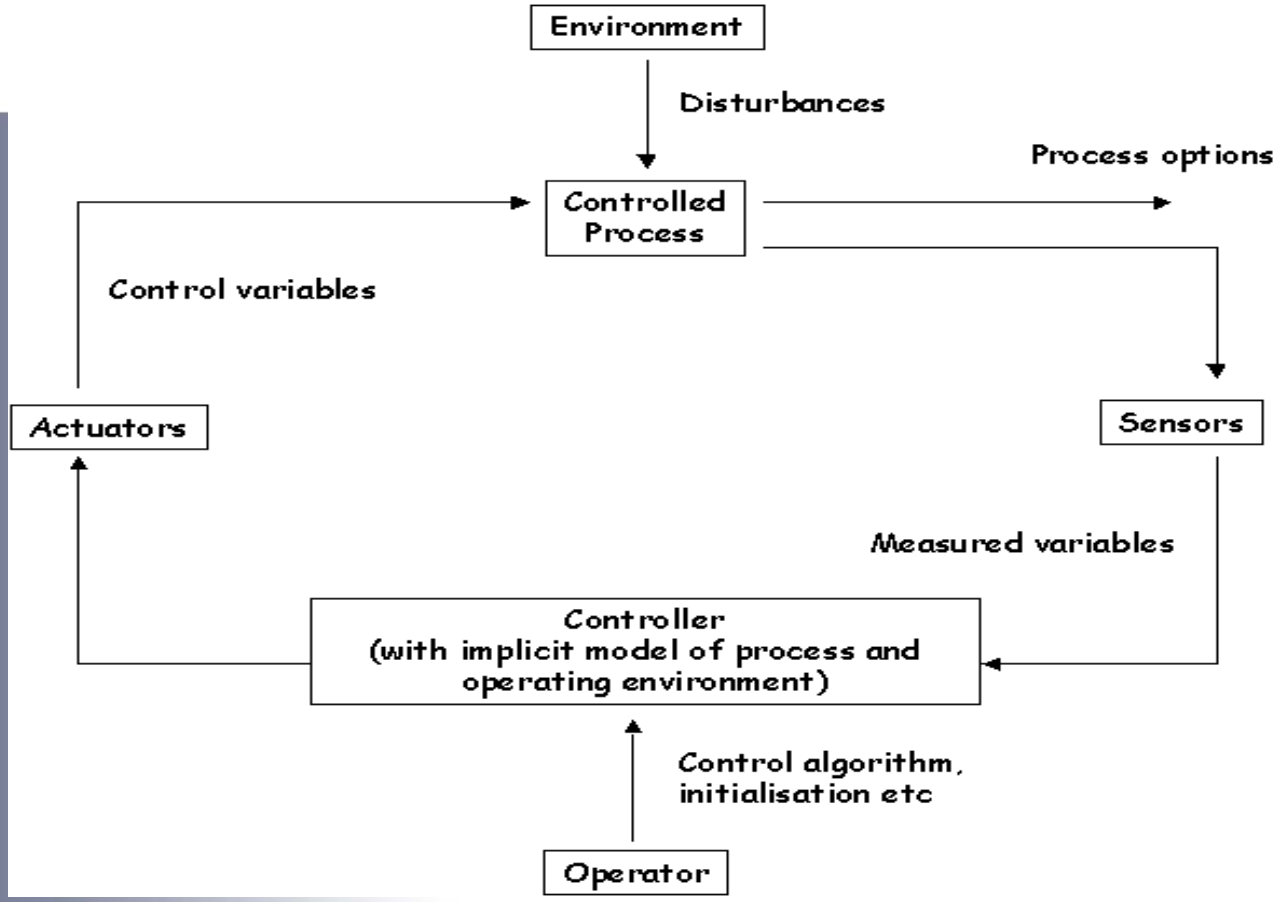
- Leveson identifies 3 components.
- Basic function or objective.
- Constraints on operating conditions.
- Prioritised quality goals;
  - to help make trade-off decisions.
- Same as general hazard analysis?
-

University of Glasgow

Kernel Requirements & Intent Specifications

- Kernel or core set of requirements.
- Determined by current knowledge of:
  - intended application functionality;
  - environment & constraints.
- Analytically independent.
- Only know they are complete if
  - we know specification intent...

Leveson's Completeness Criteria



- Remember - `Black Box' architecture.

University of Glasgow

Leveson's Completeness Criteria

- Human Computer Interface Criteria.
- State Completeness.
- Input/Output Variable Completeness.
- Trigger Event Completeness.
- Output Specification Completeness.
- Output to Trigger Relationships.
- State Transitions.

University of Glasgow

Human Computer Interface Criteria

- Criteria depend on task context.
- Eg in monitoring situation:
  - what must be observed/displayed?
  - how often is it sampled/updated?
  - what is message priority?
- Not just when to present but also
  - when to remove information...

University of Glasgow

State Completeness Criteria

- Consider input effect when state is:
  - normal, abnormal, indeterminate.
- Start-up, close-down are concerns.
- Process will change even during
  - intervals in which software is `idle'.
- Checkpoints, timeouts etc.

University of Glasgow

Input/Output Variable Completeness

- Input from sensors to software.
- Output from software to actuators.
- Specification may be incomplete if:
  - sensor isn't refered to in spec;
  - legal value isn't used in spec.

University of Glasgow

State Transitions Completeness

- Reachability: all specified states can be reached from initial state.
- Recurrent behaviour: desired recurrent behaviour must execute for at least one cycle and be bounded by exit condition.
- Reversibility: output commands should wherever possible be reversible and those which are not must be carefully controlled.
- Pre-emption: all possible pre-emption events must be considered for any non-atomic transactions.

IRISH AVIATION AUTHORITY
ÚDARÁS CITLÍOCHTA NA hÉIREANN

REPORT OF THE IRISH AVIATION AUTHORITY

INTO THE ATM SYSTEM MALFUNCTION AT DUBLIN AIRPORT

19th September 2008

CONTENTS

Dublin Airport Overview

- Busiest period of the year.
- Initial hardware failure:
  - Poor quality of service from LAN;
  - Slows flight data processing system.
- ATCOs cannot access data on radar targets:
  - including aircraft identification and type data.
- Capacity restrictions for safety reasons.

Dublin Airport - Contracting

- ATM system provided by contractor:
  - maintained under annual service contract;
  - provide both hardware and software support;
  - Preventative maintenance of components;
  - On-site support for diagnosis and debugging.
- ANSP relies upon subcontractor:
  - key areas of technical support ;
  - lacks sufficient in-house capability;
  - Is outsourcing a form of de-risking?

University of Glasgow

Initial Failure

- First symptoms observed:
  - aircraft id & type not displayed by flight tracks;
  - but only for flights entering system...
- ANSPs engineering staff correct symptoms;
  - Cannot identify root causes of the problem.
- Capacity restrictions to maintain safety levels;
  - Above operating demands so little impact?

Secondary Response

- Problem stemmed from double failure:
  - triggered by a faulty network interface card;
  - flooded network with spurious messages;
  - delayed FDPS updates on network.
- Symptoms of the fault were masked;
  - recovery mechanisms in Local Area Network;
  - made it hard for engineering teams to identify initial component failure.

University of Glasgow

Aging, Complex Critical Infrastructures...

University of Glasgow

Conclusions

- Why is software different?
- Software requirements.
- Leveson's completeness criteria.
-
- Dublin Airport Case Study
- .

University of Glasgow

Any Questions…