Therac-25

Will history repeat itself?

By Ash Tyndall

Linear Accelerators

- “LINACs”
- Medical application: treatment of various illnesses, particularly cancer.
- Dispense X-ray or Electron based radiation.
- Two companies involved:
  - Atomic Energy of Canada Limited (AECL)
  - Compagnie General Radiographique (CGR)
- Therac-25, based on AECL and CGR work, released in 1983.

Remove hardware interlocks

100% software safeguards

It’ll be fine…

- At least six accidents.
- Four deaths.
- Dosages hundreds or thousands of times greater than normal.

Importance

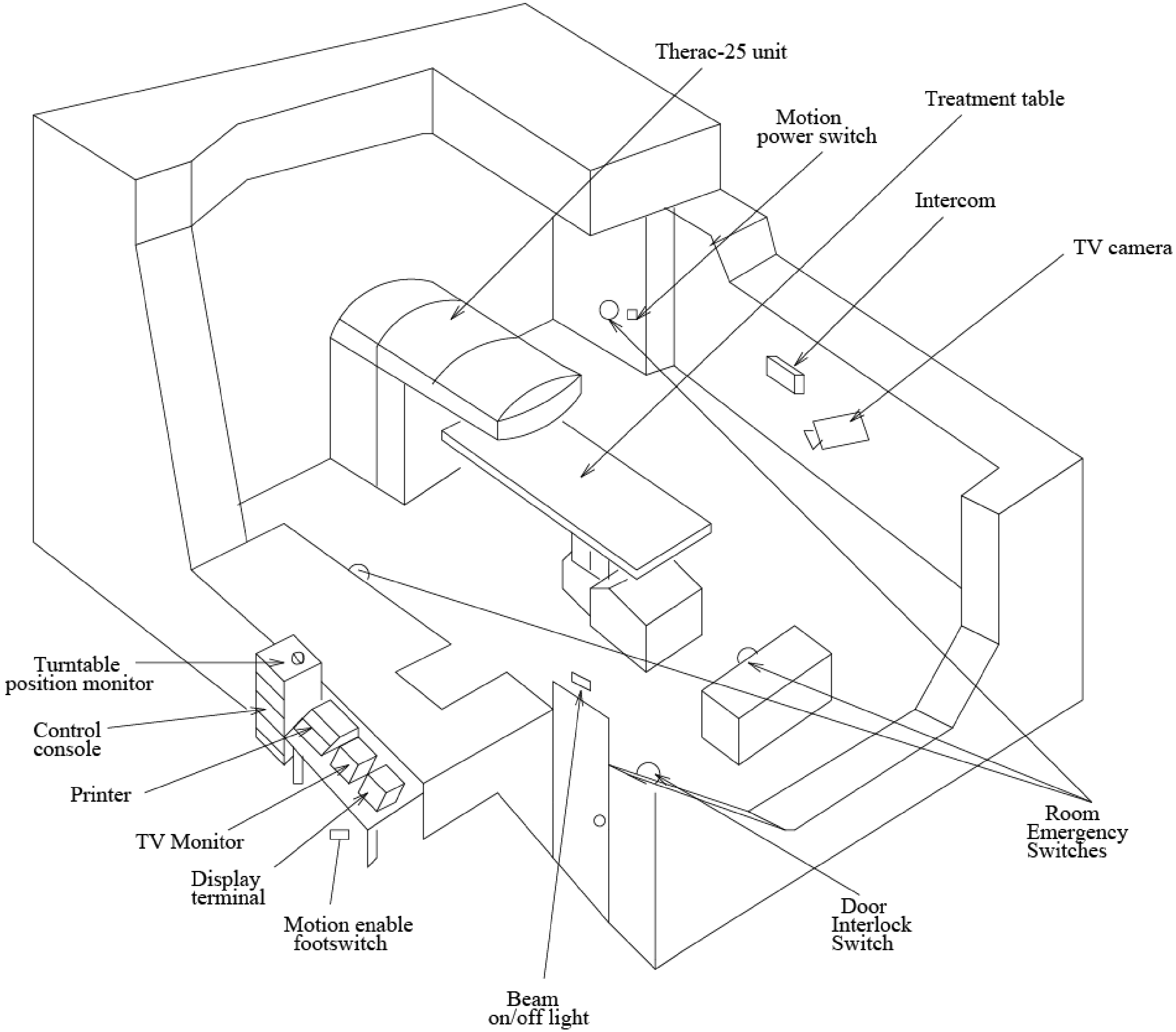
- Fairly self-evident
- Medical software has an enormous capacity to kill and injure
- All efforts should be made to prevent this
  - Understand what has gone wrong in the past
  - Know what to change in the future
  - Know what changes prevent/don't prevent accidents

Therac-25: Will history repeat itself?

- Introduction
- The Incident
- Safe Software Design
- Post-Therac
- Conclusion

The Incident

What went wrong with Therac-25?

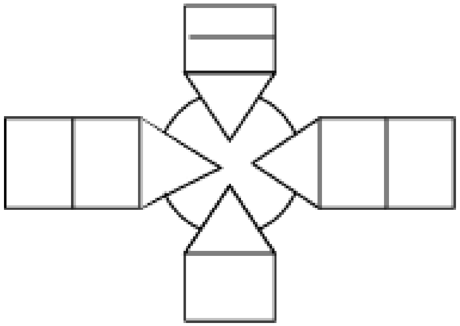


Typical Therac-25 facility circa 1988

Taken from [5]; Leveson, N. 1995.

Design Preconditions

- **Scanning magnet**
  - Converted X-rays into electrons
  - Process causes a lot of radiation to be lost as heat
  - Requires substantial increase in X-ray output for equivalent electron beam power



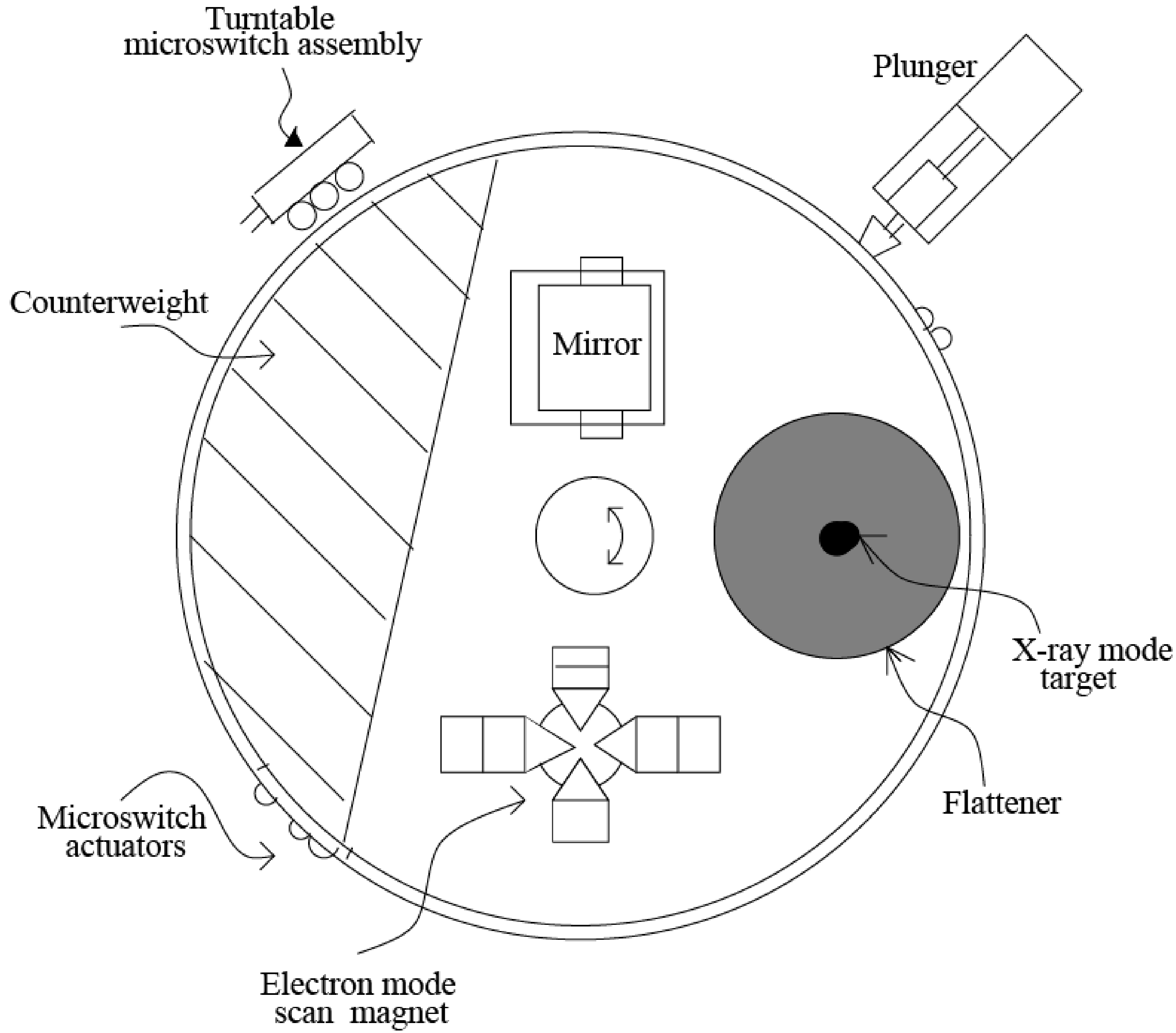
Taken from [7]

[4] Leveson, N. G., and Turner, C. S. 1993. [7] Leveson, N. 1995.

Design Preconditions

- **Rotating turntable**
  - Beam flattener would rotate into place in front of X-ray emitter
  - This process would take about 8 seconds
  - Safeguard in software to prevent operator from activating beam before that

[4] Leveson, N. G., and Turner, C. S. 1993.

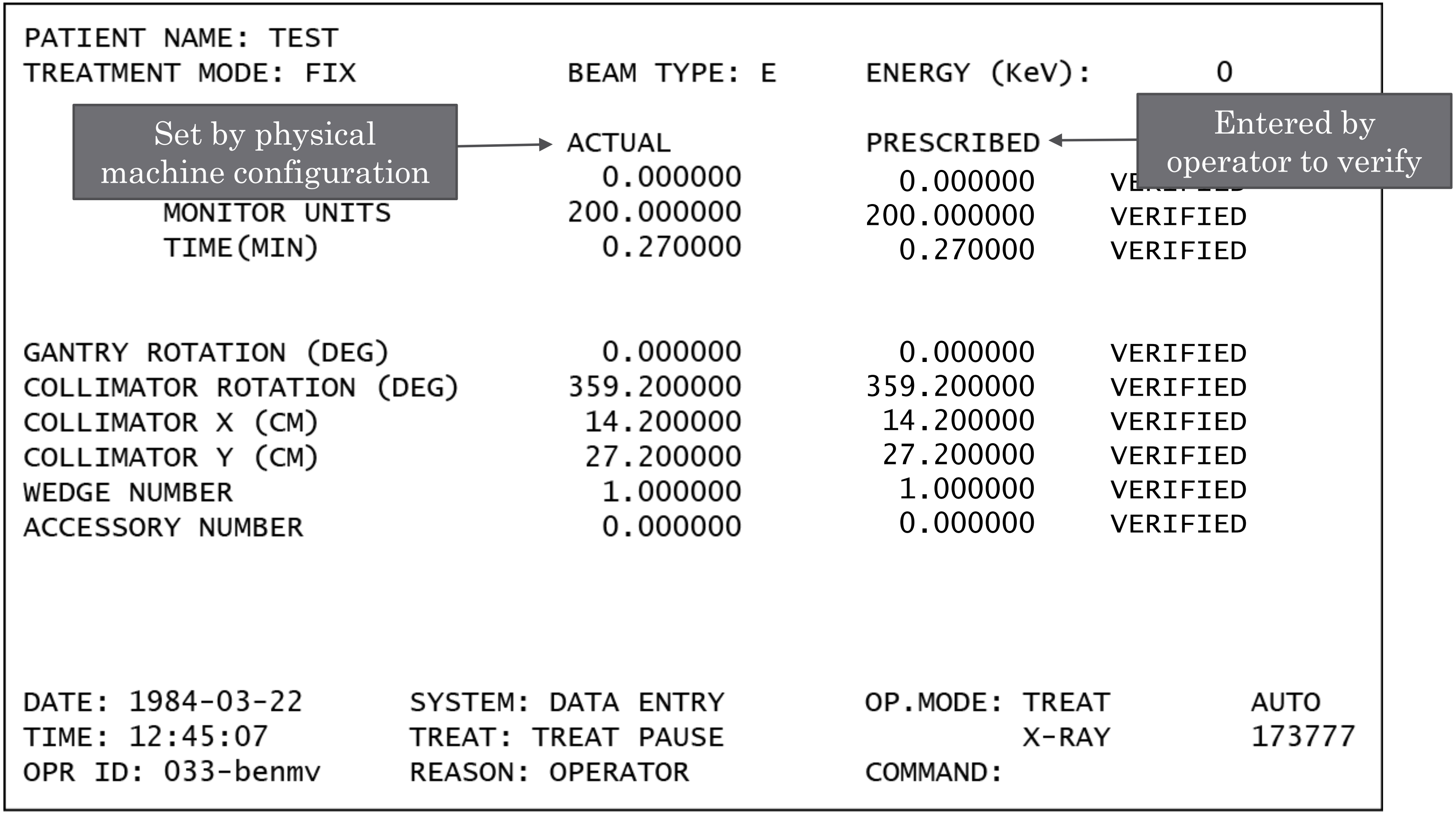


Upper turntable assembly

Taken from [7]; Leveson, N. 1995.

- **Press “Enter” to verify**
  - Operator could press Enter to confirm that “Actual” was correct without entering “Prescribed”. Very quick process.

[4] Leveson, N. G., and Turner, C. S. 1993.



Therac-25 interface

Taken from [7]; MIT Department of EE and CS. 2007.

What actually went wrong?

- The main problem: Race condition in software.
- Changing between X-ray and electron mode didn't pause the machine for 8 seconds, like it should have.
- If operator skipped through too quickly after changing from X-ray to electron mode, the turntable wouldn't completely rotated, and the patient would get a massive dose of X-rays.

[4] Leveson, N. G., and Turner, C. S. 1993.

Safe Software Design

How do we design safe software?

Acceptable “Mishap Risk”

- Safety can’t be 0% probability
- Humans are willing to accept some risks
- Mishap
  - “an unplanned event or series of events that result in death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment” — Department of Defence via Dunn [3]
- Takes into account both:
  - Probability of it occurring.
  - Severity if it were to occur.

[3] Dunn, W. R. 2003.

Safeware

- Nancy Leveson
- Published 1995
- “Safeware” design principles
- “Safeware” culture

[5] Leveson, N. 1995.

"Safeware" design

- Hazard-free design
  - Get rid of it
- Hazard reduction
  - Reduce the likelihood of it being hazardous
- Hazard control
  - Reduce the likelihood of it causing an accident
- Damage control
  - Know what to do if it all goes wrong

[5] Leveson, N. 1995.

Engineering a Safer World

- Nancy Leveson
- Published 2011
- State of the art
- System-Theoretic Accident Model and Processes

[6] Leveson, N. 2011.

STAMP

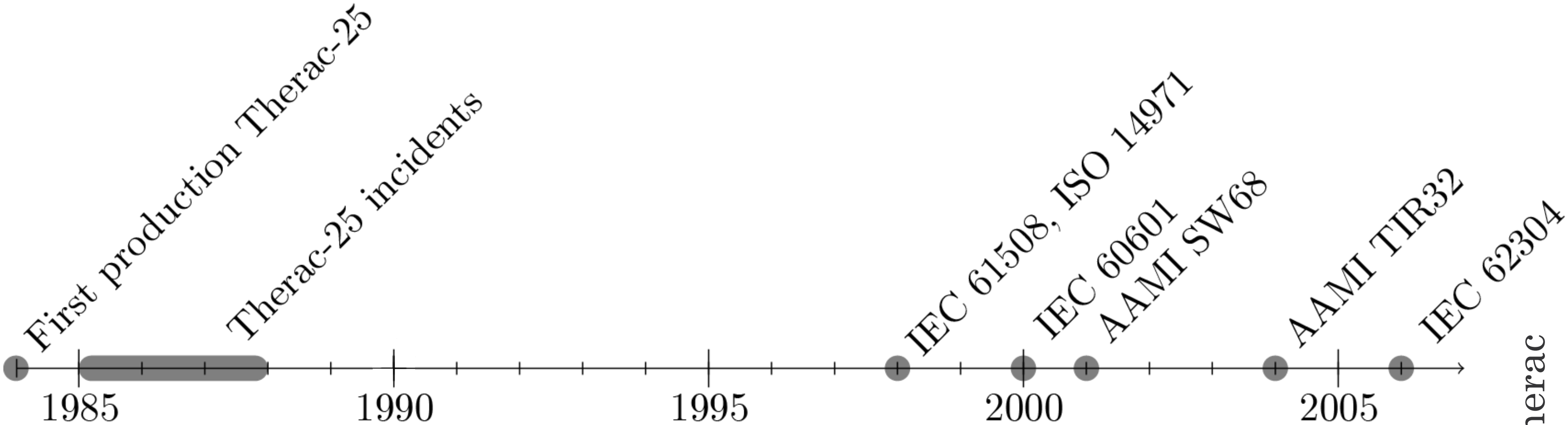
- Forms a core part of *Engineering a Safer World*
- Components:
  - Safety constraints, not events
  - Hierarchical safety control structures
  - Process models

[6] Leveson, N. 2011.

Post Therac

What was done to stop Therac-25 reoccurring?

- Regulatory bodies slow to react
- Complicated, with a variety of national and international standards
- Discussed in more detail in report



Regulation timeline

Data Analysis

- Did international regulation cause a reduction in medical device software errors?
- “Manufacturer and User Facility Device Experience” Database.
- Database of “adverse events,” (i.e. mishaps) which a variety of parties are obligated to report into.
- Available freely on the US FDA website.
- Covers mostly 1995—2014.

[8] US FDA, 2014.

Event Categories (1,000+)

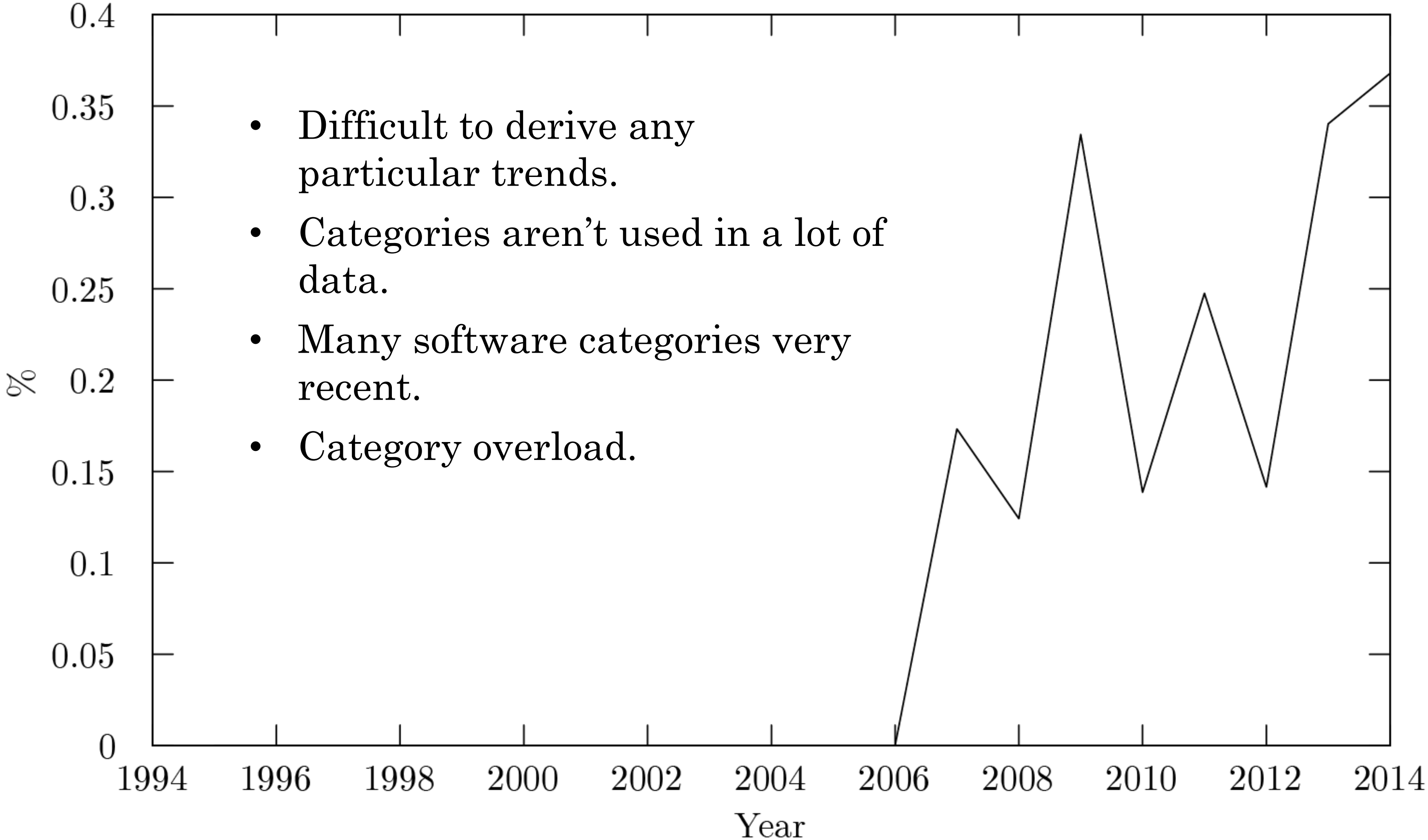
- Failure to run on AC/DC
- Abnormal
- Absorption
- Accessory incompatible
- Measurements, inaccurate
- Adaptor, failure of
- Agglutinate, failure to
- Automatic injection system overinfusion
- Failure to back-up
- Failure to convert to back-up
- Balloon rupture
- Balloon asymmetrical
- Balloon burst
- Contamination during use
- Intermittent continuity
- Continuous
- Continuous firing
- Continuous mode failure
- Use of Incorrect Control Settings
- Cool, failure to
- Coolant, contraindicated
- Cooling system, failure of
- Insufficient cooling
- Display misread
- Erratic display
- No display or display failure
- Incorrect display
- Disposable
- Dissection
- Distilled water, contaminated
- Dome collapse
- Rupture due to trauma
- Saline, use of homemade
- Salt tablet(s), use of
- Seal, incorrect
- Sediment filter problems
- Self-activation or keying
- ……
- ……

[8] US FDA, 2014.

“Software” categories

- Computer failure
- Computer hardware error
- Computer software issue
- Incorrect display
- Error or warning message, failure to produce
- Power calculation error due to software problem
- Incorrect software programming calculations
- Algorithms, inconsistent
- Semiautomatic code, failure to override
- Year 2000 (Y2K) related problem
- Date-related software issue
- Application network issue
- Application program issue
- Application program version or upgrade problem
- Application security issue
- Computer operating system issue
- Computer system security issue
- Data back-up problem
- Loss of Data
- Operating system becomes non-functional
- Operating system version or upgrade problem
- Problem with software installation
- Programming issue

[8] US FDA, 2014.



- Difficult to derive any particular trends.
- Categories aren't used in a lot of data.
- Many software categories very recent.
- Category overload.

Proportion of incidents in "software related" categories over time

Based on data from [8]; US FDA. 2014.

Regulatory Gaps

- In-house software
- Much more difficult to regulate, as it's not marketed publicly.
- 1 case: > 20% overdose due to in-house software providing data in a format the LINAC did not understand.
- Miscalculation of radiation dispersion caused 3%-7% overdose in ~4,000 people.
- "[M]any radiotherapy centres use such handcrafted software, which are not standardised and may not be thoroughly checked before clinical use."
  — Derreumaux et. al. [2]

[2] Derreumaux, S., et. al. 2008.

Conclusion – My thoughts

- Safe software design requires consideration from the beginning.
- Therac-25 did not consider safety from the beginning, and suffered serious bugs as a result.
- Regulators were very slow to regulate against medical device software, which is very worrying.
- In-house software shows that regulation is not necessarily sufficient, nor will it regulate quickly enough.
- Will history repeat itself?
  Unfortunately, it looks like it already has.

Future Research

- How have LINAC interfaces changed from a HCI perspective, and have they made incidents like Therac-25 less prevalent?
- Correlating the introduction of regulation at national and international levels and the prevalence of adverse events.
- Manually categorising some portion of MAUDE to get better insight into the prevalence of software-caused adverse events.
- More in-depth research into in-house software and how regulators could help product consumers from it.

References

1) Besnard, D., Baxter, G., et al. Human compensations for undependable systems. Tech. Rep. CS-TR-819, University of Newcastle upon Tyne, 2003. Accessed: 2014-09-20, URL: http://www.dirc.org.uk/publications/techreports/papers/12.pdf.

2) Derreumaux, S., Etard, C., Huet, C., Trompier, F., Clairand, I., Bottollier-Depois, J.-F., Aubert, B., and Gourmelon, P. Lessons from recent accidents in radiation therapy in france. *Radiation protection dosimetry* (2008).

3) Dunn, W. R. Designing safety-critical computer systems. *Computer 36*, 11 (2003), 40-46.

4) Leveson, N. G., and Turner, C. S. An investigation of the Therac-25 accidents. *Computer 26*, 7 (1993), 18-41.

5) Leveson, N. *SafeWare: System Safety and Computers*. Computer Science and Electrical Engineering Series. Addison-Wesley, 1995.

6) Leveson, N. *Engineering a Safer World: Systems Thinking Applied to Safety*. Engineering systems. MIT Press, 2011.

7) MIT Department of Electrical Engineering and Computer Sciences. Therac-25 Hands-On Assignment, 6.033 - Computer System Engineering, 2007. Accessed: 2014-10-20, URL: http://web.mit.edu/6.033/2007/wwwdocs/assignments/handson-therac.html.

8) U.S. Food and Drug Administration. Manufacturer and User Facility Device Experience Database (MAUDE), 2014. Accessed: 2014-09-20, URL: http://www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/PostmarketRequirements/ReportingAdverseEvents/ucm127891.htm.

9) U.S. Food and Drug Administration. What is a Serious Adverse Event?, 2014. Accessed: 2014-09-20, URL: http://www.fda.gov/safety/medwatch/howtoreport/ucm053087.htm.

10) Wallace, D. R., and Kuhn, D. R. Failure modes in medical device software: an analysis of 15 years of recall data. *International Journal of Reliability, Quality and Safety Engineering 8*, 04 (2001), 351-371.

Cut content follows…

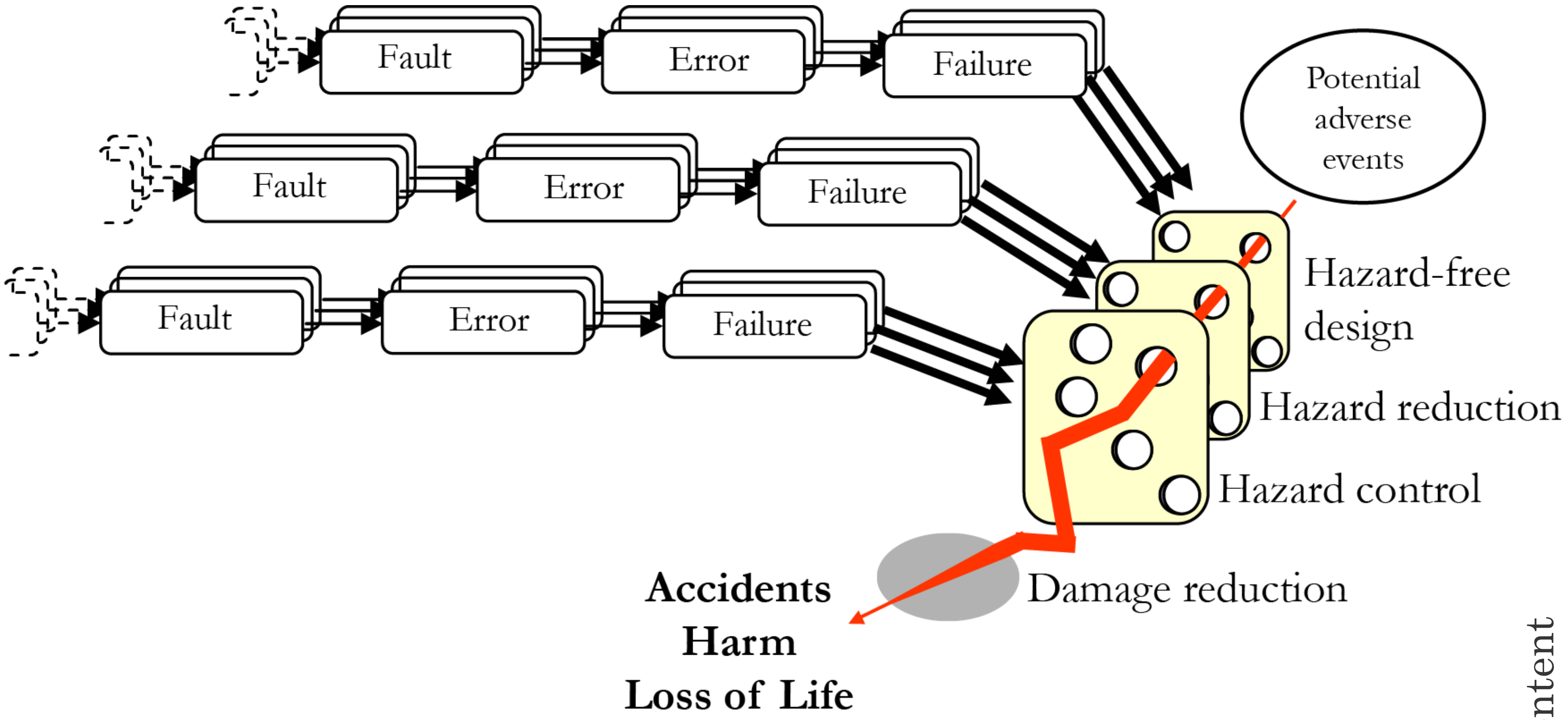
A Brief History

- 1960s-70s:
  - CGR develops the Neptune and Sagittaire LINACs
- Mid-70s
  - CGR and AECL partner to build the Therac-6 and later the Therac-20, based on CGR's previous design, but augmented with computer control.
  - AECL develops a new "double-pass" design for LINACs.
- 1981
  - CGR and AECL officially part ways due to "competitive pressures."
- 1983
  - Therac-25, based on "double-pass" design, begins public sale by AECL.

“Safeware” principles

- Safety design from the beginning.
- Safety at every level of a company.
- Safety culture.

[5] Leveson, N. 1995.



Cut Content

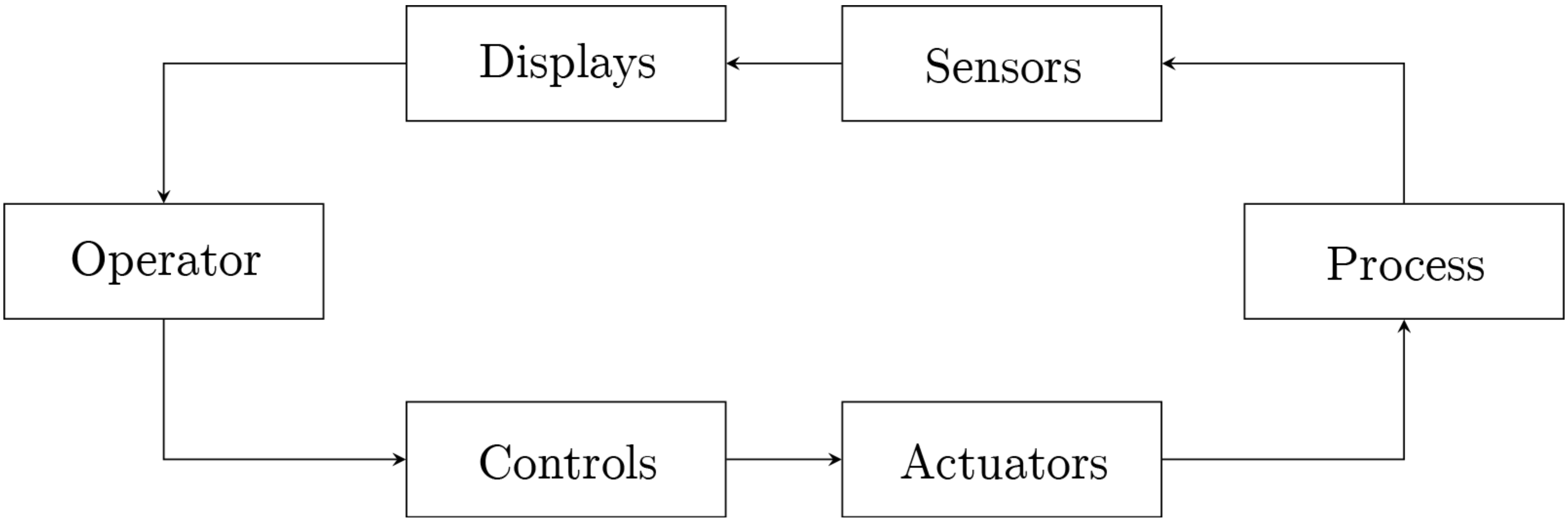
Model of *Safeware*'s hazard reduction

Adapted from [1]; Besnard, D., Baxter, G., et al. 2003.

A failure at multiple levels

- Little documentation kept on software
- Poor testing practices
  - AECL didn't test software separately
- Software of Unknown Pedigree
  - Based on the software of the Therac-6
  - Same bug found to exist in the Therac-20 (but non-fatal due to hardware interlocks)
- Poor Human-Computer Interaction
  - "Treatment Pause" function; operator could continue after error
  - Meaningless and unexplained error codes
  - "Verify" accuracy by pressing enter; habituated not thoroughly checking

[4] Leveson, N. G., and Turner, C. S. 1993.



Cut Content

Synthesised perception model

Adapted from Fig. 4.1 of [6]; Leveson, N. 2011.

Control Theory

- Human as Monitor
- Given poor information
  - Sensors indicating radiation dosage were saturated by the extreme overdose
  - System error messages were just error codes with no documentation
  - Both dangerous and benign errors had the same code, operator being habitualized
- How can the user be a successful monitor, and keep the system in check, if their perception of the system is so poor?

[5] Leveson, N. 1995. [6] Leveson, N. 2011.

STAMP

"In systems theory, emergent properties, such as safety, arise from the interactions among the system components.

The emergent properties are controlled by imposing constraints on the behaviour of and interactions among the components.

Safety them becomes a *control* problem where the goal of the control is to enforce the safety constraints.

Accidents result from inadequate control or enforcement of safety-related constraints on the development, design and operation of the system."

— Leveson [p. 75, 6]

[6] Leveson, N. 2011.

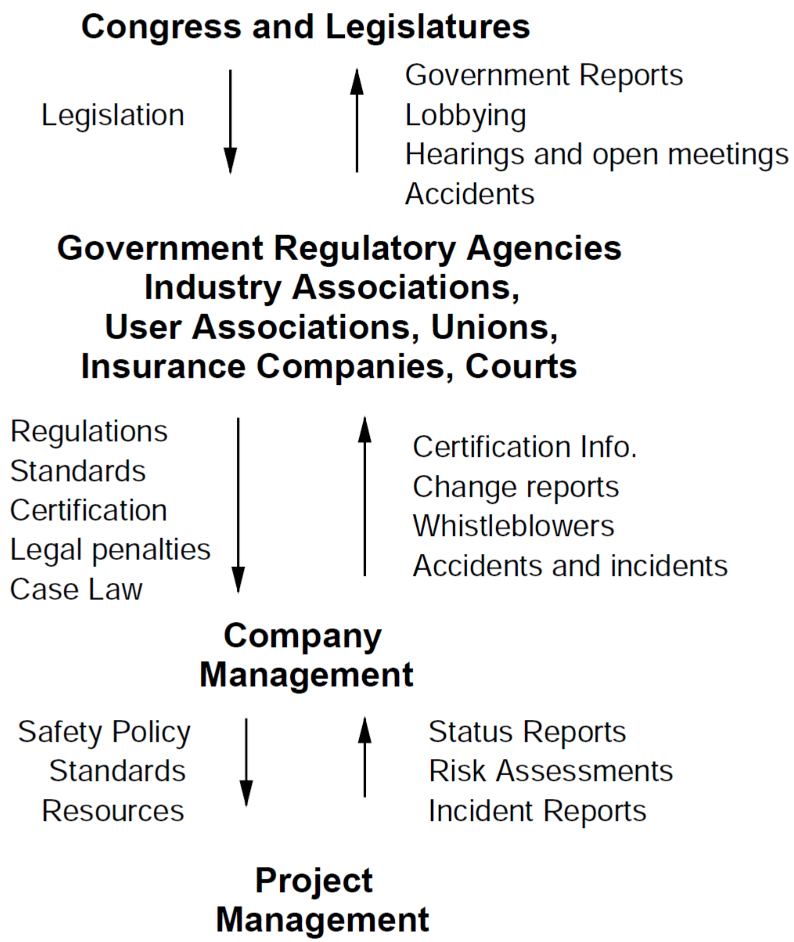
STAMP

- **Safety constraints, not events**
- "Events leading to losses occur only because safety constraints were not successfully enforced."
  — Leveson [p. 76, 6]
- Safety is a reliability problem, not a control problem.

[6] Leveson, N. 2011.

STAMP

- **Hierarchical safety control structures**
- Constraints enforce things below them in a hierarchy.



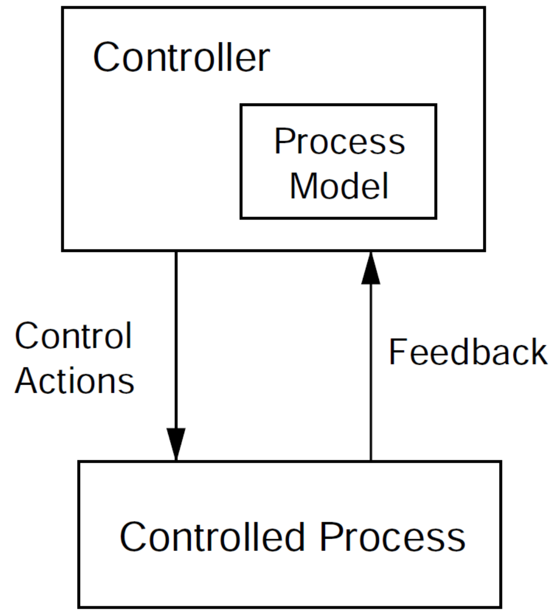
Excerpt from sociotechnical control model
Taken from [p. 82, 6]

[6] Leveson, N. 2011.

Cut Content

STAMP

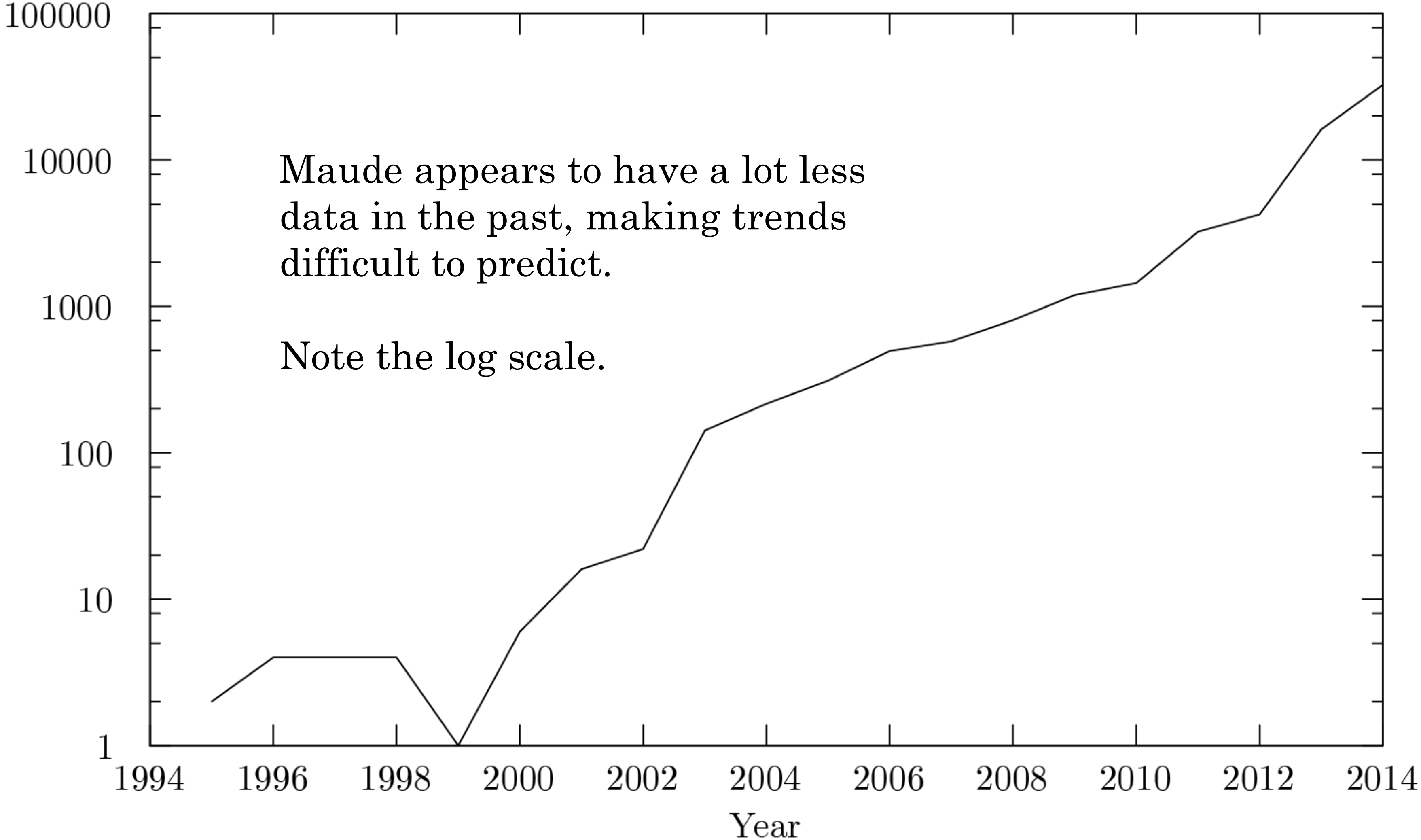
- **Process models**
- Requirements to control a process
  - Goals
  - Action Condition
  - Observation Condition
  - **Model**
- Meaning
  - Required safety constraints
  - How to accomplish them
  - How to know it's working
  - How actions cause observations, and thus goals



Process Model in Controller
Taken from [p. 88, 6]

[6] Leveson, N. 2011.

Cut Content



Cut Content

Number of reports in MAUDE over time (log scale)

Based on data from [8]; US FDA. 2014.

Other Data

- Wallace & Kuhn
- 1983-91: 6% of incidents are software related.
- 1994-96: 11%, 10%, 9%
- Contacted for more data, unfortunately didn't have more specific dates.

[10] Wallace, D. R., and Kuhn, D. R. 2001.