

Group Project -2 :Therac -25

November 5, 2002

SUMANTH TAMMA
SRIDHAR THIRUVALLUR
ROBERT JAY YOUNG
RAVI K DASARI
JOHN ALPHONSE

The four most critical design issues for the Therac-25 are

1. Hardware Safety (Hardware Safety Module State Diagram)
2. Software Safety (Software Safety Module Sequence Diagram)
3. Module Interactions (Collaboration Diagram for the System)
4. UI and Interrupt Handling (State Chart for the Modified Therac-25 UI)

The four most critical design issues of the system were based on the safety and operational efficiency of the system. We have tried to make the system as safe as possible without too much of a hinderance to the operator. The general operation of the system is as follows. It also briefly explains the functioning of the four critical design issues.

The Operator first enters data at the console in the treatment room, and 'submits' the data. On submitting the data the console UI goes dumb, if he wants to make any changes at this point he will have to reset the UI - with the escape sequence. He then walks over to the control terminal and enters some of the data , like radiation level etc, again. This is just to make sure that he hasnt made any errors while typing the data. Before he can enter data at the terminal the system gets ready to deliver the treatment. While the system is getting ready to deliver the treatment the control terminal is in a dumb state ie . it wont take in any data except the the escape sequence to reset both the UI's. After the data is entered at the terminal the system compares the data entered at the console and the control terminals, if they are the same it proceeds with the treatment. If not the system returns to the console with its values initialized. At this point the operator can abort the treatment with the escape sequence.

During the treatment the Hardware Safety Module ensures that the beam strength is within the safe limit. The hardware safety module samples the beam

every cycle and stops the treatment if the beam strength is in the hazardous range. It also gives data (strength of the beam) to the Software Safety module. The software safety module compares the data with the log and stops the treatment if the data is not equal to the prescribed value, even though the data might be within the safe range. The software safety module also acts as a backup for the hardware module if at all the hardware safety fails.

The Six levels of safety :

Safety is implemented at Six levels in our design of the Therac -25.

1. Simple UI Design
2. Redundant Data Entry
3. Validation of the data
4. Software Safety
5. Hardware Safety
6. Emergency Stop

UI (User Interface) Design: (Better Interrupt Handling capability)

The UI in our system design is modified to ensure better safety. The UI at the console and the terminal has two states - the read state and the dumb state. UI is reset and is in the read state at the console (ie: most of the values are null or empty on the console) while its in the dumb state on the terminal and wont read any data. The working of the UI is explained in the next section. To ensure added safety we added a field to the UI called the submit field. Once the operator enters this field or types in yes, the UI freezes (dunb state), ie no modification can be made to the data at this point. The keyboard interrupt handler will only respond to the escape sequence Ctr-Alt-X at this stage, all the other inputs from the keyboard are ignored. This might not be as user friendly as the original Therac-25 (as the user would have to wait till the machine is set and ready to deliver the treatment before he can modify the data) but eliminates some of the potential risks of the actual design. Once the system is ready, it displays the treatment information like the type of rays, radiation level etc. If the user accepts this information by typing in the right predetermined command at the 'operator command' field the system proceeds with the treatment. The UI is in a dumb state when the system is actually carrying out the treatment or is preparing to do so. In the frozen state the Keyboard handler only reads the escape sequence (emergency shutdown or reset depending on the whether it is actually carrying out the treatment or is preparing to do so). On getting this sequence the UI is reset and the system returns to the initial state.

Redundant DataEntry (Operator Errors)

Our design of the Therac -25 retains the original design of Therac -25 that had two user Interfaces for entering the data. This provides an additional layer of safety. The operator first enters data such as treatment field size and gantry rotation along with the general information at the console in the treatment room. He then returns to the control terminal and enters some of the critical information like treatment time, radiation strength, or whatever data is considered critical. The system then checks these values with those entered at the console, and if these values are not the same it resets both the User Interfaces. We have made sure that the operator has to first enter data into the console in the treatment room and then walks over to the control terminal to finish entering the data as initially only the console interface is in the read state while the terminal interface is in the dumb state. While he is entering data at the console, the terminal is dumb(wont read in any data) and the same is the case with the console when he's entering data into the terminal. The operator can enter data into the control terminal after he has 'submitted' all the data at the console after which he can only reset the console. If the operator wants to edit data he will have to reset the UI and re-enter all the data. This ensures that he doesnt make any mistakes while entering data such as gantry rotation at the control terminal, he would have to go to the console and enter the data again.

Data Validation: (Operator Errors)

Once the operator feeds the data to the system, a data validation module checks if the data give to the system is safe. This module elimintes to a some extent, operator errors. Eg: If the operator feeds in a radiation level of 50,000 rads of X rays, whereas the safe limit fot the X rays is just 5,000rads, this module would give a warning message to the operator.

Software Safety: (Non Fatal Mechanical Errors and Fatal Errors)

This module checks for hardware errors by the system. If the operator enters a radiation level of say 300 rads, but due to some mechanical error the system gives a beam of 500 rads, this module would sample the beam (input from the hardware safety module) compare it with the prescribed radiation strength (from the log) and shutdown the treatment if the actual beam strength is not equal to the prescribed strangth. This can also prevent fatal errors (like radiation above the safe level) even if the Hardware safety module (circuit) fails.

Hardware Safety: (Fatal Errors)

The Hardware safety module consists of a curcuit that samples the beam every cycle. If the beam strength is greater than a certain safety level (which is predetermined to be fatal) it automatically shuts down the system. It also gives

data input to the Software Safety Module. It can be argued that the software safety module provides all of the safety features (plus added features) of this module and Hardware Safety might not be required. This module just checks for fatal errors, and the software safety module is actually a backup for this module and is not meant to replace this module, and this would be the first module to respond to the fatal errors.

Emergency Stop:

There are two emergency stops. One is the software emergency stop which is Ctr-Atl-X or some safety sequence and a emergency shutdown switch. Both stop the treatment. These can be used if due to some reason the operator wants to stop the treatment. Eg If the operator noticed he has entered a wrong (but non fatal) radiation value.