Medical Devices: The Therac-25*

Nancy Leveson
University of Washington

1 Introduction

Between June 1985 and January 1987, a computer-controlled radiation therapy machine, called the Therac-25, massively overdosed six people. These accidents have been described as the worst in the 35-year history of medical accelerators [6].

A detailed accident investigation, drawn from publicly available documents, can be found in Leveson and Turner [4]. The following account is taken from this report and includes both the factors involved in the overdoses themselves and the attempts by the users, manufacturers, and governments to deal with them. Because this accident was never officially investigated, some information on the Therac-25 software development, management, and quality control procedures are not available. What is included below has been gleaned from law suits and depositions, government records, and copies of correspondence and other material obtained from the U.S. Food and Drug Administration (FDA), which regulates these devices.

2 Background

Medical linear accelerators (linacs) accelerate electrons to create high-energy beams that can destroy tumors with minimal impact on the surrounding

*This appendix is taken from Nancy Leveson, *Safeware: System Safety and Computers*, Addison-Wesley, 1995.

healthy tissue. Relatively shallow tissue is treated with the accelerated electrons; to reach deeper tissue, the electron beam is converted into X-ray photons.

In the early 1970s, Atomic Energy of Canada Limited (AECL)[1] and a French company called CGR went into business together building linear accelerators. The products of this cooperation were (1) the Therac-6, a 6 million electron volt (MeV) accelerator capable of producing X-rays only and later (2) the Therac-20, a 20 MeV, dual-mode (X-rays or electrons) accelerator. Both were versions of older CGR machines, the Neptune and Sagittaire, respectively, which were augmented with computer control using a DEC PDP-11 minicomputer. We know that some of the old Therac-6 software routines were reused in the Therac-20 and that CGR developed the initial software.

Software functionality was limited in both machines: The computer merely added convenience to the existing hardware, which was capable of standing alone. Industry-standard hardware safety features and interlocks in the underlying machines were retained.

The business relationship between AECL and CGR faltered after the Therac-20 effort. Citing competitive pressures, the two companies did not renew their cooperative agreement when scheduled in 1981.

In the mid-1970s, AECL had developed a radical new "double pass" concept for electron acceleration. A double-pass accelerator needs much less space to develop comparable energy levels because it folds the long physical mechanism required to accelerate the electrons, and it is more economical to produce. Using this double-pass concept, AECL designed the Therac-25, a dual-mode linear accelerator that can deliver either photons at 25 MeV or electrons at various energy levels.

Compared with the Therac-20, the Therac-25 is notably more compact, more versatile, and arguably easier to use. The higher energy takes advantage of the phenomenon of *depth dose*: As the energy increases, the depth in the body at which maximum dose build-up occurs also increases, sparing the tissue above the target area. Economic advantages also come into play for the customer, since only one machine is required for both treatment modalities

[1] AECL was an arms-length entity, called a crown corporation, of the Canadian government. Since the time of the incidents related in this paper, AECL Medical, a division of AECL, was privatized and is now called Theratronics International, Ltd. Currently, the primary business of AECL is the design and installation of nuclear reactors.

(electrons and photons).

Several features of the Therac-25 are important in understanding the accidents. First, like the Therac-6 and the Therac-20, the Therac-25 is controlled by a PDP-11 computer. However, AECL designed the Therac-25 to take advantage of computer control from the outset; they did not build on a stand-alone machine. The Therac-6 and Therac-20 had been designed around machines that already had histories of clinical use without computer control.

In addition, the Therac-25 software has more responsibility for maintaining safety than the software in the previous machines. The Therac-20 has independent protective circuits for monitoring the electron-beam scanning plus mechanical interlocks for policing the machine and ensuring safe operation. The Therac-25 relies more on software for these functions. AECL took advantage of the computer's abilities to control and monitor the hardware and decided not to duplicate all the existing hardware safety mechanisms and interlocks.

Some software for the machines was interrelated or reused. In a letter to a Therac-25 user, the AECL quality assurance manager said, "The same Therac-6 package was used by the AECL software people when they started the Therac-25 software. The Therac-20 and Therac-25 software programs were done independently starting from a common base" [4]. The reuse of Therac-6 design features or modules may explain some of the problematic aspects of the Therac-25 software design. The quality assurance manager was apparently unaware that some Therac-20 routines were also used in the Therac-25; this was discovered after a bug related to one of the Therac-25 accidents was found in the Therac-20 software.

AECL produced the first hardwired prototype of the Therac-25 in 1976, and the completely computer-controlled commercial version was available in late 1982.

Turntable Positioning. The Therac-25 turntable design plays an important role in the accidents. The upper turntable (see Figure 1) rotates accessory equipment into the beam path to produce two therapeutic modes: electron mode and photon mode. A third position (called the field light position) involves no beam at all, but rather is used to facilitate correct positioning of the patient. Because the accessories appropriate to each mode

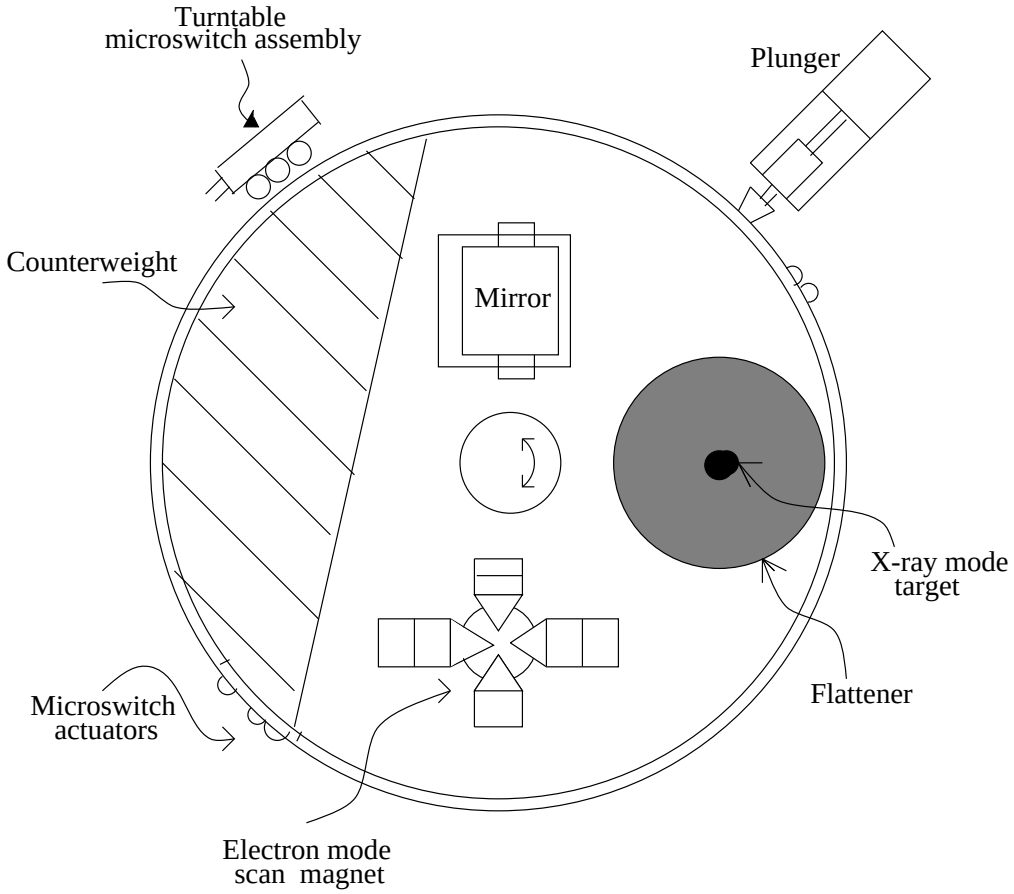

Figure 1: Upper turntable assembly.

are physically attached to the turntable, proper operation of the Therac-25 is heavily dependent on the turntable position, which is monitored by three microswitches.

The raw, highly concentrated accelerator beam is dangerous to living tissue. In electron therapy, the computer controls the beam energy (from 5 to 25 MeV) and current, while scanning magnets are used to spread the beam to a safe, therapeutic concentration. These scanning magnets are mounted on the turntable and moved into proper position by the computer. Similarly, an ion chamber to measure electrons is mounted on the turntable and also moved into position by the computer. In addition, operator-mounted electron trimmers can be used to shape the beam if necessary.

For X-ray (or photon) therapy, only one energy level is available: 25 MeV. Much greater electron-beam current is required for X-ray mode (some 100 times greater than that for electron therapy) [6] to produce comparable output. Such a high dose-rate capability is required because a "beam flattener" is used to produce a uniform treatment field. This flattener, which resembles an inverted ice cream cone, is a very efficient attenuator; thus, to get a reasonable treatment dose rate out of the flattener, a very high input dose rate is required. If the machine should produce a photon beam with the beam flattener not in position, a high output dose to the patient results. This is the basic hazard of dual-mode machines: If the turntable is in the wrong position, the beam flattener will not be in place.

In the Therac-25, the computer is responsible for positioning the turntable (and for checking the turntable position) so that a target, flattening filter, and X-ray ion chamber are directly in the beam path. With the target in place, electron bombardment produces X-rays. The X-ray beam is shaped by the flattening filter and measured by the X-ray ion chamber.

No accelerator beam is expected in the third or field light turntable position. A stainless steel mirror is placed in the beam path and a light simulates the beam. This lets the operator see precisely where the beam will strike the patient and make necessary adjustments before treatment starts. There is no ion chamber in place at this turntable position, since no beam is expected.

Traditionally, electromechanical interlocks have been used on these types of equipment to ensure safety — in this case, to ensure that the turntable and attached equipment are in the correct position when treatment is started. In the Therac-25, software checks were substituted for many of the traditional hardware interlocks.

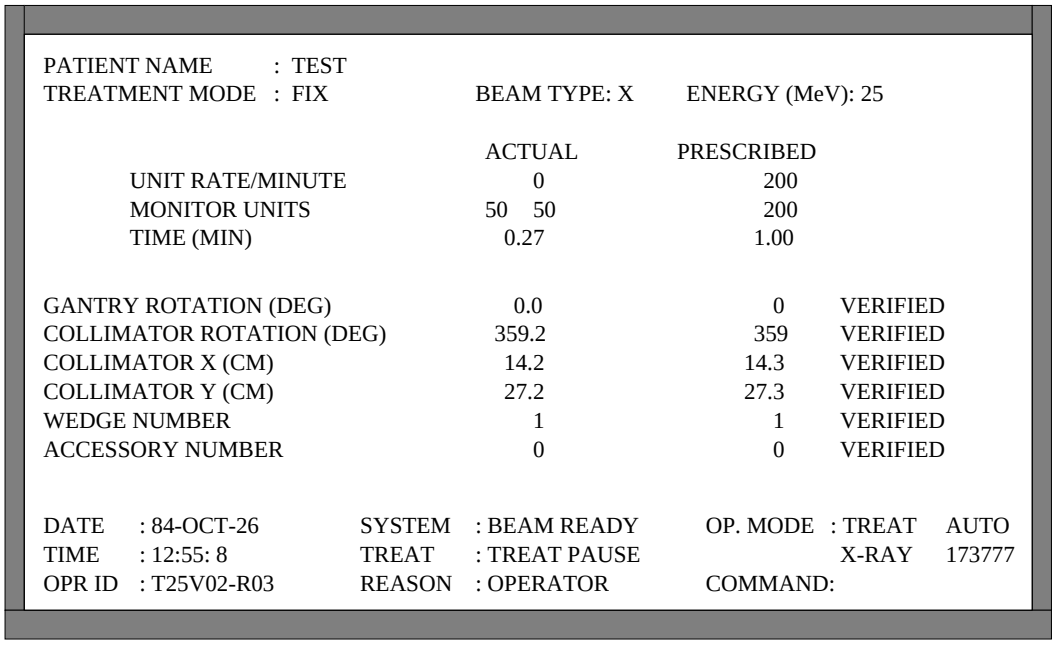

Figure 2: Operator interface screen layout.

The Operator Interface. The description of the operator interface here applies to the version of the software used during the accidents. Changes made as a result of an FDA recall are described later.

The Therac-25 operator controls the machine through a DEC VT100 terminal. In the general case, the operator positions the patient on the treatment table, manually sets the treatment field sizes and gantry rotation, and attaches accessories to the machine. Leaving the treatment room, the operator returns to the console to enter the patient identification, treatment prescription (including mode or beam type, energy level, dose, dose rate, and time), field sizing, gantry rotation, and accessory data. The system then compares the manually set values with those entered at the console. If they match, a *verified* message is displayed and treatment is permitted. If they do not match, treatment is not allowed to proceed until the mismatch is corrected. Figure 2 shows the screen layout.

When the system was first built, operators complained that it took too

long to enter the treatment plan. In response, AECL modified the software before the first unit was installed: Instead of reentering the data at the keyboard, operators could simply use a carriage return to copy the treatment site data [5]. A quick series of carriage returns would thus complete the data entry. This modification was to figure in several of the accidents.

The Therac-25 could shut down in two ways after it detected an error condition. One was a *treatment suspend*, which required a complete machine reset to restart. The other, not so serious, was a *treatment pause*, which only required a single key command to restart the machine. If a *treatment pause* occurred, the operator could press the Ⓟ key to "proceed" and resume treatment quickly and conveniently. The previous treatment parameters remained in effect, and no reset was required. This feature could be invoked a maximum of five times before the machine automatically suspended treatment and required the operator to perform a system reset.

Error messages provided to the operator were cryptic, and some merely consisted of the word MALFUNCTION followed by a number from 1 to 64 denoting an analog/digital channel number. According to an FDA memorandum written after one accident:

> The operator's manual supplied with the machine does not explain nor even address the malfunction codes. The Maintance [sic] Manual lists the various malfunction numbers but gives no explanation. The materials provided give no indication that these malfunctions could place a patient at risk.
>
> The program does not advise the operator if a situation exists wherein the ion chambers used to monitor the patient are saturated, thus are beyond the measurement limits of the instrument. This software package does not appear to contain a safety system to prevent parameters being entered and intermixed that would result in excessive radiation being delivered to the patient under treatment.

An operator involved in one of the accidents testified that she had become insensitive to machine malfunctions. Malfunction messages were commonplace and most did not involve patient safety. Service technicians would fix the problems or the hospital physicist would realign the machine and make it operable again. She said,

> "It was not out of the ordinary for something to stop the machine.... It would often give a low dose rate in which you would turn the machine back on. ... They would give messages of low dose rate, V-tilt, H-tilt, and other things; I can't remember all the reasons it would stop, but there was a lot of them."

A radiation therapist at another clinic reported that an average of 40 dose-rate malfunctions, attributed to underdoses, occurred on some days.

The operator further testified that during instruction she had been taught that there were "so many safety mechanisms" that she understood it was virtually impossible to overdose a patient.

Hazard Analysis. In March 1983, AECL performed a safety analysis on the Therac-25. This analysis was in the form of a fault tree and apparently excluded the software. According to the final report, the analysis made several assumptions about the computer and its software:

1. Programming errors have been reduced by extensive testing on a hardware simulator and under field conditions on teletherapy units. Any residual software errors are not included in the analysis.
2. Program software does not degrade due to wear, fatigue, or reproduction process.
3. Computer execution errors are caused by faulty hardware components and by "soft" (random) errors induced by alpha particles and electromagnetic noise.

The fault tree resulting from this analysis does appear to include computer failure, although apparently, judging from the basic assumptions above, it considers hardware failures only. For example, in one OR gate leading to the event of getting the wrong energy, a box contains "Computer selects wrong energy," and a probability of 10^{-11} is assigned to this event. For "Computer selects wrong mode," a probability of 4×10^{-9} is given. The report provides no justification of either number.

Tyler Accidents Fault Description

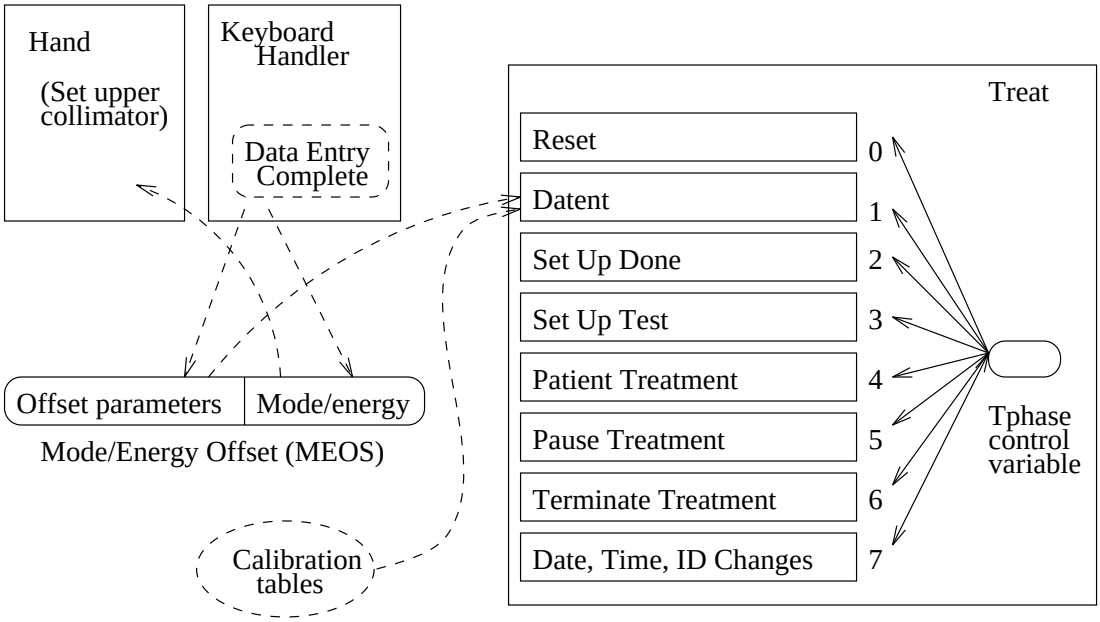

Figure 3: Tasks and subroutines in the code blamed for the Tyler accidents.

The treatment monitor task (Treat) controls the various phases of treatment by executing its eight subroutines. The treatment phase indicator variable (Tphase) is used to determine which subroutine should be executed (Figure 3). Following the execution of a particular subroutine, Treat reschedules itself.

One of Treat's subroutines, called Datent (data entry), communicates with the keyboard handler task (a task that runs concurrently with Treat) via a shared variable (Data Entry Complete flag) to determine whether the prescription data has been entered. The keyboard handler recognizes the completion of data entry and changes the Data Entry Complete variable to denote this. Once this variable is set, the Datent subroutine detects the variable's change in status and changes the value of Tphase from 1 (Datent) to 3 (Set Up Test). In this case, the Datent subroutine exits back to the Treat subroutine, which will reschedule itself and begin execution of the Set Up Test subroutine. If the Data Entry Complete variable has not been set, Datent leaves the value of Tphase unchanged and exits back to Treat's

mainline. Treat will then reschedule itself, essentially rescheduling the Datent subroutine.

The command line at the lower right-hand corner of the screen (see Figure 2) is the cursor's normal position when the operator has completed all the necessary changes to the prescription. Prescription editing is signified by moving the cursor off the command line. As the program was originally designed, the Data Entry Complete variable by itself is not sufficient because it does not ensure that the cursor is located on the command line; under the right circumstances, the data entry phase can be exited before all edit changes are made on the screen.

The keyboard handler parses the mode and energy level specified by the operator and places an encoded result in another shared variable, the 2-byte Mode/Energy Offset variable (MEOS). The low-order byte of this variable is used by another task (Hand) to set the collimator/turntable to the proper position for the selected mode and energy. The high-order byte of the MEOS variable is used by Datent to set several operating parameters.

Initially, the data-entry process forces the operator to enter the mode and energy except when the photon mode is selected, in which case the energy defaults to 25 MeV. The operator can later edit the mode and energy separately. If the keyboard handler sets the Data Entry Complete flag before the operator changes the data in MEOS, Datent will not detect the changes because it has already exited and will not be reentered again. The upper collimator (turntable), on the other hand, is set to the position dictated by the low-order byte of MEOS by another concurrently running task (Hand) and can therefore be inconsistent with the parameters set in accordance with the information in the high-order byte. The software appears to contain no checks to detect such an incompatibility.

The first thing Datent does when it is entered is to check whether the keyboard handler has set the mode and energy in MEOS. If so, it uses the high-order byte to index into a table of preset operating parameters and places them in the digital-to-analog output table. The contents of this output table are transferred to the digital-to-analog converter during the next clock cycle. Once the parameters are all set, Datent calls the subroutine Magnet, which sets the bending magnets. The following shows a simplified pseudocode description of relevant parts of the software:

Datent:

```
        if mode/energy specified then
            begin
                calculate table index
                repeat
                    fetch parameter
                    output parameter
                    point to next parameter
                until all parameters set
                call Magnet
                if mode/energy changed then return
            end
        if data entry is complete then set Tphase to 3
        if data entry is not complete then
            if reset command entered then set Tphase to 0
        return

    Magnet:
        Set bending magnet flag
        repeat
            Set next magnet
            call Ptime
            if mode/energy has changed, then exit
        until all magnets are set
        return

    Ptime:
        repeat
            if bending magnet flag is set then
                if editing taking place then
                    if mode/energy has changed then exit
        until hysteresis delay has expired
        Clear bending magnet flag
        return
```

Setting the bending magnets takes about eight seconds. Magnet calls a subroutine called Ptime to introduce a time delay. Since several magnets need

to be set, Ptime is entered and exited several times. A flag to indicate that the bending magnets are being set is initialized upon entry to the Magnet subroutine and cleared at the end of Ptime. Furthermore, Ptime checks a shared variable, set by the keyboard handler, that indicates the presence of any editing requests. If there are edits, then Ptime clears the bending magnet variable and exits to Magnet, which then exits to Datent. But the edit change variable is checked by Ptime only if the bending magnet flag is set. Because Ptime clears it during its first execution, any edits performed during each succeeding pass through Ptime will not be recognized. Thus, an edit change of the mode or energy, although reflected on the operator's screen and the mode/energy offset variable, will not be sensed by Datent so it can index the appropriate calibration tables for the machine parameters.

Recall that the Tyler error occurred when the operator made an entry indicating the mode and energy, went to the command line, then moved the cursor up to change the mode or energy and returned to the command line all within eight seconds. Because the magnet setting takes about eight seconds and Magnet does not recognize edits after the first execution of Ptime, the editing had been completed by the return to Datent, which never detected that it had occurred. Part of the problem was fixed after the accident by clearing the bending magnet variable at the end of Magnet (after *all* the magnets have been set) instead of at the end of Ptime.

But this is not the only problem. Upon exit from the Magnet subroutine, the data entry subroutine (Datent) checks the Data Entry Complete variable. If it indicates that data entry is complete, Datent sets Tphase to 3 and Datent is not entered again. If it is not set, Datent leaves Tphase unchanged, which means it will eventually be rescheduled. But the Data Entry Complete variable only indicates that the cursor has been down to the command line, not that it is still there. A potential race condition is set up. To fix this, AECL introduced another shared variable controlled by the keyboard handler task that indicates the cursor is not positioned on the command line. If this variable is set, then prescription entry is still in progress and the value of Tphase is left unchanged.

3.5.4 The Government and User Response

The FDA does not approve each new medical device on the market: All medical devices go through a classification process that determines the level

(but which had not yet been installed) would have prevented the Yakima accident.

The patient died in April from complications related to the overdose. He had a terminal form of cancer, but a lawsuit was initiated by his survivors alleging that he died sooner than he would have and endured unnecessary pain and suffering due to the radiation overdose. The suit, like all the others, was settled out of court.

3.6.1 The Yakima Software "Bug"

The software problem for the second Yakima accident is fairly well-established and different from that implicated in the Tyler accidents. There is no way to determine what particular software design errors were related to the Kennestone, Hamilton, and first Yakima accidents. Given the unsafe programming practices exhibited in the code, unknown race conditions or errors could have been responsible for them. There is speculation, however, that the Hamilton accident was the same as this second Yakima overdose. In a report of a conference call on January 26, 1987, between the AECL quality assurance manager and Ed Miller of the FDA discussing the Yakima accident, Miller notes

> This situation probably occurred in the Hamilton, Ontario accident a couple of years ago. It was not discovered at that time and the cause was attributed to intermittent interlock failure. The subsequent recall of the multiple microswitch logic network did not really solve the problem.

The second Yakima accident was again attributed to a type of race condition in the software — this one allowed the device to be activated in an error setting (a "failure" of a software interlock). The Tyler accidents were related to problems in the data-entry routines that allowed the code to proceed to Set Up Test before the full prescription had been entered and acted upon. The Yakima accident involved problems encountered later in the logic after the treatment monitor Treat reaches Set Up Test.

The Therac-25's field light feature allows very precise positioning of the patient for treatment. The operator can control the machine right at the treatment site using a small hand control that offers certain limited functions for patient setup, including setting gantry, collimator, and table motions.

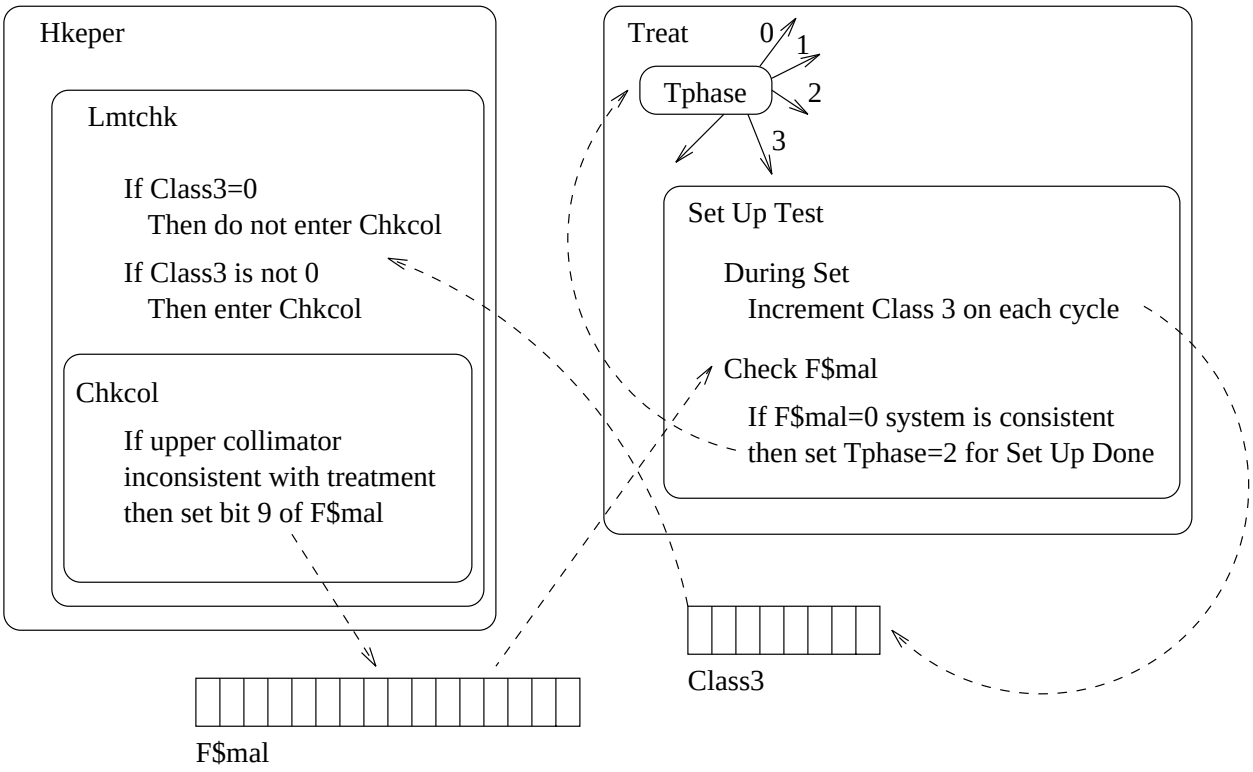

Figure 4: The Yakima software flaw.

Normally, the operator enters all the prescription data at the console (outside the treatment room) before the final setup of all machine parameters is completed in the treatment room. This gives rise to an UNVERIFIED condition at the console. The operator then completes patient setup in the treatment room, and all relevant parameters now VERIFY. The console displays a message to PRESS SET BUTTON while the turntable is in the field light position. The operator now presses the *set* button on the hand control or types "set" at the console. That should set the collimator to the proper position for treatment.

In the software, after the prescription is entered and verified by the Datent routine, the control variable Tphase is changed so that the Set Up Test routine is entered (Figure 4). Every pass through the Set Up Test rou-

tine increments the upper collimator position check, a shared variable called Class3. If Class3 is nonzero, there is an inconsistency and treatment should not proceed. A zero value for Class3 indicates that the relevant parameters are consistent with treatment, and the software does not inhibit the beam.

After setting the Class3 variable, Set Up Test next checks for any malfunctions in the system by checking another shared variable (set by a routine that actually handles the interlock checking) called F$mal to see if it has a nonzero value. A nonzero value in F$mal indicates that the machine is not ready for treatment, and the Set Up Test subroutine is rescheduled. When F$mal is zero (indicating that everything is ready for treatment), the Set Up Test subroutine sets the Tphase variable equal to 2, which results in next scheduling the Set Up Done subroutine and the treatment is allowed to continue.

The actual interlock checking is performed by a concurrent Housekeeper task (Hkeper). The upper collimator position check is performed by a subroutine of Hkeper called Lmtchk (analog-to-digital limit checking). Lmtchk first checks the Class3 variable. If Class3 contains a non-zero value, Lmtchk calls the Check Collimator (Chkcol) subroutine. If Class3 contains zero, Chkcol is bypassed and the upper collimator position check is not performed. The Chkcol subroutine sets or resets bit 9 of the F$mal shared variable, depending on the position of the upper collimator—which in turn is checked by the Set Up Test subroutine of Treat to decide whether to reschedule itself or to proceed to Set Up Done.

During machine setup, Set Up Test will be executed several hundred times because it reschedules itself waiting for other events to occur. In the code, the Class3 variable is incremented by one in each pass through Set Up Test. Since the Class3 variable is one byte, it can only contain a maximum value of 255 decimal. Thus, on every 256th pass through the Set Up Test code, the variable will overflow and have a zero value. That means that on every 256th pass through Set Up Test, the upper collimator will not be checked and an upper collimator fault will not be detected.

The overexposure occurred when the operator hit the "set" button at the precise moment that Class3 rolled over to zero. Thus, Chkcol was not executed and F$mal was not set to indicate that the upper collimator was still in the field-light position. The software turned on the full 25 MeV without the target in place and without scanning. A highly concentrated electron beam resulted, which was scattered and deflected by the stainless

steel mirror that was in the path.

The technical "fix" implemented for this particular software flaw is described by AECL as simple: the program is changed so that the Class3 variable is set to some fixed nonzero value each time through Set Up Test instead of being incremented.

3.6.2 Manufacturer, Government, and User Response

On February 3, 1987, after interaction with the FDA and others, including the user's group, AECL announced to its customers

1. A new software release to correct both the Tyler and Yakima software problems
2. A hardware single-pulse shutdown circuit
3. A turntable potentiometer to independently monitor turntable position
4. A hardware turntable interlock circuit

The second item, a hardware single-pulse shutdown circuit, essentially acts as a hardware interlock to prevent overdosing by detecting an unsafe level of radiation and halting beam output after one pulse of high energy and current. This interlock effectively provides an independent way to protect against a wide range of potential hardware failures and software errors. The third item, a turntable potentiometer, was the safety device recommended by several groups after the Hamilton accident.

After the second Yakima accident, the FDA became concerned that the use of the Therac-25 during the CAP process, even with AECL's interim operating instructions, involved too much risk to patients. The FDA concluded that the accidents demonstrated that the software alone could not be relied upon to assure safe operation of the machine. In a February 18, 1987, internal FDA memorandum, the Director of the Division of Radiological Products wrote:

> It is impossible for CDRH to find all potential failure modes and conditions of the software. AECL has indicated the "simple software fix" will correct the turntable position problem displayed at Yakima. We have not yet had the opportunity to evaluate that modification. Even if it does, based upon past history, I am not