

CS 4001: Computing, Society & Professionalism



Munmun De Choudhury | Assistant Professor | School of Interactive Computing

Week 1: Case Study: Therac-25
January 11, 2018

Homework 1

- Available on class website:
http://www.munmund.net/courses/spring2018/Homework_1.pdf
- Due: January 23, 2018 (11:55pm Eastern Time)
- Submission on TSquare.

Discussion on Class Activity 1a: Ethical Robot

- Imagine that you work for a computer company. You have just been assigned to a team which is charged with coming up with an approach for building a robot capable of making ethical or moral decisions. Where would you start? How would you proceed? Is such a task one that you believe is even possible?

Discussion on Class Activity 1b: Ethical Robot

- Do we need robots capable of making ethical or moral decisions? When? For what?
- Are robots the kinds of entities capable of making ethical or moral decisions?
- Whose morality or what morality should be implemented?

Genesis of the Therac-25


- Atomic Energy of Canada Ltd (AECL) and French company CGR built Therac-6 and Therac-20
- Therac-25 built by AECL
 - PDP-11 an integral part of system
 - Hardware safety features replaced with software
 - Reused code from Therac-6 and Therac-20
- First Therac-25 shipped in 1983
 - Patient in one room
 - Technician in adjoining room

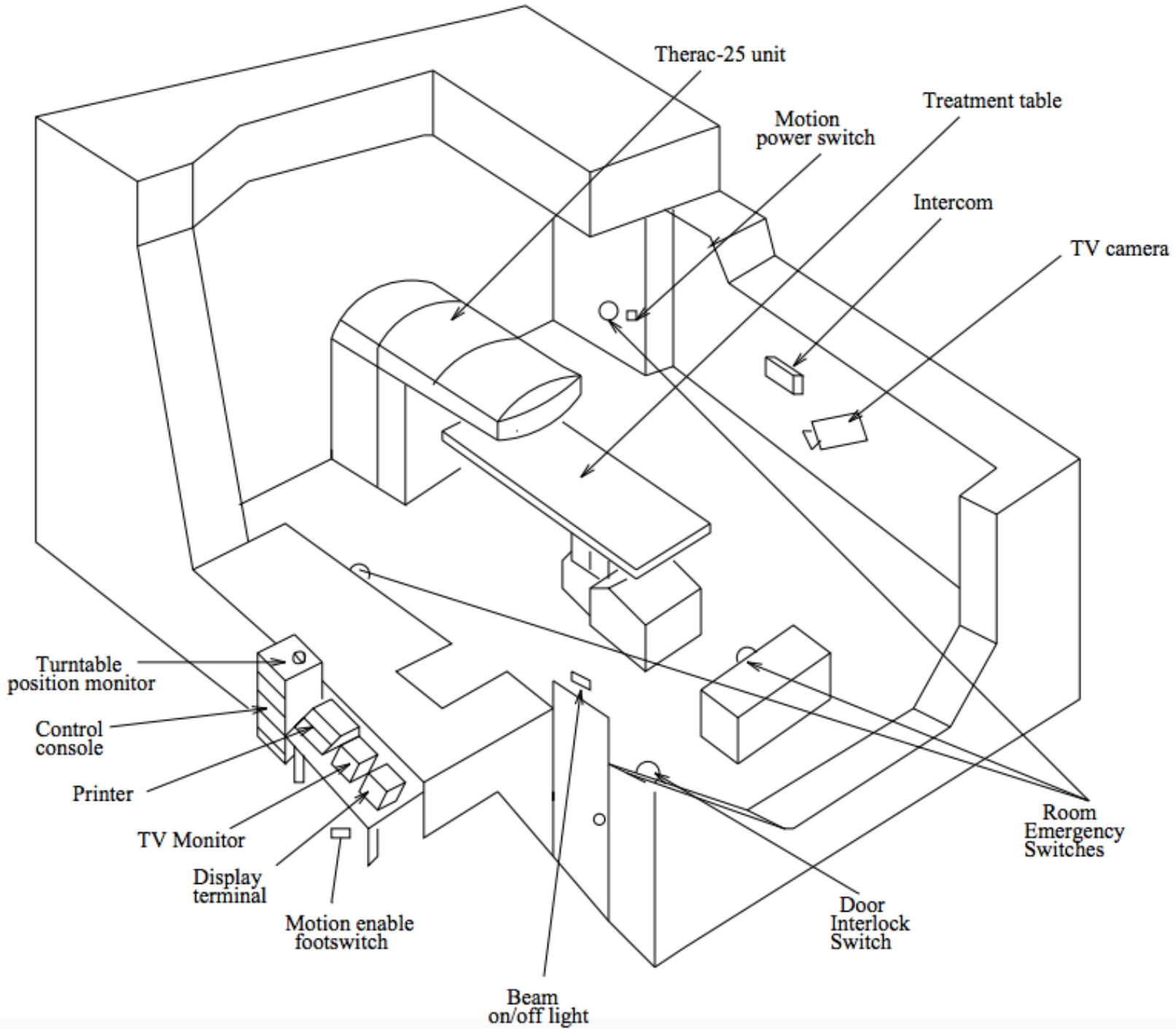
The Context

- Radiation therapy
 - Many people with cancer were diagnosed and treated, but were also exposed more radiation than they needed

The Context

- 11 installed machines; 6 major accidents; 3 deaths
 - Improper scanning of the spread of the radiology beam, causing radiation burn and secondary cancer
- Denial – manufacturer and operation refused to believe that the system could make a mistake

- 
- A Philadelphia hospital gave the wrong radiation dose to more than 90 patients with prostate cancer — and then kept quiet about it.
 - A Florida hospital disclosed that 77 brain cancer patients had received 50 percent more radiation than prescribed because one of the most powerful — and supposedly precise — linear accelerators had been programmed incorrectly for nearly a year.

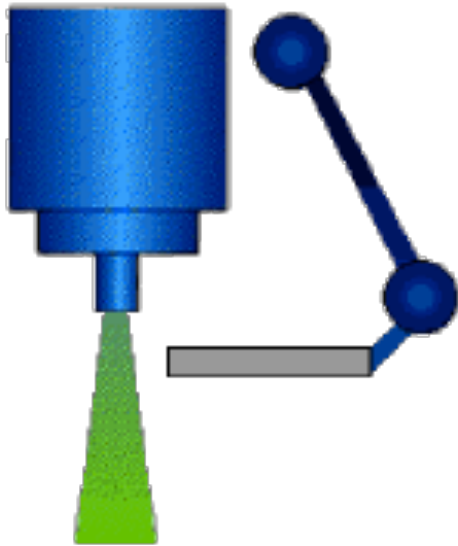


Operation

- The radiation software required that **three essential programming instructions** be saved in sequence:
 - first, the quantity or dose of radiation in the beam;
 - then a digital image of the treatment area; and
 - finally, instructions that guide the multileaf collimator.

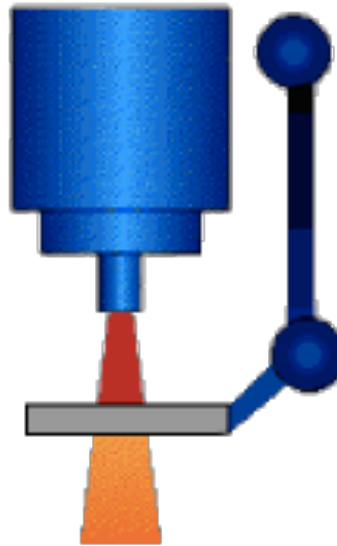
Operation

low current
electron beam
was scanned
across the field



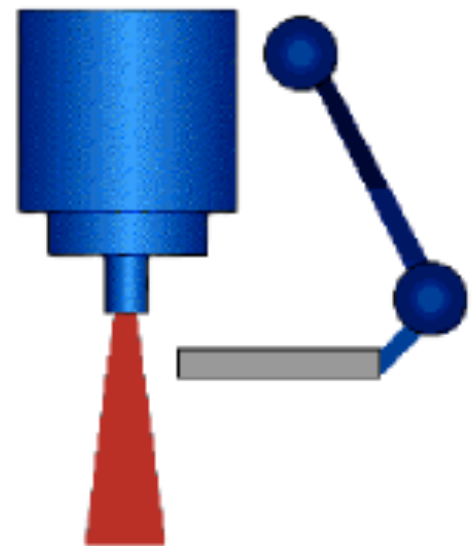
Electron Mode

high current
electron beam
was tracked
at the target



X-Ray Mode

high current
electron beam
with no target
> 'lightning'



THE PROBLEM

What Went Wrong

- When the computer kept crashing, the medical physicist, did not realize that her instructions had not been saved.
- Software errors showing dose was not delivered, technician failed to verify

What Went Wrong

- It was customary — though not mandatory — that the physicist would run a test before the first treatment to make sure that the computer had been programmed correctly. But the hospital had a staffing shortage.

What Went Wrong

- One therapist mistakenly programmed the computer for “wedge out” rather than “wedge in,” as the plan required.
- Another therapist failed to catch the error.
- And the physics staff repeatedly failed to notice it during their weekly checks of treatment records.

What Went Wrong

- AECL focused on fixing individual bugs
- System not designed to be fail-safe
- No devices to report overdoses
- AECL did not communicate fully with customers


Post Mortem

➤ Software lessons

- Difficult to debug programs with concurrent tasks
- Design must be as simple as possible
- Documentation crucial
- Code reuse does not always lead to higher quality

Why Detection is Difficult

- Identifying radiation injuries can be difficult.
- Organ damage and radiation-induced cancer might not surface for years or decades, while underdosing is difficult to detect because there is no injury.
- For these reasons, radiation mishaps seldom result in lawsuits, a barometer of potential problems within an industry.



Dr. Howard I. Amols, chief of clinical physics at Memorial Sloan-Kettering Cancer Center in New York: “Linear accelerators and treatment planning are enormously more complex than 20 years ago. But hospitals are often too trusting of the new computer systems and software, relying on them as if they had been tested over time, when in fact they have not.”

Computerization of Radiation Technology

- Computerization reduced human time needed to calibrate machines and perform safety checks
- But human intervention was still needed to check whether the technology's software came up with a good treatment solution for a patient

People involved in the tragedies

- Company who made the softwares for the accelerometers
- Programmers and testers behind the softwares
- Doctors who prescribed medication
- Staff and technicians who managed the accelerometers

Stakeholders: Class Activity 1

- Split into groups (four groups – company, programmers, doctors, technicians), have each come up with:
 - what was their moral responsibility
 - what each stakeholder did
 - what they didn't do
 - what they could have done differently

Solution: Defensive Design

- Designing for when things go wrong.
- Defensive design is the practice of anticipating all possible ways that an end-user could misuse a device, and designing the device so as to make such misuse impossible, or to minimize the negative consequences.

Automation: Classroom Activity 2

- Most of you wouldn't work with technology with life-critical implications. But automation is pervasive. In the Therac-25 case, automation was in the form of computerization.
- When is automation good?
- When is it not good?
- What checks should be in place to ensure automation is safe and reliable?

Software Reuse: Classroom Activity 3

- Most of you wouldn't work with technology with life-critical implications. But software reuse is pervasive in many applications.
 - When is reuse good?
 - When is it not good?
 - What checks should be in place to ensure reuse is safe and reliable?