“An Investigation of the Therac-25 Accidents” by Nancy G. Leveson and Clark S. Turner

Catherine Schell

CSC 508

October 13, 2004

Description of Therac-25

- The Therac-25 is a medical linear accelerator.
  - Accelerates high-energy beams that can destroy tumors with minimal impact on surrounding tissue
  - Beam can be accelerated electrons or X-ray photons.

Development of the Therac-25

- Early 1970's: Atomic Energy of Canada Limited (AECL) and CGR, a French company, collaborated and developed the Therac-6 and, later, the Therac-20
  - Therac-6: 6 MeV accelerator that produced X-rays only
  - Therac-20: 20 MeV dual-mode accelerator
  - Both were versions of older CGR machines that were augmented with computer control

Development of the Therac-25 cont'd

- Mid 1970's: AECL developed "double-pass" accelerator
  - This was used in the design of the Therac-25
- 1976: AECL produced first hardwired prototype of the Therac-25
- 1981: AECL and CGR did not renew their agreement due to competitive pressures

Development of the Therac-25 cont’d

- 1982: Computerized commercial version of the Therac-25 available
- March 1983: AECL performed safety analysis, which made several assumptions:
  - Programming errors reduced by extensive testing; software errors not included in analysis
  - Software does not degrade
  - Computer execution errors caused by faulty hardware and random errors due to noise

Important Features of the Therac-25

- AECL designed Therac-25 to use computer control from the start.
  - Therac-6 and Therac-20 had histories of clinical use without computer control
- Therac-25 software had more responsibility for safety than in previous machines.
- Software in the Therac-6 and Therac-20 was reused in the Therac-25.

Therac-25 Software

- Four major components:
    - Stored data
    - Scheduler
    - Set of critical and non-critical tasks
    - Interrupt services
- Software allows concurrent access to shared memory
- Software has no real synchronization aside from data stored in shared variables
- “Test” and “set” operations for shared variables are not indivisible

Major Event Timeline: 1985

- June
  - 3rd: Marietta, GA overdose
  - Hospital physicist called AECL to ask if overdose by Therac-25 possible, received reply three days later saying it was not
- July
  - 26th: Hamilton, Ontario, Canada overdose; machine repeatedly shut down with “H-tilt” error message; AECL notified, cause determined as microswitch failure
- August
  - 1st: Four users in the US were advised in a letter from AECL to check ionization chamber to make sure it was positioned correctly; treatment should be discontinued if an “H-tilt” message with incorrect dosage displayed

Major Event Timeline: 1985 cont’d

- September
  - AECL changes microswitch, notifies users
  - Independent consultant for Hamilton clinic recommends potentiometer on turntable
- October
  - Georgia patient files suit against AECL and hospital

Major Event Timeline: 1985 cont’d

- November
  - 8th: Letter from Canadian Radiation Protection Bureau to AECL asking for hardware interlocks and software changes
- December
  - Yakima, WA overdose

Major Event Timeline: 1986

- January
  - Attorney for Hamilton clinic requests potentiometer on turntable
  - 31st: Letter to AECL from Yakima reporting possibility of overdose
- February
  - 24th: Letter from AECL to Yakima saying overdose not possible, no other incidents had occurred

Major Event Timeline: 1986 cont’d

- March
  - 21st: Tyler, TX overdose: AECL notified; AECL claims overdose impossible, no other accidents occurred, suggests electrical problem in hospital as cause
- April
  - 7th: Tyler machine put back in service after no electrical problem found
  - 11th: Second Tyler overdose: AECL notified; AECL finds software problem
  - 15th: AECL files accident report with the FDA

Major Event Timeline: 1986 cont’d

- May
  - 2nd: FDA declares Therac-25 defective; FDA asks for CAP and proper notification of users
- June
  - 13th: AECL submits CAP to FDA
- July
  - 23rd: FDA responds, asks for more info
- August
  - First user group meeting

Major Event Timeline: 1986 cont’d

- September
  - 26th: AECL sends FDA additional info
- October
  - 30th: FDA requests more info
- November
  - 12th: AECL submits revision of CAP
- December:
  - Therac-25 users notified of software bug
  - 11th: FDA requests further changes to CAP
  - 22nd: AECL submits second revision of CAP

Major Event Timeline: 1987

- January
  - 17th: Second Yakima, WA overdose
  - 26th: AECL sends FDA revised test plan
- February
  - Hamilton clinic investigates first accident, concludes overdose occurred
  - 3rd: AECL announces changes to Therac-25
  - 10th: FDA notifies AECL of adverse findings declaring Therac-25 defective under US law, asks AECL to notify users not to use it for routine therapy; Health Protection Branch of Canada does the same.

Major Event Timeline: 1987 cont'd

- March
  - Second user group meeting
  - 5th: AECL submits third revision of CAP
- April
  - 9th: FDA requests additional info from AECL
- May
  - 1st: AECL submits fourth revision of CAP
  - 26th: FDA approves CAP subject to final testing and safety analysis

Major Event Timeline: 1987 cont’d

- June
  - 5th: AECL sends final test plan and draft of safety analysis to FDA
- July
  - Third user group meeting
  - 21st: AECL submits fifth revision of CAP

Major Event Timeline: 1988

- January
  - 29th: Interim safety analysis report issued
- November
  - 3rd: Final safety analysis report issued

Lessons Learned

- Do not put too much confidence in the software.
- Do not remove standard hardware interlocks when adding computer (software) control.
- Software should not be solely responsible for safety.

Lessons Learned cont’d

- Systems should not be designed wherein a single software error can be catastrophic.
- Software error should not be the last possibility investigated in an accident.
- Engineers need to design for the worst case.

Lessons Learned cont’d

- Companies building hazardous equipment should include
  - hazard logging and tracking
  - incident reporting
  - incident analysis

  as part of quality control procedures.
- Risk assessment numbers should be meaningful, and statistics should be treated with caution.

Lessons Learned cont’d

- Documentation is important.
- Software quality assurance practices and standards should be established.
- Designs should be simple.
- Error logging or software audit trail reporting should be designed into the software from the beginning.
- System testing alone is not adequate; there should also be testing and formal analysis at the module and software levels.

Lessons Learned cont’d

- Safety-critical software projects must incorporate safety-analysis and design procedures.
- Reusing software modules does not guarantee safety in the new system.
- Software engineers need additional training and experience when working on safety-critical systems.

Lessons Learned cont’d

- Software engineers need
  - better training in interface design, or
  - more input from human factors engineers.
- There must be recognition of the potential conflict between user-friendly interfaces and safety.

Lessons Learned cont’d

- Reasons for design decisions must be recorded.
- Users of safety-critical systems should be involved in resolving problems.