

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Sebastjan Kozoglav

**Simulator v sistemu za dostavo
odmerka protonskega žarka**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: viš. pred. dr. Robert Rozman

Ljubljana, 2018

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Razvoj naprav in njihove programske opreme za uporabo v medicini je zelo zahteven in mora zadovoljevati najstrožje standarde ter priporočila. Predvsem sta pomembni varnost in učinkovitost tovrstnih naprav. Izdelajte programski simulator, ki bo omogočal podroben preizkus funkcionalnosti sistema za dostavo odmerka protonskega žarka in njegove povezave z nadzoranim sistemom celotnega postopka zdravljenja. Pri tem zagotovite skladnost z obstoječimi standardi in ostalimi izhodišči za izdelavo tovrstne programske opreme. Simulator preizkusite tudi v praksi in opišite svoje izkušnje.

Rad bi se zahvalil mentorju, viš. pred. dr. Robertu Rozmanu, za odzivnost kljub nasičenemu urniku in prijazno pomoč pri izdelavi moje diplomske naloge. Zahvaljujem se tudi Gašperju Tomažiču iz podjetja Cosylab d.d., za strokovno svetovanje in spodbujanje pri izdelavi diplomskega dela. Posebna zahvala gre moji partnerki, ki mi je ves čas študija stala ob strani in verjela vame. Iskreno se zavaljujem tudi družini, za uso podporo in finančno pomoč med študijem. Zahvaljujem se tudi vsem ostalim, ki so mi kakorkoli pomagali pri delu.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Protionska terapija	3
2.1	Splošno o protionski terapiji	3
2.2	Opis delovanja protionske terapije	5
3	Kontrolni sistem za nadzor zdravljenja s protionsko terapijo	9
3.1	Vloga kontrolnega sistema v protionski terapiji	9
3.2	Sistem za nadzor zdravljenja Cosylab TCS	9
4	Sistem za dostavo odmerka protionskega žarka	13
4.1	Opis delovanja	13
4.2	Povezava sistema z ostalimi komponentami	16
5	Standardi, pravila in dogovori pri implementaciji	19
5.1	Medicinski standardi	19
5.2	Interna pravila in dogovori	21
6	Razvoj simulatorja v sistemu za dostavo odmerka protionskega žarka	23
6.1	Uporabljene tehnologije	23

6.2	Postopek implementacije	27
6.3	Razvoj	32
7	Sklepne ugotovitve	39
	Literatura	42

Seznam uporabljenih kratic

kratica	angleško	slovensko
TCS	Treatment Control System	Sistem za nadzor zdravljenja
DDS	Data Distribution Service	Storitev za distribucijo podatkov
SVN	Subversion	Sistem za nadzor različic datotek
CM-100	Treatment console	Nadzorna konzola za protonsko terapijo
IDE	Integrated development environment	Integrirano razvojno okolje
OIS	Oncology information system	Onkološki informacijski sistem
ACS	Accelerator control system	Sistem za nadzor pospeševanja
PACS	Picture archiving and communication system	Sistem za arhiviranje slik in komunikacije

Povzetek

Naslov: Simulator v sistemu za dostavo odmerka protonskega žarka

Avtor: Sebastjan Kozoglav

Trenutno obstaja več načinov za zdravljenje raka. Eden izmed teh je tudi radioterapija. Obsevanje s protoni je vrsta radioterapije, ki uničuje rakave celice s pomočjo protonskih žarkov. Protonska terapija predstavlja prednost pred bolj pogosto uporabljeno terapijo s fotonskim snopom, saj omogoča boljši nadzor pri dostavi žarkov. Njihova smer in hitrost se lahko upravlja z večjo natančnostjo, kar pa predstavlja zahtevo po bolj kompleksnem kontrolnem sistemu. V sodelovanju s podjetjem Cosylab je bil razvit simulator takega sistema. Simulator je potreben za testiranje in validacijo delovanja kontrolnega sistema za nadzor zdravljenja s protonsko terapijo. Implementacija simulatorja vključuje implementacijo delovanja naprave, njen simulator, komunikacijo med njima in komunikacijo s sistemom za nadzor zdravljenja protonske terapije.

Ključne besede: Protonska terapija, protonski žarek, simulator, nadzor sistema.

Abstract

Title: Simulator in the Delivery System of a Proton Beam Dose

Author: Sebastjan Kozoglav

There are currently several ways to treat cancer. One of these is radiotherapy. Proton therapy is a type of radiotherapy, that destroys cancer cells with the help of proton beams. Proton therapy is an advantage over a more commonly used photon beam therapy, as it provides better control in the delivery of rays. Their direction and speed can be managed with a higher precision, which implicates a more sophisticated control system. In collaboration with Cosylab, a simulator of such a system was developed. The simulator is necessary for testing and validating the operation of the treatment control system for proton therapy. The simulator implementation includes the implementation of the device's operation, its simulator, communication between them, and communication with the proton therapy treatment control system.

Keywords: Proton therapy, proton beam, simulator, system control.

Poglavje 1

Uvod

Rak je ena izmed hujših bolezni današnjega časa, ki vsakemu od nas požene “strah v kosti”. Kljub temu, da bolezen že dolgo poznamo, je še vedno zelo težko ozdravljiva, hkrati pa vse bolj pogosta. Pri raku celice zaradi nenadzorovane delitve tvorijo tumor, ki kasneje dobi tudi sposobnost širjenja v druge dele telesa. Brez zdravljenja je napredovanje bolezni za bolnike pogosto usodno.

Ko sem začel z delom v podjetju Cosylab, sem se kot del ekipe spoznal s področjem računalništva v medicini. Delo ekipe je bilo osredotočeno na razvoj sistema za nadzor zdravljenja protonske terapije. Med delom sem izvedel več o načinih zdravljenja raka. Spoznal sem, da se pri zdravljenju raka s protonskimi žarki prizadane manj zdravega tkiva, kot pri terapiji s fotonskim snopom. To mi je dalo dodatno motivacijo pri razvoju simulatorja in pri pisanju diplomske naloge.

Rak se lahko zdravi na različne načine: s kemoterapijo, operativnim posegom in radioterapijo. Pri zdravljenju z radioterapijo se pri nas najpogosteje uporablja obsevanje s fotonskim snopom.

Ena izmed vrst radioterapije je tudi obsevanje s protoni, ki pa zaradi visokih stroškov razvoja pri nas ni tako poznana in se izvaja predvsem v tujini. Prednost te terapije je boljši nadzor nad obsevanjem, za razliko obsevanja s fotonskim snopom, ki lahko z zadostno dozo žarka, poleg tumorja prizadane

tudi zdravo tkivo. Protonski žarek sprošča najvišjo energijo na mestu kjer se ustavi. Zdravljenje s protonsko terapijo s pomočjo sistema za dostavo odmerka protonskega žarka izkorišča to lastnost. Njegova naloga je nadzorovanje smeri in globine, ki jo žarek doseže. Na ta način, se pri zdravljenju s protonsko terapijo prizanesse zdravemu tkivu in se z večjim delom doze obseva predvsem tumor.

V diplomski nalogi sem se osredotočil na protonsko terapijo, natančneje na razvoj sistema za nadzor zdravljenja te terapije, ki je del sistema za nadzor zdravljenja protonske terapije, ki ga razvija podjetje Cosylab.

Pri razvoju sistema za nadzor zdravljenja protonske terapije je potrebno poznati osnove delovanja protonske terapije, vlogo sistema pri terapiji, pravila in dogovore, ki se jih moramo držati pri implementaciji, da projekt uspe.

V nadaljevanju diplomskega dela najprej spoznamo osnove protonske terapije in njenega delovanja. Sledi splošen opis kontrolnega sistema za nadzor zdravljenja s protonsko terapijo in opis konkretnega kontrolnega sistema, ki ga razvijamo v podjetju Cosylab. Spoznamo tudi, kakšno vlogo v zdravljenju s protonsko terapijo ima sistem za dostavo odmerka protonskega žarka in kakšne metode uporablja pri dostavi žarkov do pacienta. Dotaknemo se medicinskih standardov, saj je njihova uporaba pri razvoju medicinskih naprav ključnega pomena za zagotavljanje varnega in učinkovitega delovanja. Sledi opis razvoja simulatorja sistema za dostavo odmerka protonskega žarka. Diplomsko nalogo zaključim z mislijo o izkušnjah, ki sem jih pridobil pri razvoju simulatorja in predlogom za izboljšavo, s katerimi bi bil nadzor simulatorja uporabniku bolj prijazen.

Poglavje 2

Protonska terapija

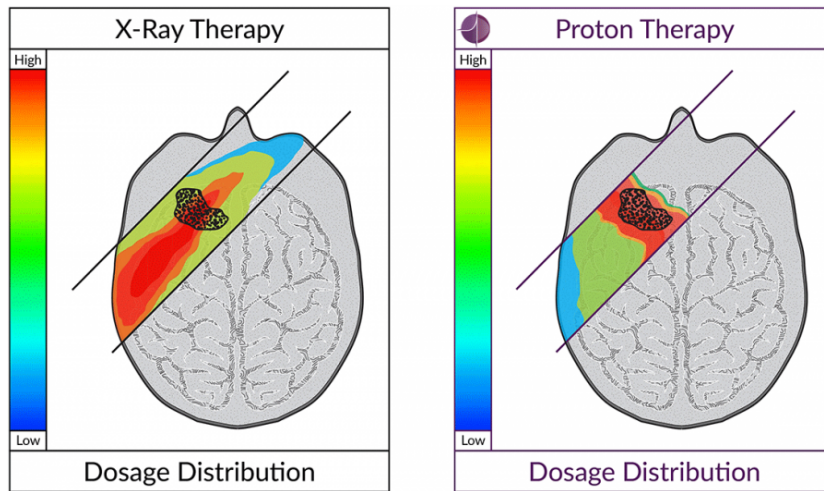
Za lažje razumevanje diplomske naloge, bomo v tem poglavju spoznali kaj je protonska terapija in kako deluje.

2.1 Splošno o protonski terapiji

Trenutno obstajajo tri glavne vrste zdravljenja raka: kemoterapija, operativni poseg in radioterapija. Obsevanje s protoni je ena izmed vrst radioterapij. Uporablja se lahko kot samostojna terapija, ali pa se združi z drugimi tipi zdravljenja.

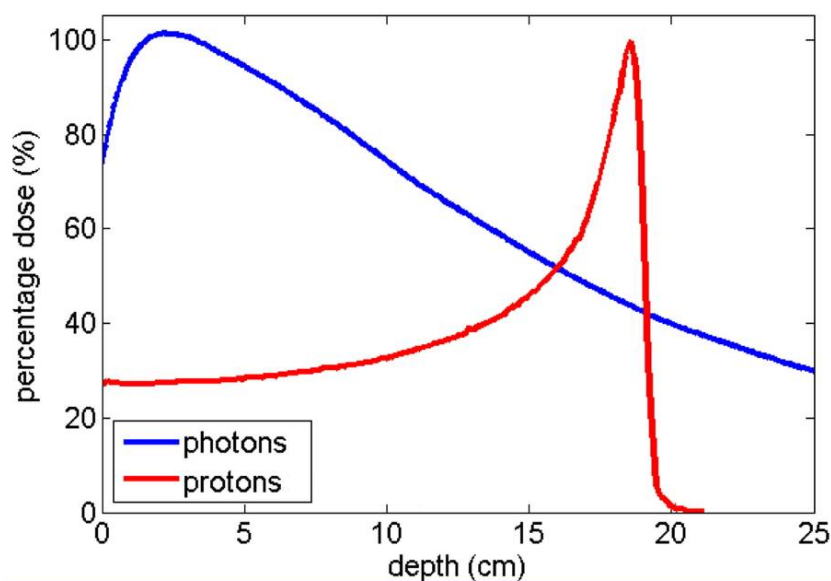
Najbolj pogosti tip radioterapije, ki se uporablja za zdravljenje raka, je obsevanje s fotonskim snopom. Čeprav terapija s fotoni z zadostno dozo žarka uspešno zdravi rakov tumor, je tako kot tumor, prizadeto tudi zdravo tkivo saj dozo žarka prejmejo zdravo tkivo na površju, rakov tumor in zdravo tkivo za tumorjem. Zaradi tega se pri zdravljenju s fotonskim snopom dozo žarka pogosto zmanjša, da se izogne nezaželenim stranskim učinkom in poškodbam zdravega tkiva. Ker imamo pri protonski terapiji boljši nadzor nad obsevanjem zdravega tkiva, je ta bolj primerna za zdravljenje raka, ki se nahaja blizu pomemnih delov telesa, npr. rak blizu očesa ali hrbtenjače. Pogosto se uporablja tudi na otrocih obolelih za rakom, saj je pri njih zelo pomembno, da je nezaželenih učinkov čim manj. Kljub prednosti pred terapijo s foton-

skim snopom, je zaradi potrebe po visoko specializiranih in dragih napravah, protonska terapija manj razširjena po svetu. Zaradi visoke cene zdravljenja, pa se pogosto pojavlja vprašanje: “Ali so manjši stranski učinki vredni višje cene terapije?” Odgovor na to vprašanje je največkrat odvisen od tipa in mesta rakovega tumorja. Na sliki 2.1 je prikazana primerjava obsevanja s fotoni in protoni.



Slika 2.1: Primerjava obsevanja s fotoni in protoni [8].

Fotonski žarki sprostito največ energije pri vstopu skozi tkivo, žarek protonske terapije pa ima obliko braggove krivulje in je njegova energija najvišja na mestu, kjer se proton ustavi. Na poti žarka je sevanja sorazmerno malo, po tem ko se ustavi, pa skoraj nič več [16, 10, 19]. Slika 2.2 prikazuje razmerje med energijo žarka in njegovo globino v telesu pacienta ob obsevanju s fotoni in protoni.



Slika 2.2: Primerjava energije žarkov fotonov in protonov [15].

2.2 Opis delovanja protonske terapije

Protoni so pozitivno nabiti delci. V protonski terapiji izvirajo iz vodikovega atoma. Vodik je sestavljen iz enega protona in enega elektrona, ki se ju loči s pomočjo električnega polja. Protone se nato usmeri v krožni pospeševalnik, kateri s pomočjo magnetov pospeši protonske delce. Ko protoni dosežejo dovolj veliko hitrost, se jih po cevovodu usmeri v pojemalnik ¹. Tu se hitrost protonov zmanjša, saj je globina, ki jo protonski delci dosežejo v pacientu odvisna od njihove hitrosti. Protone se nato skozi transportni sistem, sestavljen iz magnetnih polj, usmeri do sobe za zdravljenje. Preden se dostavi žarek protonov, se naredi rentgensko slikanje pacienta. S tem se preveri ali je tumor na ustreznem mestu in se glede na rezultat slikanja, po potrebi premakne mizo na kateri je pacient ali pa strukturo, v kateri je nameščena oprema, ki dostavlja žarek pacientu. Protonski žarek se nato usmeri v pacienta s pomočjo elektromagnetnih polj [19].

¹Naprava, ki zmanjšuje energijo protonov.

Na sliki 2.3 je prikazan primer sobe za zdravljenje. Iz slike je razvidno, kako izgleda površina, na kateri leži pacient, nad posteljno površino pa je naprava, ki dostavlja žarek do pacienta.



Slika 2.3: Primer sobe za zdravljenje [4].

2.2.1 Potek zdravljenja

Ko se pacienta seznanijo z boleznijo in zdravljenjem, ki sledi, se ga pred obsevanjem naroči na simulacijo obsevanja. Med simulacijo se pacientu prilagodi načrt zdravljenja. Naredi se rentgenska slika pacienta in se na njem zariše orientacijske točke, na podlagi katerih se pacienta nastavi v pravilen položaj pri obsevanju.

Na obsevalni terapiji, se pacienta pozicionira na enak način, kot je bil pozicioniran na simulaciji obsevanja. Ko je pacient na ustreznem mestu, se naredi rentgensko slikanje pacienta, kjer se preveri natančno lokacijo tumorja. V primeru, da se je premaknil pacient ali sam tumor, se pacienta premakne na ustrezen položaj. Obsevanje se nato prične.

Protonskih žarkov se ne vidi in ne čuti. Pacient mora biti med obsevanjem popolnoma pri miru, v sobi za zdravljenje pa ne sme biti nikogar drugega.

Osebjem, ki upravlja z napravo za nadzor zdravljenja protonske terapije, spremlja in nadzoruje potek zdravljenja iz druge sobe.

Sistem za nadzor zdravljenja se lahko upravlja na daljavo s pomočjo posebne konzole, na sliki 2.4, s katero se lahko zdravljenje prične, začasno ustavi, zaključi ali prekine, če je to potrebno. Konzola ima tudi zaslon, na katerem se prikazujejo različne informacije med zdravljenjem, ki so za osebo, ki upravlja z napravo, pomembne [6, 7].



Slika 2.4: Nadzorna konzola za protonsko terapijo [1].

Poglavje 3

Kontrolni sistem za nadzor zdravljenja s protonsko terapijo

Naloga kontrolnega sistema za nadzor zdravljenja s protonsko terapijo (Treatment Control System ali krajše TCS) je popolna koordinacija celotnega procesa zdravljenja in omogočanje medicinskemu osebju zdravljenje pacientov.

3.1 Vloga kontrolnega sistema v protonski terapiji

Sistem nudi grafični vmesnik, kateri prikazuje ustrezne informacije protonske terapije. Uporabnikom tudi omogoča nadzor nad samim potekom terapije in njeno strojno opremo, npr. sistem za slikanje pacienta.

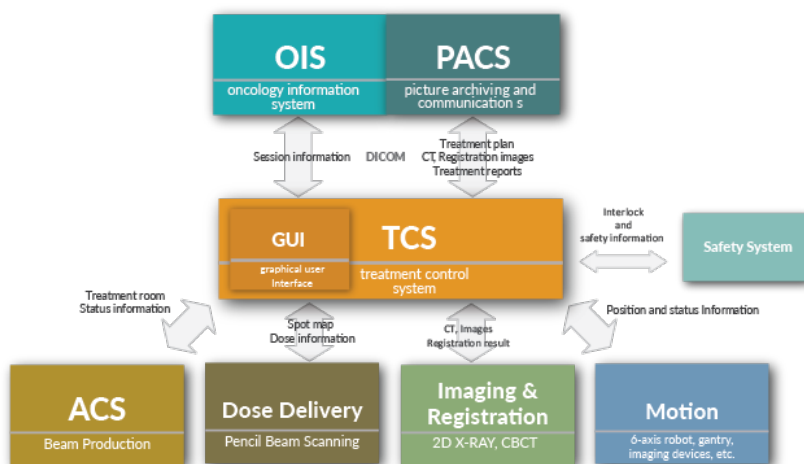
TCS je glavni koordinator med različnimi sistemi, ki so vključeni pri zdravljenju s protonsko terapijo [17].

3.2 Sistem za nadzor zdravljenja Cosylab TCS

Podjetje Cosylab je razvilo svoj TCS sistem, ki se imenuje Cosylab Treatment Control System (C-TCS). Sistem nudi vse funkcionalnosti, ki jih zahtevajo

moderni TCS sistemi in povezuje vse sisteme, ki so del zdravljenja. Zasnovan je tako, da podpira komercialne in prilagojene rešitve. Omogoča tudi nadgradnje sistema za delo z novimi prihajajočimi tehnologijami [9].

Na sliki 3.1 je prikazan pregled nad sistemom TCS in komponentami, ki jih koordinira.



Slika 3.1: Struktura sistema TCS [9].

Onkološki informacijski sistem (OIS) in sistem za arhiviranje slik in komunikacije (PACS) sta medicinska sistema, ki hranita podatke o pacientih. C-TCS zagotavlja vmesnik za branje načrtov zdravljenja, rentgenskih slik in podatkov predhodnjega zdravljenja, katere pridobi iz teh dveh sistemov. C-TCS obvešča OIS o napredku poteka dela in shranjuje podatke v PACS (npr. pridobljene slike).

Varnostni sistem (Safety System) skrbi, da se pacienti in naprave, ki so del zdravljenja, ne poškodujejo. Sistem neprestano obvešča uporabnika sistema C-TCS o stanjih naprav, ki se uporabljajo med zdravljenjem.

Pacient med zdravljenjem leži na posebni robotski mizi, nad pacientom pa se nahaja naprava, ki dostavlja protonske žarke v pacienta. Obe komponenti se lahko premikata, če je to potrebno. Premikanje opravlja sistem za

pozicioniranje pacientov (Motion). C-TCS podpira vse funkcionalnosti, ki so potrebne za nadzor in pozicioniranje robotske mize in naprave, ki dostavlja protonske žarke do pacienta. Če pride do napake pri nastavitvi, lahko C-TCS samodejno premakne mizo, na kateri leži pacient, ali napravo, ki dostavlja protonske žarke do pacienta, v pravilen položaj.

Postopek slikanja in registracije (Imaging and Registration) se opravi pred vsakim začetkom zdravljenja. Postopek preveri, ali je pacient v pravilnem položaju in pripravljen na zdravljenje. Naredi se rentgensko sliko pacienta in se jo primerja z referenčno. Glede na odstopanje se popravi pacienta v pravilen položaj, ki je ustrezen za njegovo zdravljenje. C-TCS se lahko poveže na različne sisteme za slikanje in registracijo in jim zagotavlja vse potrebne podatke (npr. referenčno rentgensko sliko). C-TCS samodejno pridobi podatke, ki so rezultat postopka registracije in jih upošteva pri pozicioniranju pacienta (Motion). Prav tako C-TCS zagotavlja shranjevanje vseh rentgenskih slik, ki so bile vpletene v postopek slikanja in registracije.

Sistem za dostavo odmerka protonskega žarka (Dose Delivery) je odgovoren, da se žarek dostavi v pacienta na točno določeno mesto. C-TCS zagotavlja sistemu vse informacije, ki so potrebne za njegovo delovanje (npr. recept o pacientu), iz sistema pa bere informacije o skeniranju in dostavljeni dozi.

Med obsevanjem, C-TCS naroča sistemu za nadzor pospeševanja (ACS), naj zagotovi potrebne protonske žarke v pravo sobo za zdravljenje.

Poglavje 4

Sistem za dostavo odmerka protonskega žarka

Sistem za dostavo odmerka protonskega žarka je odgovoren za dostavo in nadzor odmerka žarka, ki je namenjen pacientu. V tem poglavju bom opisal njegovo delovanje in povezavo z ostalimi komponentami.

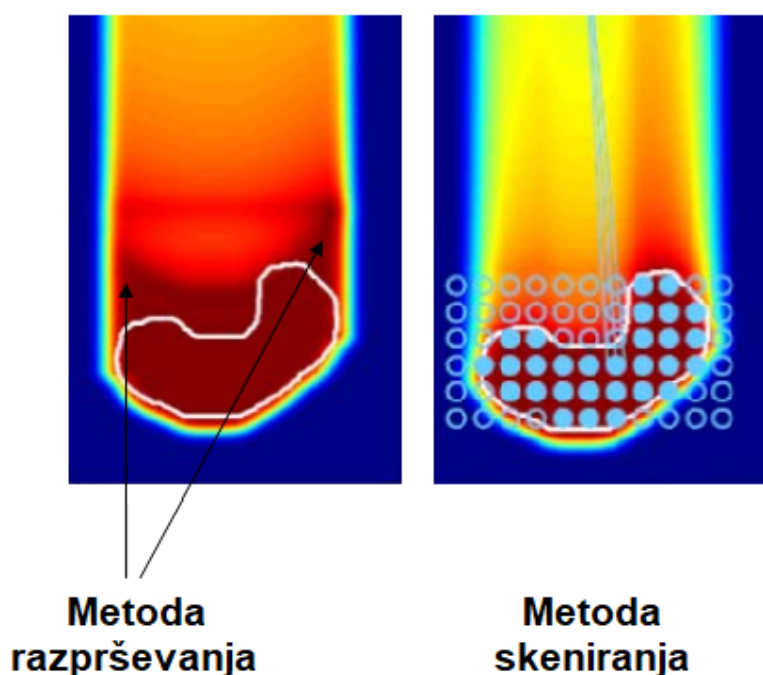
4.1 Opis delovanja

Da lahko sistem zagotavlja dostavljanje potrebnih odmerkov na pravo mesto, med samim zdravljenjem, upravlja z dvema parametroma. Pomembna sta smer protonskih delcev, da se žarek usmeri na pravo mesto in hitrost žarka, ki določa njegovo globino. Večja kot je hitrost, večja je globina, ki jo žarek doseže. Pomembno je, da se protonski delci ustavijo na pravem mestu, saj je njihova energija takrat najvišja [19, 5].

Da sistem za dostavo odmerka protonskega žarka lahko doseže to funkcionalnost, uporablja dve metodi:

- metoda razprševanja,
- metoda skeniranja dostavnega žarka.

Delovanje teh dveh metod se med seboj razlikuje po načinu dostave protonskih žarkov. Pri metodi razprševanja, sistem žarke oblikuje kot je oblikovan tumor na neki globini. Pri metodi skeniranja, pa se tumor razdeli po plasteh in obseva po točkah posamezne plasti, pri čemer pride do manj obsevanja zdravega tkiva okoli tumorja. Slika 4.1 prikazuje primerjavo obsevanja tumorja obeh metod.



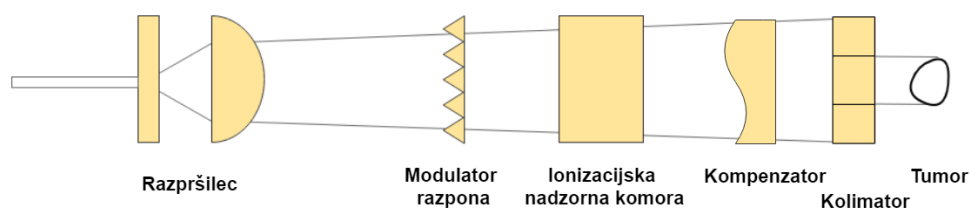
Slika 4.1: Primerjava obsevanja med metodama [11].

4.1.1 Metoda razprševanja dostavnega žarka

Pri metodi razprševanja dostavnega žarka se tumor obseva po horizontalnih plasteh enake globine. Posebne naprave oblikujejo protonski žarek tako, da se prilagaja obliki tumorja na posamezni plasti.

Potovanje žarka v sistemu za dostavo odmerka z metodo razprševanja, je prikazano na sliki 4.2. Žarek najprej prispe v razpršilec, kateri ga razprši

v horizontalnih in vertikalnih smereh. Nato gre skozi modulator razpona, ki s pomočjo grebenskega filtra in modulatorskega kolesa določi globino razpršenega žarka. Ionizacijska nadzorna komora nato pregleduje odmerek in položaj razpršenega žarka, ki se odzove, če sta odmerek ali položaj žarka izven sprejemljivih meja. Preden žarek zapusti sistem za dostavo odmerka protonskega žarka, gre še skozi dve komponenti imenovani kompenzator in kolimator, kateri razpršeni žarek oblikujeta tako, kot je oblikovan tumor na trenutni plasti [19, 5].

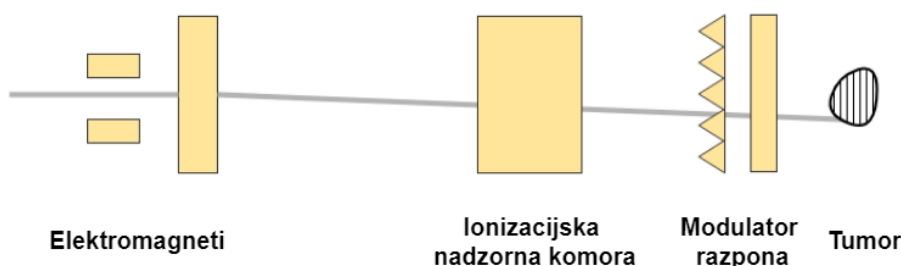


Slika 4.2: Metoda razprševanja dostavnega žarka.

4.1.2 Metoda skeniranja dostavnega žarka

Metoda skeniranja dostavnega žarka je bolj napredna od metode razprševanja, saj v celoti izkorišča prednosti obsevanja s protoni. Tumor se prav tako razdeli v plasti enake globine, kjer pa je vsaka plast obsevana po točkah, na katere se žarke usmerja s pomočjo elektromagnetnih polj.

Potovanje žarka v sistemu za dostavo odmerka z metodo skeniranja, je prikazano na sliki 4.3. Začne se pri posebnih elektromagnetih, ki usmerjajo žarek v horizontalni in vertikalni smeri. Žarek gre nato skozi ionizacijsko nadzorno komoro. Ta zagotavlja sistemu podatke o odmerku in položaju žarka v realnem času, kar omogoča sistemu možnost da žarek popravi, če ta ni ustrezen. Nato pa gre žarek samo še skozi modulator razpona, kjer se prilagaja globino, ki jo protonski žarek doseže v pacientu [19, 5].



Slika 4.3: Metoda skeniranja dostavnega žarka.

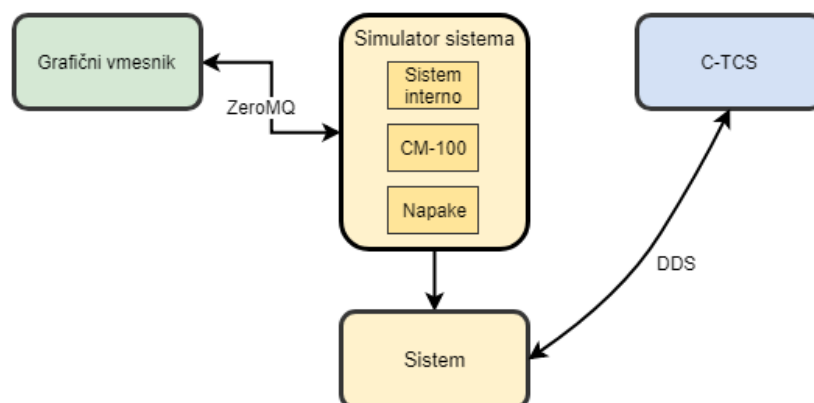
4.2 Povezava sistema z ostalimi komponentami

Celotna implementacija simulatorja sistema za nadzor dostave odmerka protonskega žarka je sestavljena iz dveh delov:

- sistem za nadzor dostave odmerka protonskega žarka,
- simulator sistema.

Slika 4.4 prikazuje povezave med sistemom in njegovim simulatorjem ter ostalih ključnih komponent. Sistem za dostavo odmerka protonskega žarka načeloma komunicira le s sistemom TCS. Komunikacija poteka na način pošiljatelj-naročnik, kjer za prenos podatkov skrbi protokol DDS. Več o protokolu v poglavju 6.1.1.

Simulator lahko poljubno nadzoruje sam sistem in ga postavlja v razna stanja z različnimi delovnimi procesi. V grobem lahko simulator nadzoruje interno delovanje sistema, delovanje nadzorne konzole CM-100 in postavlja sistem v razna stanja napak. Simulator komunicira s sistemom na enak način kot TCS sistem. Povezana sta prav tako s protokolom DDS, vendar je tukaj povezava enosmerna. Simulator samo govori, kako naj se sistem obnaša. S simulatorjem se lahko upravlja s knjižnico ZeroMQ; več v poglavju 6.1.2. Za namen nadzorovanja simulatorjev, imamo v podjetju Cosylab že implementiran grafični vmesnik, ki omogoča preprosto in učinkovito komunikacijo



Slika 4.4: Shema celotnega sistema za dostavo odmerka protonskega žarka.

med željeno programsko opremo. V tem primeru je ta programska oprema simulator sistema za nadzor dostave odmerka protonskega žarka.

Poglavje 5

Standardi, pravila in dogovori pri implementaciji

Uporaba standardov pri razvoju medicinskih naprav je ključnega pomena za zagotavljanje varnega in učinkovitega delovanja naprav, ki se uporabljajo v medicinske namene. Uporaba internih pravil, pa je priporočljiva za izogibanje nepotrebnim zmedam, če na projektu sodeluje več razvijalcev. V tem poglavju na kratko opisujem medicinske standarde in interna pravila, ki se jih držimo v podjetju Cosylab pri razvoju sistema C-TCS.

5.1 Medicinski standardi

Mednarodne medicinske standarde običajno izdajajo dve organizaciji: mednarodna organizacija za standardizacijo (ISO) in mednarodna komisija za elektrotehniko (IEC). Vsi standardi, ki jih organizaciji izdaja, veljajo za mednarodne, kar pomeni da veljajo po vsem svetu. V tem poglavju bom opisal medicinske standarde, ki se jih držimo pri implementaciji sistema C-TCS.

5.1.1 Standard ISO 14971:2012

To je standard, ki razvijalcem zagotavlja ogrodje, v katerem se opredeljuje in ocenjuje tveganja ter načrtuje, uvaja, nadzira in izboljšuje ukrepe

za zmanjševanje tveganja povezana z uporabo medicinskih naprav. Temu procesu se reče obvladovanje tveganja [13].

Glavni dejavniki, ki vplivajo na to, kako se ocenjuje tveganje in posledično kako so zasnovane medicinske naprave so:

- upoštevati se mora vse dejavnike tveganj,
- analiza tveganja mora biti izvedena za vse primere in mora vključevati vsa možna tveganja,
- vsa tveganja se morajo zmanjšati kolikor je mogoče,
- vsi obvladovalni ukrepi se morajo izvesti,
- tveganje se naj zmanjša z varno zasnovano arhitekturo.

5.1.2 Standard ISO 13485:2016

To je mednarodno dogovorjen standard, ki določa zahteve za sisteme vodenja kakovosti, ki je specificiran za industrijo medicinskih pripomočkov. Za uporabo je namenjen proizvajalcem in dobaviteljem medicinskih naprav.

ISO 13485:2016 se osredotoča na to, kako naj podjetja upravljajo z odločitvami, ki so rezultat procesa za obvladovanje tveganja [3].

5.1.3 Standard IEC 62304:2006

To je standard, ki določa življenski cikel zahtev pri razvoju in vzdrževanju programske opreme, ki se uporablja v medicinskih napravah. Programska oprema je lahko že sama po sebi medicinska naprava, ali pa vgrajeni ali sestavni del končne medicinske naprave [18].

5.2 Interna pravila in dogovori

Ker na samem projektu sodeluje več razvijalcev, se za preprečevanje zmede in konfliktov pri razvoju vsi držimo vnaprej določenih internih pravil.

Razvoj se hrani v git repozitoriju namenjenemu C-TCS projektu. Ta repozitorij se deli na podmodule, kjer je vsak podmodul posamezen del C-TCS sistema ali sistema protonske terapije, ki ga C-TCS koordinira.

Vso dokumentacijo o projektu, hranimo v svn repozitoriju, vse zahteve in arhitekturo pa hranimo na CodeBeamer platformi, katera je namenjena za upravljanje življenjskega cikla aplikacij.

Pri implementaciji programskih rešitev se držimo naslednjih načel:

- piši pregledno in preprosto programsko kodo,
- naredi načrt pred implementacijo,
- poskrbi da je programska koda enostavna za vzdrževanje in nadgrajevanje,
- piši JavaDoc komentarje in Unit teste.

Poglavje 6

Razvoj simulatorja v sistemu za dostavo odmerka protonskega žarka

V tem poglavju bom opisal tehnologije, ki sem jih uporabil pri razvoju, opisal bom postopke implementacije in sam razvoj simulatorja sistema za dostavo odmerka protonskega žarka.

6.1 Uporabljene tehnologije

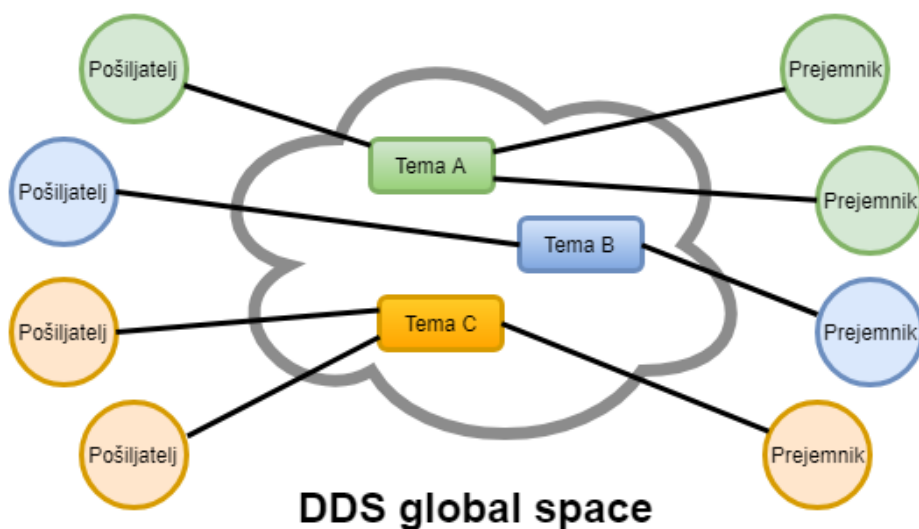
Razvoj celega C-TCS sistema je potekal v programskem jeziku Java 1.8 v razvojnem okolju Eclipse IDE. Po dogovoru se lahko uporablja tudi drugo razvojno okolje, vendar se mora poskrbeti za ustrezno formatiranje kode. Projekt se upravlja s sistemom za vodenje in gradnjo projektov Maven, vzdržuje pa se ga na git repozitoriju.

6.1.1 Protokol DDS

Simulator sistema za nadzor dostave odmerka protonskega žarka uporablja protokol DDS (Data distribution service) za medsebojno komunikacijo s C-TCS sistemom. DDS nudi komunikacijo, ki temelji na temah (topics). Na

določeno temo so lahko povezani pošiljatelj in prejemniki. Ko pošiljatelj pošlje podatek na njegovo temo, DDS poskrbi, da se podatek dostavi vsem prejemnikom, ki so vezani na njo [2].

Primer uporabe protokola je viden na sliki 6.1. Zeleni pošiljatelj je vezan na temo A. Če pošlje podatek, ga prejmeta oba zelena prejemnika.



Slika 6.1: Primer uporabe protokola DDS.

6.1.2 Knjižnica ZeroMQ

Sistem C-TCS za njegovo delovanje uporablja več različnih storitev. Vsaka storitev je odgovorna za del funkcionalnosti celotnega sistema. Da lahko storitve sodelujejo, morajo biti sposobne komunicirati med seboj. Za to komunikacijo uporabljajo knjižnico ZeroMQ.

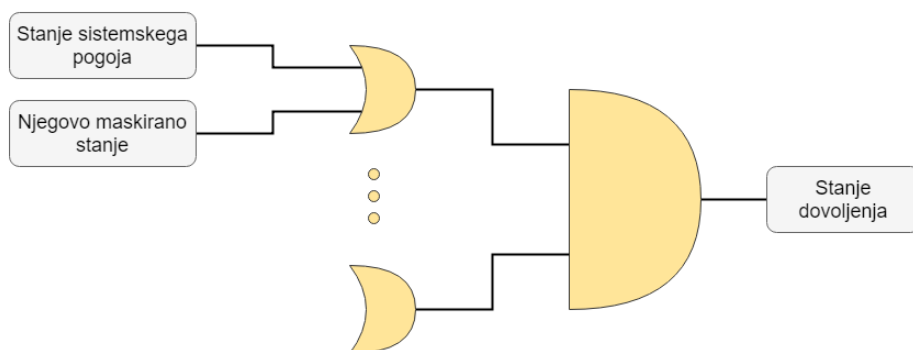
ZeroMQ je knjižnica za sporočanje, namenjena za uporabo porazdeljenih ali sočasnih programskih rešitev. ZeroMQ zagotavlja zanesljivo hkratno oddajanje več prejemnikom na različnih vzorcih vtičnic (request/response, push/pull, publish/subscribe in exclusive pair) [20].

6.1.3 Dovoljenja in sistemski pogoji

Za lažji nadzor samega sistema, njegovega stanja in delovanja uporabljamo tako imenovana dovoljenja (permits) in sistemske pogoje (interlocks). S pomočjo sistemskih pogojev se nadzoruje stanje dovoljenj, s pomočjo dovoljenj pa nadzorujemo prehode med stanji končnega avtomata sistema za dostavo odmerka protonskega žarka. Avtomat je prikazan na sliki 6.3.

Dovoljenja so vrednosti z dvema stanjema. Označujejo dovoljenje, na katere se sistem odziva. Stanja so lahko resnična ali neresnična. Izračuna se jih na podlagi kombinacij določenih sistemskih pogojev. Stanja sistemskih pogojev so prav tako lahko resnična ali neresnična, določijo pa jih stanja sistemov, ki so del zdravljenja.

Da se dovoljenje nastavi na resnično stanje, morajo biti vsi sistemski pogoji, ki dovoljenje sestavljajo, nastavljeni. Vrednosti sistemskih pogojev pa nastavlja več vrednosti. Nastavi se lahko dejanska vrednost sistema pogoja (Avtomobil je prižgan, lahko spelješ), ali pa njegova maskirana vrednost, kjer se simulira dejansko stanje pogoja (obnašaj se, kot da je avtomobil prižgan, lahko spelješ). Če je nastavljena dejanska vrednost sistema pogoja na resnično stanje, maskirana vrednost nanj ne vpliva. Če pa je nastavljena dejanska vrednost na neresnično, lahko z maskiranjem vplivaš na njegovo vrednost, ki jo predstavlja. Delovanje dovoljenj in sistemskih pogojev je prikazano z logičnimi vrati na sliki 6.2.



Slika 6.2: Prikaz delovanja dovoljenj in sistemskih pogojev.

Za lažjo predstavo delovanja zgoraj opisanega principa dovoljenj in pogojev, ga bom uporabil na poenostavljenem primeru iz vsakdanjega življenja - Ogrevanje stanovanja s centralno kurjavo.

- Sistemski pogoji
 - Ali je sobni termostat prižgan?
 - Ali je temperatura sobe manjša od nastavljene?
 - Ali je kurilna peč prižgana?
 - Ali je temperatura peči dovolj visoka?
- Dovoljenja
 - Vklopi obtočno črpalko za ogrevanje in s tem potisni vročo vodo v radiatorje ter posledično ogrej stanovanje

6.1.4 Nespremenljivi objekti

Nespremenljiv (angl. immutable) objekt je objekt, kateremu se vrednosti ne morejo spreminjati. Informacija, ki se hrani v nespremenljivem objektu, je dana ob njegovi inicializaciji. Ko je objekt inicializiran, se njegova vrednost ne more več spreminjati. Če želimo spremeniti vrednost nespremenljivega objekta, se ponavadi naredi njegova kopija s spremenjenimi vrednostmi, lahko pa se ustvari tudi novega [12].

Pri C-TCS razvoju, je uporaba nespremenljivih objektov zelo zaželjena. Ker se vrednosti nespremenljivih objektov po inicializaciji ne more spremeniti, bo prebrana vrednost vedno pravilna, tudi pri sočasnih dostopih. Ker pa se vsakič, ko želimo spremeniti vrednost nespremenljivega objekta, naredi nov objekt, to predstavlja slabost pri primerih, za katere pričakujemo pogosto spreminjanje, saj se zaradi več nerabljenih objektov lahko bolj pogosto sproži proces sproščanja pomnilnika ¹.

6.2 Postopek implementacije

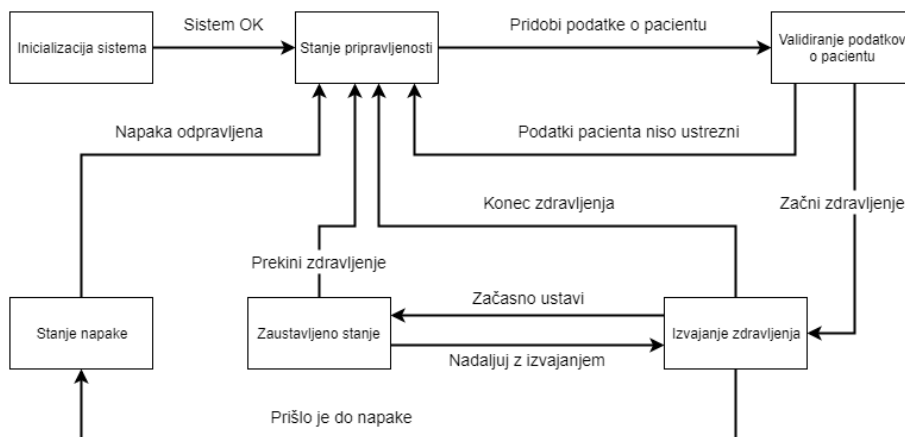
V tem poglavju opisujem postopek implementacije, ki sem se ga držal med razvojem. Prvi korak je bila implementacija delovanja sistema za dostavo odmerka protonskega žarka, nato je bilo potrebno za sistem razviti simulator. Po implementaciji obeh, je sledilo sprva ročno testiranje, nato pa pisanje JUnit testov, s katerimi se preveri funkcionalnost kode. Po uspešno zaključenem testiranju, je bil zadnji korak pisanje dokumenta, v katerem je bilo potrebno podrobno opisati strukturo in delovanje simulatorja.

6.2.1 Sistem za dostavo odmerka protonskega žarka

Delovanje sistema za dostavo odmerka protonskega žarka je natančno definirano s končnim avtomatom. Dejanska stanja sem poenostavil in združil v smiselne skupine, prikazano na sliki 6.3. Za lažje nadzorovanje sistema upo-

¹Obnavljanje prostora v pomnilniku z brisanjem neuporabljenih objektov.

rabljamu dovoljenja in sistemske pogoje, katerih delovanje je predstavljeno v poglavju 6.1.3.



Slika 6.3: Osnovno delovanje sistema.

Ob zagonu sistem prebere konfiguracijo naprave ter nastavi vse potrebne parametre, ki so potrebni za njegovo delovanje. Sistem nato preide v stanje pripravljenosti, kjer čaka na podatke o pacientu, katere pridobi s strani TCS sistema. Podatki o pacientu so v obliki recepta. Recept sestavljajo edinstvena identifikacijska številka pacienta, končni odmerek protonskega žarka in seznam vseh plasti tumorja. Vsaka plast vsebuje podatke o skupnem odmerku ter o točkah, ki se nahajajo na tej plasti. Vsaka točka pa vsebuje koordinate nahajališča ter odmerek, ki ga mora sistem za dostavo odmerka protonskega žarka dostaviti.

Ko sistem za dostavo odmerka protonskega žarka prejme podatke o pacientu, jih preveri skozi naslednji validacijski postopek. Sistem sprejme recept o pacientu, ki ga pošlje TCS. Nato preveri ali so podatki v receptu ustrezni in pošlje recept nazaj na TCS. Ta nato poslani in prejeti recept primerja med seboj. Če sta recepta enaka, sistem prične z zdravljenjem. V primeru da nista enaka, pa se recept zavrne in do zdravljenja ne pride. Sistem se nato vrne v stanje pripravljenosti, kjer ponovno čaka na recept.

Pred pričetkom izvajanja zdravljenja, mora sistem iz danega recepta pre-

brati in nastaviti začetno plast in točko tumorja, kjer bo pričel z obsevanjem. Nato se lahko zdravljenje začne. Sistem dostavi odmerek v trenutno točko, po tem pa se pomakne na naslednjo. Ta postopek se ponavlja, dokler niso obsevane vse točke v trenutni plasti. Ko je cela plast obsevana, sistem prične z obsevanjem naslednje plasti. Ta postopek se ponavlja, dokler ni obsevan cel tumor. Več o samem zdravljenju si lahko preberete v poglavju 2.2.

Po končanem izvajanju zdravljenja sistem preveri ali se je zdravljenje končalo uspešno, nato pa se vrne v stanje pripravljenosti. Sistem je sedaj pripravljen na zdravljenje novega pacienta.

Med izvajanjem zdravljenja lahko uporabnik izvajanje sistema začasno ustavi. Zdravljenje se lahko nato nadaljuje, ali pa se zaustavi dokončno in preide v stanje pripravljenosti. Sistem tako med zdravljenjem, kot tudi med samim delovanjem sistema, aktivno preverja ali je prišlo do kakršne koli napake in se na njo ustrezno odzove.

6.2.2 Simulator sistema

Preden se sistem za nadzor zdravljenja s protonsko terapijo uporabi v praksi, se mora za sisteme, ki jih koordinira, napisati simulator, s katerim se preveri njegovo pravilno delovanje. Simulator se nadzira s pomočjo knjižnice ZeroMQ. Med razvojem in testiranjem, smo za ta namen uporabljali že narejen grafični vmesnik, ki je omogočal pošiljanje zahtev s pomočjo ZeroMQ knjižnice.

Funkcionalnosti simulatorja se lahko delijo v šest večjih skupin, ki jih bom opisal spodaj.

Nadzor stikala na ključ CM-100 konzole

Na konzoli je stikalo na ključ, s katerim se nadzoruje delovni proces zdravljenja. Stikalo omogoča tri različne pozicije; prikazan je na sliki 6.4.

Pozicija OFF se uporablja, ko soba za zdravljenje ni aktivna, ali ko se pacient še pripravlja na začetek zdravljenja. Pozicija PREP označuje da je sistem pripravljen za začetek zdravljenja. V tem koraku se pridobi podatke

o zdravljenju pacienta. Ko je ključ v poziciji TREAT, pa se sistemu dovoli, da prične z zdravljenjem.

Pozicije ključev so sistemu za nadzor dostave odmerka protonskega žarka predstavljene kot sistemski pogoji. Ko simulator dobi ukaz, naj premakne ključ v določen položaj stikala, sistemu ustrezno nastavi določene sistemske pogoje.



Slika 6.4: Stikalo na ključ CM-100 konzole [1].

Nadzor žarka

Simulator omogoča nadzor nad žarkom med potekom zdravljenja. Sistemu lahko pošlje ukaz, ki prične z dostavo žarkov, jih začasno ustavi, nadaljuje, dostavo žarkov prekine ali pa uspešno zaključiti.

Simuliranje napak v receptu pacienta

Pred pričetkom zdravljenja, mora sistem za dostavo odmerka protonskega žarka pridobiti recept o pacientu. Recept mu pošlje TCS. Če je sistem za dostavo v stanju pripravljenosti, glej sliko 6.3, recept sprejme, preveri če je veljaven in nato na TCS pošlje kot odgovor enak recept nazaj. V vseh drugih stanjih, sistem za dostavo odmerka protonskega žarka recept zavrne. TCS nato primerja poslani in prejeti recept med seboj, saj morata biti enaka, če želimo uspešno zaključiti postopek validacije.

Pri postopku validacije recepta simulator omogoča simuliranje obnašanja sistema, če pride do naslednjih napak:

- napačno stanje sistema ob prejetju recepta,
- napačni podatki v receptu,
- napačni odgovor recepta na TCS.

Upravljanje s sistemskimi pogoji

Simulator lahko upravlja s sistemskimi pogoji. Spreminja lahko posamezno stanje systemskega pogoja, ali pa mu nastavi njegovo maskirano vrednost. Delovanje sistemskih pogojev je opisano v poglavju 6.1.3.

Postavljanje sistema v stanje napake

Simulator lahko tudi postavi sistem v stanje napake, ne glede na to, kakšno nalogo v tistem trenutku sistem opravlja. S to funkcionalnostjo se lahko opazuje kako se sistem obnaša ob napakah, v različnih primerih uporabe.

6.2.3 Testiranje

Po končani implementaciji sistema in njegovega simulatorja, sem sprva preveril ali moja implementacija izpolnjuje vse zahteve, ki so mi bile definirane na samem začetku, nato pa sem preveril njegovo delovanje s poganjanjem sistema. Ročno sem preveril več primerov uporabe, kjer sem se najbolj osredotočil na osnovni in najbolj pogost delovni proces.

Ko pri sistemu, ki sem ga ročno poganjal, nisem več opazil napak, sem zanj napisal Unit teste. To so testi, ki testirajo samo funkcionalnost kode razdeljeno na odseke, ponavadi funkcije. Teste sem pisal v JUnit 4 ogrodju, testiral pa sem delovanje, obnašanje ob primeru napak v sistemu in pravilno beleženje sporočil.

6.2.4 Dokumentacija

Ko sem zaključil s testiranjem, sem za mojo implementacijo začel s pisanjem dokumentacije. Napisal sem en dokument, v katerem podrobno opisujem simulator sistema za nadzor dostave odmerka protonskega žarka, njegovo strukturo, funkcionalnosti in delovanje.

6.3 Razvoj

V tem poglavju bom opisal osnovne gradnike, ki sem jih razvil pri izdelavi simulatorja sistema za dostavo odmerka protonskega žarka. Vsi primeri kode, ki se nanašajo na to poglavje, so simbolični.

6.3.1 Končni avtomat

Končni avtomat natančno definira delovanje sistema za dostavo odmerka protonskega žarka; glej poglavje 6.2.1.

Prehajanje med stanji končnega avtomata sem implementiral s pomočjo enum ² razreda, kot je prikazan na sliki 6.5.

```
/**
 * Stanja, ki se uporabljajo pri koncnem avtomatu
 */
public enum Stanja {
    INICIALIZACIJA_SISTEMA(new StanjeInicilizacije()),
    STANJE_PRIPRAVLJENOSTI(new StanjePripravljenosti()),
    VALIDACIJA_PODATKOV(new StanjeValidacije()),
    IZVAJANJE_ZDRAVLJENJA(new StanjeIzvajanjaZdravljenja()),
    ZAUSTAVLJENO_STANJE(new StanjeZaustavitve()),
    STANJE_NAPAKE(new StanjeNapake());
}
```

Slika 6.5: Enum razred stanj končnega avtomata.

Vsako stanje končnega avtomata predstavlja vmesnik, kot je prikazan na sliki 6.6. Vmesnik ima eno metodo, katere namen je, da poskrbi za vse naloge,

²Enum je poseben tip razreda v jeziku Java, ki lahko vsebuje konstante in metode.

ki jih ima posamezno stanje. Za vsako stanje sem nato naredil nov objekt, ki implementira vmesnik in njegovo metodo. Ker ima vsako stanje svojo nalogo, je pri vsakem njegova implementacija drugačna. Kljub temu, metoda pri vsakem stanju deluje na enak način. Sprva poskrbi za nalogo, ki jo ima posamezno stanje, nato pa z uporabo dovoljenj preveri ali je potreben prehod na novo stanje. Metoda vrne novo stanje, če je to potrebno, v nasprotnem primeru pa vrne samo sebe.

```
/**
 * Definicija stanj koncnega avtomata
 */
public interface Stanje {

    /**
     * Metoda poskrbi za trenutno stanje
     *
     * @return Če je potreben prehod med stanjem avtomata,
     *         se vrne novo stanje, v nasprotnem primeru
     *         se ohrani trenutno stanje
     */
    Stanja poskrbiZaStanje();
}
```

Slika 6.6: Vmesnik za stanja končnega avtomata.

Ker pri implementaciji uporabljam večnitnost, lahko hitro pride do prehoda v napačno stanje. Vsako zahtevo sem zato za prehod med stanji dodal v vrsto. Z novo nitjo se medtem neprestano izvaja neskončna zanka, ki jemlje in izvaja stanja iz vrste. Če je vrsta prazna, se uporabi zadnje pridobljeno stanje. Zanka se izvaja dokler se izvaja sistem za dostavo odmerka protonskega žarka.

Med izvajanjem končnega avtomata, sta za njegovo delovanje pomembna dva objektna razreda, ki držita podatke o zdravljenju in podatke o dogodkih, ki se pojavljajo med zdravljenjem.

Podatki o zdravljenju

Objektni razred s podatki o zdravljenju hrani čas začetka in konca zdravljenja, recept o pacientu ter trenutno plast in točko, ki je na vrsti za obsevanje. Razred je prikazan na sliki 6.7.

```
/**
 * Objekt shranjuje podatke o zdravljenju
 */
public class PodatkiZdravljenja {
    private int plast;
    private int tocka;
    private Recept recept;
    private Instant casZacetka;
    private Instant casKonca;
}
```

Slika 6.7: Objekt s podatki o zdravljenju.

Podatki o dogodkih

Zaradi večje zanesljivosti pri sočasnem izvajanju, uporabljam za podatke o dogodkih nespremenljive objekte. Objekt hrani trenutno stanje končnega avtomata in tip dogodka, ki se lahko med izvajanjem zgodi. Nespremenljiv objekt je prikazan na sliki 6.8.

Med izvajanjem moje implementacije sistema za dostavo odmerka protonskega žarka, lahko pride do štirih glavnih dogodkov: sistem lahko prejme recept o pacientu, pripeti se lahko napaka med izvajanjem sistema, lahko pride do napake pri validaciji recepta ali ukaz s strani simulatorja, za prisilno spremembo v stanje napake končnega avtomata [14]. Tipe dogodkov hranim v enum razredu, kot je prikazano na sliki 6.9.

```
/**
 * Nespremenljiv objekt, ki hrani tip dogodka, ki se lahko
 * zgodi med delovanjem in trenutno stanje koncnega avtomata
 */
@Value.Immutable
public interface PodatkiDogodkov {
    /**
     * @return opcijsko tip dogodka, ki se lahko zgodi
     */
    @Value.Parameter
    @Nullable
    TipDogodka dobiTip();

    /**
     * @return trenutno stanje koncnega avtomata
     */
    @Value.Parameter
    Stanja dobiStanje();
}
```

Slika 6.8: Nespremenljiv objekt s podatki o dogodkih.

6.3.2 Simulator sistema

Simulator nam omogoča nadzorovanje sistema za dostavo odmerka protonskega žarka in njegovega stanja. Delovanje simulatorja je preprosto. Zahtevo prejme, jo obdela in izvede ustrezno akcijo nad sistemom. Med postopkom testiranja in validacije, smo za namen pošiljanja zahtev na simulator uporabljali že narejen splošni grafični vmesnik.

Simulator lahko prejme tri vrste zahtev: nastavljanje vrednosti, nastavljanje stanj in izvajanje akcij.

Nastavljanje vrednosti

Z zahtevo po nastavljanju vrednosti lahko simulator nastavi poljuben sistemski pogoj na poljubno vrednost. Nastavlja lahko maskirane vrednosti sistemskih pogojev ali pa njihovo dejansko vrednost.

Z zahtevo je potrebno podati akcijo, ki jo izvedeš (npr. nastavi maskirano stanje) in sistemski pogoj, ki mu želiš nastaviti vrednost. Metoda, ki se uporablja za ta namen je prikazana na sliki 6.10.

```
/**
 * Tipi dogodkov, ki se lahko zgodijo med izvajanjem
 */
public enum TipDogodka {
    RECEPT_JE_PREJET,
    NAPAKA_DELOVANJE,
    NAPAKA_RECEPT,
    PRISILNA_SPREMEMBA_STANJA;
}
```

Slika 6.9: Enum razred s tipi dogodkov.

```
/**
 * Metoda za nastavljanje vrednosti
 */
public void nastaviVrednost(String akcija, String argument)
```

Slika 6.10: Metoda za nastavljanje vrednosti.

Nastavljanje stanj

Za testiranje pravilnega delovanja sistema C-TCS, se morajo preveriti tudi primeri uporabe, kjer lahko pride do napak. Z zahtevo po nastavljanju stanj, lahko upravljamo s tremi različnimi napakami, ki se lahko pojavijo med validacijo recepta o pacientu; več o napakah v poglavju 6.2.2. Ob prejeti zahtevi simulator nastavi stanje ene izmed spremenljivk, ki jih sistem preverja med samim delovanjem. Če je vrednost spremenljivke resnična, se sistem obnaša, kot da je prišlo do dogodka, ki jo vrednost predstavlja - npr. v primeru nastavljene spremenljivke "soPodatkiReceptaNapacni" na resnično vrednost, se sistem obnaša, kot da vrednosti v receptu niso veljavne.

Z zahtevo je potrebno podati akcijo, ki predstavlja spremenljivko nad katero upravljáš in stanje samo, ki je lahko resnične ali neresnične vrednosti. Metoda, ki se uporablja za ta namen je prikazana na sliki 6.11.

```
/**  
 * Metoda za nastavljanje stanj  
 */  
public void nastaviStanje(String akcija, Boolean stanje)
```

Slika 6.11: Metoda za nastavljanje stanj.

Izvajanje akcij

Simulator lahko kot zahtevo dobi tudi akcijo, ki jo nato izvede nad sistemom. Akcije so predvsem simuliranje interakcije s strani medicinskega osebja. Osebe lahko prične z zdravljenjem, ga začasno ustavi, nadaljuje, prisilno ustavi ali uspešno zaključi. Akcije, ki jih simulator podpira, so tudi obračanje ključa nadzorne konzole CM-100 na položaje Off, Prep in Treat. Dodatna funkcionalnost, ki ni posledica uporabniške interakcije, je prisilno postavljanje sistema za dostavo odmerka protonskega žarka v stanje napake. Simulator se lahko kadarkoli med njegovim delovanjem postavi v stanje napake in opazuje njegovo obnašanje.

Metoda, ki se uporablja za ta namen je prikazana na sliki 6.12.

```
/**  
 * Metoda za izvajanje akcij  
 */  
public void izvediAkcijo(String akcija)
```

Slika 6.12: Metoda za izvajanje akcij.

Poglavje 7

Sklepne ugotovitve

Med pisanjem diplomske naloge sem spoznal, kako pomembno je računalništvo v svetu medicine. Razvoj računalniških sistemov, ki se uporabljajo v medicini, je zahteven proces, pri katerem se je potrebno držati določenih standardov in pravil. Uspelo mi je izpolniti zahteve implementacije in narediti simulator sistema za dostavo odmerka protonskega žarka. Sistem je odgovoren za dostavljanje protonskih žarkov v pacienta. Žarek mora usmeriti na točno določeno mesto v sprejemljivem času.

Simulator je postal del Cosylabovega sistema za nadzor zdravljenja protonske terapije. Trenutno se uporablja kot orodje za testiranje in validiranje sistema C-TCS. Nadzorovanje simulatorja s pomočjo že narejenega grafičnega vmesnika deluje brez napak, vendar za uporabnika njegova uporaba ni samoumevna. Za uporabniku bolj prijazno izkušnjo, bi lahko za upravljanje simulatorja sistema za nadzor zdravljenja protonske terapije razvili svoj grafični vmesnik. Ta bi lahko simulator upravljal s pomočjo preprostih klikov nekaj tipk in prikazoval stanje simulatorja ter sistema C-TCS.

Za razvoj celotnega sistema C-TCS je potrebna ekipa dobro usposobljenih in usklajenih razvijalcev, da naredijo sistem ustrezen za uporabo na ljudeh. Ta terapija se zaradi visokega stroška razvoja zaenkrat uporablja samo v tujini. Glede na to, da je tudi v Sloveniji veliko kandidatov, ki bi to terapijo potrebovali, si želim, da bi bila terapija dostopna tudi pri nas.

Vesel sem, da sem bil tudi sam član tega projekta. Z delom na projektu sem dobil občutek, da sem naredil nekaj dobrega za druge, posledično pa tudi zase, saj sem pri projektu nadgradil svoje znanje.

Literatura

- [1] Cm-100 user manual. Dosegljivo: http://www.ptcusa.com/files/manual/CM100_UM_180228.pdf. [Dostopano: 19.8.2018].
- [2] The dds tutorial - prismtech. Dosegljivo: http://download.prismtech.com/docs/Vortex/pdfs/OpenSplice_DDSTutorial.pdf. [Dostopano: 14.8.2018].
- [3] Iso 13485:2016 revision factsheet. Dosegljivo: https://mail.tuv-sud.com/public/a_14616_VCJT1/file/data/294_tuv-sud-mhs-revision-iso13485.pdf. [Dostopano: 12.8.2018].
- [4] Pinpoint precision. Dosegljivo: <http://www.thehealthjournals.com/protontherapy/>. [Dostopano: 2.9.2018].
- [5] Proton beam therapy scattering versus scanning treatment techniques. Dosegljivo: https://www.varian.com/sites/default/files/resource_attachments/Proton_Therapy_TreatmentTechniques_0.pdf. [Dostopano: 15.8.2018].
- [6] Proton therapy - mayo clinic. Dosegljivo: <https://www.mayoclinic.org/tests-procedures/proton-therapy/about/pac-20384758>. [Dostopano: 28.8.2018].
- [7] Proton therapy - treatment for tumors. Dosegljivo: <https://www.radiologyinfo.org/en/info.cfm?pg=protonthera>. [Dostopano: 28.8.2018].

- [8] Proton therapy treatment. Dosegljivo: <https://www.protominternational.com/proton-therapy/proton-therapy-treatment/>. [Dostopano: 2.9.2018].
- [9] Treatment control system for pt. Dosegljivo: <https://www.cosylab.com/treatment-control-system-for-pt/>. [Dostopano: 20.8.2018].
- [10] Casar Božidar and Strojjan Primož. Radioterapija s protonskimi žarki. *Onkologija : strokovno-znanstveni časopis za zdravnike*, 2018.
- [11] Teresa Cywicka-Jakiel. Proton therapy center at the institute of nuclear physics in kraków - from the eye treatment to the scanning gantry. Dosegljivo: <https://docplayer.net/8879349-Proton-therapy-center-at-the-institute-of-nuclear-physics-in-krakow-from-the-eye-treatment-to-the-scanning-gantry.html>, 2013. [Dostopano: 3.9.2018].
- [12] Christian Haack, Erik Poll, Jan Schäfer, and Aleksy Schubert. Immutable objects for a java-like language. In *European Symposium on Programming*, pages 347–362. Springer, 2007.
- [13] Erika Huffman. Impact of en iso 14971:2012 on medical device risk assessment in the eu. Dosegljivo: <https://www.namsa.com/wp-content/uploads/2015/10/Impact-of-EN-ISC-2013-11-6-2013.pdf>. [Dostopano: 12.8.2018].
- [14] Sebastjan Kozoglav. Pyramid scan dose system simulator - software detailed design, 2018.
- [15] Stephanie Lang and Oliver Riesterer. Modern techniques in radiation oncology. Dosegljivo: <https://www.sps.ch/en/articles/progresses/modern-techniques-in-radiation-oncology-36/>. [Dostopano: 2.9.2018].
- [16] WP Levin, H Kooy, JS Loeffler, and TF DeLaney. Proton beam therapy. *British journal of Cancer*, 93(8):849, 2005.

-
- [17] Wei Li, Ping Tan, and Dong Li. Conceptual design of treatment control system for a proton therapy facility at hust. 2018.
- [18] Ingo Lütkebohle. Iec international 62304 standard. Dosegljivo: https://webstore.iec.ch/preview/info_iec62304%7Bed1.0%7Den_d.pdf. [Dostopano: 12.8.2018].
- [19] Harald Paganetti. *Proton therapy physics*. CRC Press, 2016.
- [20] Hugh Powell. A quick and dirty introduction to zeromq. Dosegljivo: <https://blog.scottlogic.com/2015/03/20/ZeroMQ-Quick-Intro.html>. [Dostopano: 14.8.2018].