Social Implications of Computing

Guest Lecture by Andrew Huang

General Trends between Society and Technology

- Technology usually takes an existing thing and makes it more efficient!
  - Good if it's something beneficial... like mapping the human genome or launching stuff into space.
  - Bad if it's something that was already a problem, like money laundering or surveillance
- Also, conventional law-making bodies tend to lag behind technology.
  - How do you judge these new situations if there aren't even laws for them yet?!
- Covered in CS(H)195

What will this lecture be about?

- Death!
- Death by Software!
- Why is this even a thing?
- What can we do to prevent this situation?

Therac 25

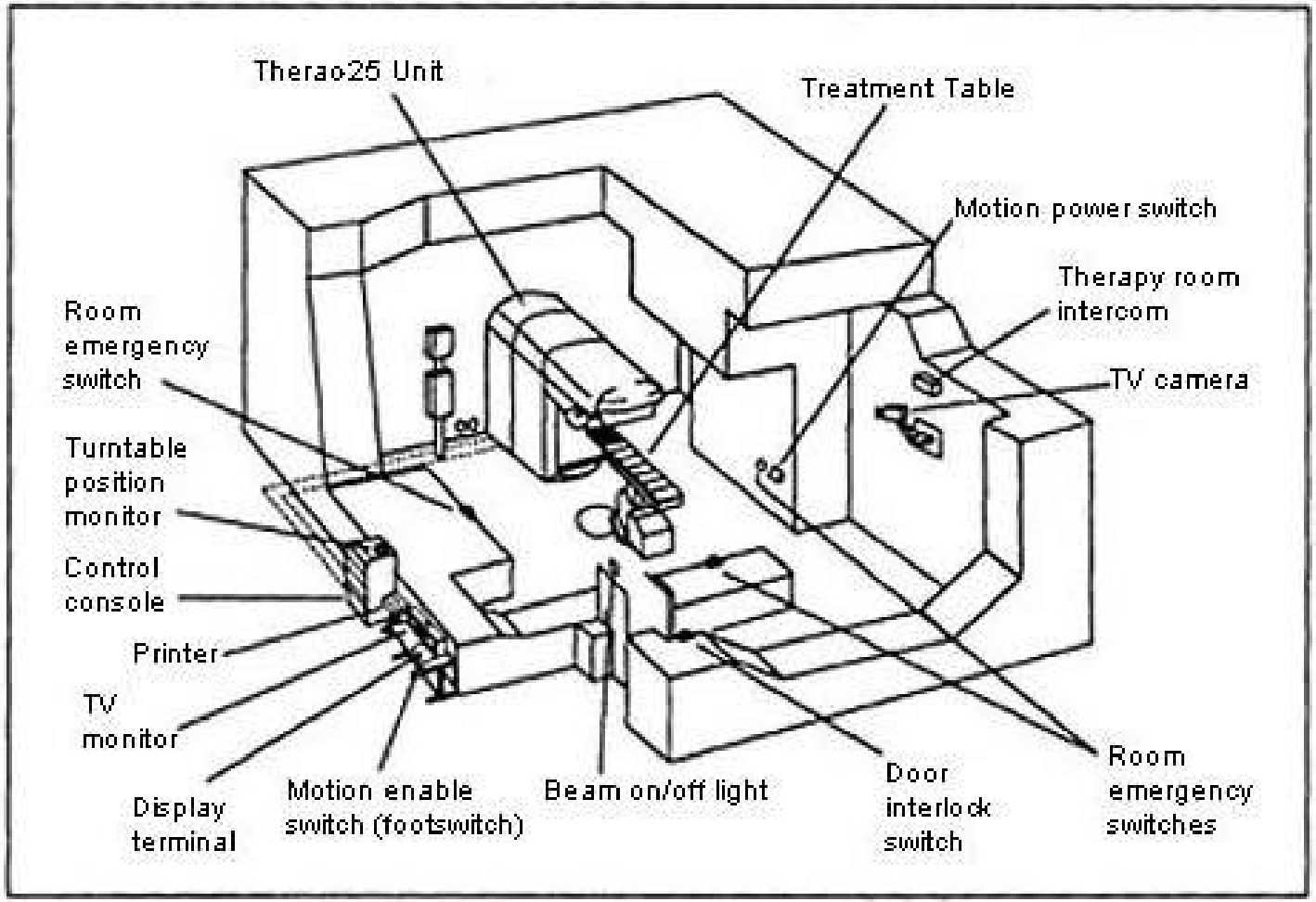

Figure 1. Typical Therac-25 facility

Therac-25

- In theory, A Good Thing... in theory
  - Radiation therapy machine--useful if you have cancer
  - Not so useful if the software is buggy, the hardware has no fail-safes, and the operators don't know what's going on and aren't really paying attention.
- At least four patients died (and 6 serious injuries)
  - When the machine malfunctioned, it spewed out about 100 times as much radiation as it was suppose to.
  - ... So, who's fault was it?

Therac-25 Who's fault is it anyways?

- It's complicated
  - Mistakes were made on ALL the sides:
    - Software guys didn't write clean code.
    - Administrators didn't UPDATE to the new code that was written
    - Hardware didn't have fail-safes
    - Operators saw error messages so often that they started to ignore them all...
  - The scary part:
    - No one had a clue...
    - until people started dying
- In the end, does it even matter?

Therac-25 Who's fault is it anyways?

- Well, yes. We can see what we can do to prevent it:
  - Software can be tested and bulletproofed
    - How to avoid Software "Rot"?
    - How to not underestimate the complexity of your code?
  - Operators can react better:
    - Understand the machines they are running
    - Better user interfaces
  - Hardware safeties:
    - Hardware should never let Software do something STUPID

Therac-25 Software

- Why software is complicated:
  - Hardware can change
  - Other Software can change
  - Environment could change
- In general, it's good to not underestimate the code.
- Solutions?

Therac-25 User Interfaces

[At this point, Andrew switched to the other slides...]

Therac-25 Hardware

- No Interlocks
- There were race conditions in the software (concurrency issue)
  - The hardware had no way to effectively solve these.

What was the point?

- We're not here to tell you "be good", or "eat your veggies".
- Just be aware that code is complicated and easily underestimated. Sometimes, innocent people die because of it.
- [One day, you'll be building something great for other people. Remember your responsibility to them]
- "Just go forward in all your beliefs and prove to me that I am not mistaken in mine."