4.5.2020

Rinnakkaisuus, projekti 3

Therac-25

Therac-25 on sädehoitolaite, joka kykenee tuottamaan usean eri energian elektroni- ja röntgensäteilyä. Elektronisäteily voidaan valita liukuvasti välillä 5-25 MeV, mutta röntgensäteilylle on saatavilla ainoastaan 25 MeV energiataso. Therac-25 aiheutti kuusi eri tapaturmaa, joissa potilaalle annettu säteilyannos oli paljon suurempi kuin laitteen käyttäjän laitteelle antamat hoitoparametrit. Todennäköisesti lähes kaikki tapaturmat ovat laitteeseen jääneiden ohjelmointivirheiden aiheuttamia.

Toisaalta ongelma oli myös Therac-25 osittain kierrättämä aikaisemman Therac-6:n ohjelmisto. Myös Therac-20:n laitteisto oli tehty sen pohjalta, mutta se Therac-6:n tavoin sisälsi Therac-25:sta poiketen erillisen järjestelmän säteilytasojen tarkkailuun sekä mekaanisia järjestelmälukituksia, joiden ansiosta osa ohjelmointivirheistä pysyi piilossa aiheuttamatta ongelmia laitteen käytössä. Therac-25:ssa lukkojen puuttuessa koko valvonta on pelkästään ohjelmiston vastuulla, minkä takia nämä virheet pääsivät aiheuttamaan kriittisiä seurauksia.

Kaksi kuudesta Therac-25:n aiheuttamasta tapaturmasta liittyivät samaan rinnakkaisen ohjelmoinnin virheeseen. Nämä kaksi potilaan kuolemaan johtanutta tapaturmaa tapahtuivat Tylerissä East Texas Cancer Center -sairaalassa. Ohjelmointivirhe oli jäänyt laitteeseen, sillä se ilmeni vain, jos käyttäjä syötti tietyn näppäinyhdistelmän tietyn ajan sisällä.

Toinen rinnakkaisen ohjelmoinnin virhe aiheutti kuolemaan johtaneen tapaturman vuonna 1987 Yakima Valley Memorial hospital -sairaalassa. On todennäköistä, että sama ohjelmointivirhe olisi ollut syypää myös vähintään yhteen toiseen Therac-25:n aiheuttamaan tapaturmaan.

Luulisin, että osaisimme korjata rinnakkaisuuteen liittyvät ohjelmointivirheet kurssin aikana opittuja asioita hyödyntämällä. En kuitenkaan antaisi Therac-25:n hoitaa minua, ainakaan jos minulla olisi vaihtoehtoja, sillä Therac-25:n koko ohjelmisto on mielestäni vähintäänkin kyseenalainen. Ensinnäkin ohjelmisto on suurimmaksi osaksi kirjoitettu alun perin toiselle laitteelle. Lisäksi ohjelmiston kehityksessä on laiminlyöty lähes kokonaan muun muassa laadunvalvonta, käytettävyys, testaaminen ja dokumentointi.

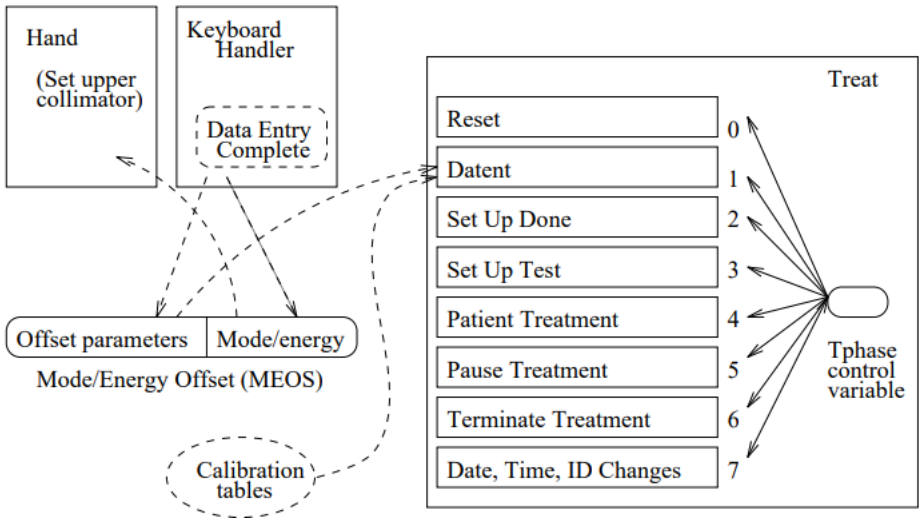
Jos Therac-25:n tekeminen turvalliseksi olisi minun vastuullani ja jos aika ja raha ei minua rajoittaisi, kirjoituttaisin koko ohjelmiston uudelleen. Tällä kertaa Therac-25:n ohjelmisto kirjoitettaisiin suoraan sille laitteistolle, jossa sitä käytetään. Ohjelmistoa ja laitetta testattaisiin myös ankarasti. Ohjelmistokehityksessä pidettäisiin huoli myös muusta laadunvalvonnasta, käytettävyydestä ja dokumentoinnista.

Ohjelmointivirhe, joka aiheutti East Texas Cancer Centerissä sattuneet tapaturmat

Therac-25:ssä on yhteensä 10 eri tehtävää (Task), joita suoritetaan yksi kerrallaan ja joiden suoritusjärjestyksen päättää laitteen vuorontaja. Yksi näistä tehtävistä on *Treat*, joka ohjaa ja valvoo

koko hoito-operaation kulkua. *Treat* koostuu kahdeksasta eri vaiheesta. Suoritettavan vaiheen määrää *Tphase*-muuttuja.

Treat on vuorovaikutuksesssa *Keyboard Handler*-tehtävän kanssa, jonka vastuulla on tiedonvälittäminen käyttäjältä ohjelmistolle. *Keyboard Handleriä* kutsutaan keskeytyksenä, kun käyttäjä tekee muutoksia hoitoparametreihin käyttöliittymän kautta. Kun käyttäjä muokkaa hoitoparametreja, *Keyboard Handler* tallentaa muutokset *MEOS*-muuttujaan, joka ylläpitää tietoa mm. valitusta hoitomuodosta ja säteilytehosta.



Kuva 1. East Texas Cancer Centerin tapaturmiin liittyvät tehtävät ja muuttujat. Hand on tehtävä, jonka vastuulla on osa laitteiston asetusten asettamisesta. Hand toimii Treatista riippumatta, jolloin Handin asettamat asetukset saattoivat olla yhteensopimattomia Datentin asettamien kanssa.

Tieto siitä, onko hoitoparametrien syöttäminen valmista, välittyy *Data Entry Complete* -muuttujan kautta *Treat*-tehtävälle. *Data Entry Complete* -muuttujan arvo riippuu käyttäjän kursorin sijainnista näytöllä. Jos kursori on *Command*-rivillä, muuttujan arvo on tosi.

Ensimmäinen *Treat*-tehtävän vaihe on *Datent* (data entry). Jos käyttäjä on asettanut *MEOS*-muuttujan arvot, *Datent* vie muuttujan tiedot laitteistolle ja osa laitteistosta asetetaan muuttujan mukaisesti. Tämän jälkeen *Datent* tarkistaa *Data Entry Complete* -muuttujan arvon. Jos arvo on epätosi, *Tphase*-muuttujan arvo ei muutu, jolloin seuraavan kerran *Treat*-tehtävää suorittaessa suoritetaan *Datent*-vaihe. Jos *Data Entry Complete* -muuttujan arvo on tosi, *Tphase*-muuttujan arvoksi asetetaan 3, jolloin seuraavan kerran *Treatia* suorittaessa suoritetaan *Set Up Test* ja *Treat*-tehtävä etenee.

Ongelman aiheutti se, että laitteiston asetusten asettaminen, tarkemmin sanoen magneettien asettaminen, vei noin kahdeksan sekuntia. Magneettien asettamisen aikana nopea ja kokenut laitteen käyttäjä ehti muuttamaan hoitoparametreja ja asettamaan kursorin *Command*-riville. Tämä aiheutti sen, että osa laitteistosta asetettiin muokkausta edeltävien, eli väärien, hoitoparametrien mukaisesti, vaikka käyttäjän näytöllä näkyivät oikeat hoitoparametrit. Koska kursori oli *Command*-rivillä, *Treat*-tehtävä ei enää suorittanut tämän jälkeen *Datent*-vaihetta ja väärät asetukset jäivät voimaan. *Datent*-vaiheessa oli tarkistus, joka tarkisti, onko käyttäjä tehnyt muutoksia laitteiston asettamisen aikana, mutta se ei toiminut ohjelmointivirheen takia.

4.5.2020

Ohjelmointivirhe, joka aiheutti Yakima Valley Memorial Hospital -sairaalan vuoden 1987 tapaturman

Therac-25:ssa oleva työvalo-ominaisuus mahdollistaa potilaan tarkan asettelemisen hoitoa varten. Laitteen käyttäjä pystyy hallitsemaan laitteen portaalia, kollimaattoria sekä pöydän liikkeitä käsiohjauksella suoraan hoitopaikalla.

Tavallisesti hoito-ohjeet asetetaan laitteen konsoliin ennen kuin lopulliset säädöt tehdään itse hoitohuoneessa. Tämä luo konsoliin *"unverified"* tilan, jonka jälkeen käyttäjä tekee lopulliset säädöt ja parametrit varmistuvat. Tämän jälkeen käyttäjä painaa "set" -nappia tai kirjoittaa komennon konsoliin, jolloin kollimaattorin pitäisi asettua oikeaan asentoon hoitoa varten.

Kun hoito-ohjeet on asetettu ja varmistettu Datent-vaiheen toimesta, muuttuja *Tphase* muutetaan niin, että *Set Up Test* etenee. Jokainen kierros *Set Up Test* suorituksesta kasvattaa ylemmän kollimaattorin sijaintitarkistusta eli muuttujaa *Class3.* Mikäli tämän muuttujan arvo on nolla, parametrit ovat oikein ja hoito voi jatkua, mutta muuttujan arvon ollessa jokin muu hoidossa on ristiriitaisuuksia. *Set Up Test* tarkistaa hoidon oikeellisuuden myös toisella muuttujalla *F$mal,* jonka ollessa nolla hoitoa voidaan jatkaa.

Ohjelmisto käyttää rinnakkaista *Housekeeper*-ohjelmaa. Tämän aliohjelma *Lmtchk* tarkistaa ensin muuttujan *Class3* arvon, jonka ollessa nollasta poikkeava kutsutaan aliohjelmaa *Check Collimator,* jolla kollimaattorin asento tarkistetaan. *Class3* muuttujan ollessa nolla, kollimaattorin tarkistus ohitetaan.

Laitteen säädösten aikana *Set Up Test* suoritetaan satoja kertoja ja jokainen kerta kasvattaa yhdellä muuttujaa *Class3.* Koska *Class3* on yhden tavun kokoinen, sen maksimikoko on 255 desimaalia. Tästä johtuen joka 256. *Set Up Test* kierros aiheutti ylivuodon, jolloin *Class3* oli arvoltaan nolla. Tämä taas tarkoittaa sitä, että kollimaattorin sijaintia ei tarkisteta ja mahdollinen virhe jää huomaamatta.

Laitteen käyttäjän painaessa "set"-nappia juuri, kun *Class3* vaihtui nollaksi, yliannostus pääsi tapahtumaan. Koska *Class3* oli arvoltaan nolla, aliohjelmaa *Check Collimator* ei suoritettu ja tästä johtuen muuttuja *F$mal* ei ilmoittanut, että ylempi kollimaattori oli edelleen työvalotilassa.

Muut tapaturmat eivät (todistetusti) liittyneet rinnakkaisuuteen.

Korjaukset rinnakkaisuuden näkökulmasta

Niin *East Texas Cancer Centerin* kuin *Yakima Valley Memorial Hospitalin* tapauksissa on kyse toisaalta hyvin yksinkertaisista rinnakkaisuuden ongelmista. Ensimmäisessä käyttäjä tekee muutoksia, joita osa ohjelmistosta hallinnoi. Tämä osa kuitenkin muuttaa samoja muistialueita, joita toinen koneen toimintaa ohjaava osa lukee, jolloin muodostuvan kriittisen alueen käyttöä ei ohjelmiston puutteiden vuoksi hallinnoida onnistuneesti. *Data Entry Complete* muuttujasta ei ole hyötyä rinnakkaisuuden kontrollointiin, koska sen arvoja voidaan muuttaa samaan aikaan, kun järjestelmä muutokset ovat käynnissä.

Toisaalta *Yakima Valleyn* tapauksessa käyttäjä siirtää työvalotilaan ja hoidon alkaessa tämä tila jää päälle. Ohjelmisto, joka käyttäjän käskyn kanssa rinnakkaisen virheentarkastus muuttuja *Class3* ylivuodon takia mahdollistaa säteilyttämisen väärässä asennossa, mikä antaa potilaalle moninkertaisen annoksen turvalliseen verrattuna.

Ajatellen pelkästään tässä käsiteltyjen virheiden korjausta olisi ensimmäinen mahdollista välttää vaikkapa estämällä käyttäjän syötteen käsittely kokonaan, kun laitetta asetetaan säätöjen mukaiseksi. Therac-20:ssa tämä oli toteutettu laitteistotason lukituksina, mutta sama olisi mahdollista toteuttaa myös ohjelmistossa.

Toinen tapaus olisi vastaavasti mahdollista estää esimerkiksi Therac-20 tapaisella erillisellä tarkkailujärjestelmällä, joka estäisi säteilyttämisen väärässä asennossa tai ensimmäisen korjauksen tavoin ohjelmiston lukituksella, joka estää käyttäjän syötteen prosessoinnin, kun virhetarkistus on käynnissä. (Virhearvon muuttaminen kiinteäksi, kuten AECL asian korjasi, saattaa estää ylivuodon, muttei korjaa itse kilpailutilanteen vaaraa.)

Paras vaihtoehto lienee kuitenkin, kuten aiemmin mainittu, luoda kokonaan uusi ohjelmisto ottaen huomioon kyseisen laitteiston ominaisuudet ja vaatimukset sekä rinnakkaisuuden ja kilpailutilanteiden vaarat ohjelmistotasolla heti alusta alkaen.

Lähde

Leveson, Nancy: *Medical devices: Therac-25,* University of Washington.
Saatavilla: http://sunnyday.mit.edu/therac-25.html