

AARHUS SCHOOL OF ENGINEERING

TELEMEDICINSK ROBOTSTYRET ULTRALYDSSKANNING

PROJEKT NR: 15138

BACHELORPROJEKT EFTERÅR 2015

Projektrapport

Forfattere:

Rasmus Holm Laursen [201270843]

Andreas Lauridsen [10571]

Karsten Michaelsen [11007]

Vejleder:

Michael Alrøe

15. december 2015

Abstract

Background

Currently, Denmark needs a solution to the technical problems and expenditures which exist regarding ultrasound scans. The future expenditures are expected to increase and the medical specialists are being centralized. Telemedicinsk Robotstyret Ultralydsscanning has a huge potential to solve these two issues. Among others, this solution has been examined by a master thesis which developed a telemedicine system consisting of a robotized ultrasound. This solution showed, the operation of the robot needed to be more intuitive and an increase of the efficiency could be obtained by joining the relevant functions in one application. The purpose of the project is to develop a product, which enables the user to get an intuitive operation of a Universal Robot UR10 using a Geomagic Touch 6-dof joystick. Furthermore, a PC application containing a video feed of the robot, a view of live ultrasound image and the possibility to regulate the robot on an ad hoc basis. The software of the system is desired to meet the medical authorization.

Methods

The development process of the project is based on the Rational Unified Process. Since the product is medical equipment, medical authorization has been a necessary part of the process. The end product was exposed to usability testing.

Results

A PC application is developed. The application has the following four main features. Firstly, it deals with the communication between the robot and the joystick. Secondly, it contains a video feed of the location of the robot relative to the patient. Thirdly, it televises a live ultrasound image. It is possible for the user to interact with this. Lastly, it is possible to regulate the robot directly using the application.

Conclusion

The PC application satisfies the requirements, which was stated at the beginning of the project. The developed solution is intuitive and effective. This was confirmed by the usability testing. An insufficient initial knowledge of the medical authorization has been a hampering factor in the work of getting the product to meet the requirements within the scheduled time frame.

Resumé

Baggrund

I Danmark er der et behov for en løsning af de tekniske problemer og omkostninger, der er forbundet med udførelsen af ultralydsscanninger. Det forventes, at omkostninger for ultralydsscanninger i fremtiden vil stige samt at specialelægerne centraliseres. Telemedicinsk Robotstyret Ultralydsscanning har et stort potentiale for at løse disse to problemstillinger. Dette er bl.a. blevet undersøgt i et specialestudie, hvor der blev udviklet et telemedicinsk system med en robotstyret ultralydsscanning. Denne løsning viste, at der var et behov for en mere intuitiv styring af robotten samt at have alle relevante funktioner samlet i én applikation. Formålet med dette projekt er derfor at udarbejde et produkt, hvor brugeren har en intuitiv styring af en Universal Robot UR10 med et Geomagic Touch 6-dof joystick. Desuden ønskes én PC applikation indeholdende et videofeed af robotten, visning af live scanningsbillede og muligheden for at indstille robotten ad hoc. Systemets software skal udarbejdes iht. medicinske standarder.

Metoder

Udviklingen af projektet er sket med udgangspunkt i udviklingsprocessen Rational Unified Proces. Medicinsk godkendelse har været en del af denne proces grundet, at der er tale om medicinsk udstyr. Der er anvendt en brugertest af det endelige produkt.

Resultater

Der er udviklet en PC applikation, der håndterer kommunikationen mellem robot og joystick, indeholder et videofeed af robotens position ift. patienten, viser live ultralydsscanningsbillede, som brugeren kan interagere med, og der er mulighed for at indstille robotten direkte i applikationen. Kommunikationen foregår gennem et lokalt netværk. Medicinsk godkendelse er delvist blevet implementeret.

Konklusion

Det udviklede produkt løser de krav, der har været opsat for projektet. Brugertesten bekræfter, at der er skabt en intuitiv og effektiv løsning. Manglende initial kendskab til medicinsk godkendelse har været hæmmende for implementering heraf, da det sideløbende med udviklingen har været nødvendigt med tilegnelse af viden for medicinsk godkendelsen.

Indholdsfortegnelse

Kapitel 1	Forord	1
Kapitel 2	Forkortelser og definitioner	2
Kapitel 3	Indledning	3
3.1	Baggrund	3
3.2	Problemstillinger	4
Kapitel 4	Systemoversigt	5
4.1	Systembeskrivelse	5
4.2	Funktionelle krav	7
Kapitel 5	Metoder og processer	8
5.1	Samarbejdsproces	8
5.2	Udviklingsproces	10
5.3	Brugertest	12
5.4	Medicinsk godkendelse af software	12
Kapitel 6	Medicinsk godkendelse	13
6.1	Lovgivning	13
6.2	CE-Mærkning	14
6.3	Klassificering	14
Kapitel 7	Produktudvikling	16
7.1	Specifikation	16
7.2	Analyse	19
7.3	Systemarkitektur	21
7.4	Softwaredesign	29
7.5	Udviklingsværktøjer	32
7.6	Implementering	32
Kapitel 8	Resultater	34
Kapitel 9	Diskussion	35
Kapitel 10	Konklusion	37
Kapitel 11	Perspektivering	38
	Referenceliste	39

Forord 1

Bachelorprojekt er udarbejdet i perioden 24. august 2015 til 16. december 2015. Dette er det afsluttende projekt for diplomingeniøruddannelsen i Sundhedsteknologi på Aarhus School of Engineering. Projektet er udsprunget af et kandidatspeciale omhandlende telemedicinske ultralydsscanninger, der blev afsluttet d. 1. juni 2015.

Anerkendelse

Bachelorgruppen vil gerne takke følgende personer og virksomheder for deres hjælp og sparring gennem hele projektets forløb:

Cand.Polyt. og lektor **Michael Alrøe** for support og vejledning gennem projektføreløbet.

Cand.scient.med. **Søren Holm Pallesen** for sparring og samarbejde ved diverse relevante møder, og for hjælpen med at få gennemført en brugertest.

Cand.scient.med. **Johannes Hollensberg** for sparring og samarbejde i projektets opstartsfasen.

Overlæge og lektor **Lars Bolvig** for at gennemføre brugertest af det udviklede system i afslutningen af projektet og efterfølgende interview.

Universal Robots supportafdeling for ydelsen af teknisk support på trods af, at det var inden for et område der normalt ikke ydes support.

Forkortelser og definitioner 2

Begreb/Forkortelse	Betydning
ASE	Aarhus School of Engineering
UR10	Universal Robots UR10
GMT	GeoMagic Touch
RE	Robottenhed er den samlede enhed ved patienten, som består af UR10, Ultralydsscanner og Webcam
SE	Styringsenhed er den samlede enhed ved operatøren bestående af GeoMagic Touch og PC applikation
[]	Klammer angiver knapper ved navn. Hvis klammer indeholder tal er det en reference
MoSCoW	Akronym udledt fra det første bogstav i følgende: Must have, Should have, Could have, and Would like but won't get
BDD	Block Definition Diagram
IBD	Internal Block Diagram
RTSP	Real Time Streaming Protocol
RTP	Real-time Transport Protocol
RUP	Rational Unified Process
Telesonografi	Fjerndiagnostik via ultralydsscanninger
MDD	Medical Device Directive: 93/42/ECC
FMEA	Failure Mode Effect Analysis
RPN	Risk Priority Number

Tabel 2.1. Forkortelser og definitioner

Indledning 3

Denne rapport er udarbejdet i forbindelse med det afsluttende bachelorprojekt på diplomingeniøruddannelsen i Sundhedsteknologi ved Aarhus School of Engineering (ASE). Projektet omhandler udviklingen af et system, der kan udføre en telemedicinsk robotstyret ultralydsscanning med tilhørende videofeed og visning af live scanningsbillede. Rapporten er en gennemgang af projektforsøget i form af de løsninger, der er udviklet, analyse af og begrundelse for den enkelte løsning, projektstyring- og udviklingsprocesser samt en opsamling af resultater og en diskussion af disse. Yderligere er der i projektet blevet fokuseret på medicinsk godkendelse.

3.1 Baggrund

I de kommende årtier vil Danmark opleve store demografiske ændringer, hvilket vil medføre færre personer i den arbejdsduelige alder og flere ældre[14]. Denne ændring i demografi vil betyde flere patienter med kroniske lidelser[19], som derved vil medføre et øget pres og øgede omkostningerne for det offentlige sundhedsvæsen[16].

Telemedicin øger muligheden for at levere sundhedsydelser i hele Danmark. Telemedicin giver speciallægerne, der i højere og højere grad bliver centraliseret på enkelte sygehuse, mulighed for at foretage diagnosticering og behandling uden at være fysisk tilstede[18]. Telemedicin giver desuden øget adgang til behandling i udkantsområder, hvilket sikrer en kosteffektiv behandling for både patient og udbyder[13].

Antallet af ultralydsscanninger og derved omkostninger hertil er i de seneste år steget i Danmark[17]. Da ultralydsscanninger er en kombination af styring af en probe og realtidsanalyse af ultralydsbilledet, kræver det en specialist at foretage både scanningen og analysen[15].

Telesonografi kombinerer telemedicin og ultralydsscanninger. Specialisten har på baggrund af ultralydsscanningen mulighed for at foretage en diagnostik og muligvis igangsætte en behandling uden at være fysisk tilstede[15].

Ovenstående problemstillinger belyste Johannes Hollensberg og Søren Holm Pallesen under deres speciale i Biomedicinsk Teknik i foråret 2015 (Se Bilag 2). Specialet udviklede to telesonografiske system-prototyper, en robotstyret og en assistentstyret og sammenlignede disse i et komparativt feasibilitetsstudie. Den robotstyrede telesonografiske system-prototype bestod af tre separate programmer:

- Styringsapplikation der tillader specialisten at styre UR10 med påmonteret ultralydsscanner via et fly-joystick.

- Teamviewer (fjernskrivebord) hvor specialisten kan se og interagere med ultralydsscanneren.
- Skype (videofeed) hvor specialisten kan se Universal Robots UR10's (UR10) bevægelse.

Specialet opnåede en løsning, hvor specialisten kan foretage en ultralydsscanning af patienten uden at være fysisk tilstede. For at opnå en mere funktionel løsning henviser specialet til at der videreudvikles på fjernskrivebord, videofeed, tilføres force feedback fra UR10 til joystick og at der udvikles en mere intuitiv styring af UR10.

3.2 Problemstillinger

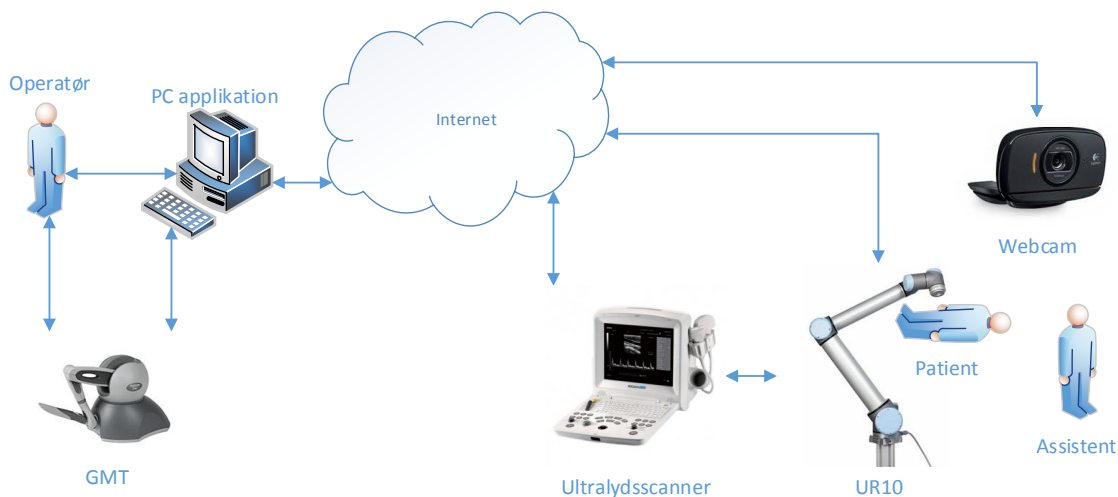
Med udgangspunkt i projektets baggrund og projektbeskrivelse fra ASE (Se Bilag 1) er der defineret følgende problemstillinger, som skal løses og belyses gennem udførelsen af bachelorprojektet:

1. Udvikling af én samlet applikation, der indeholder følgende:
 - Intuitiv styring af UR10
 - Styring af UR10 med Geomagic Touch(GMT)
 - Anvendelse af haptisk feedback i GMT på baggrund af force feedback fra UR10
 - Fjernskrivebord der viser en ultralydsscanning
 - Videofeed der viser UR10's position
2. Medicinsk godkendelse af den udviklede applikation, der overholder følgende standarder:
 - Kvalitetssikringssystem - DS/EN ISO 13485
 - Risikohåndtering - DS/EN ISO 14971
 - Software for medicinsk udstyr - DS/EN 62304

Systemoversigt 4

4.1 Systembeskrivelse

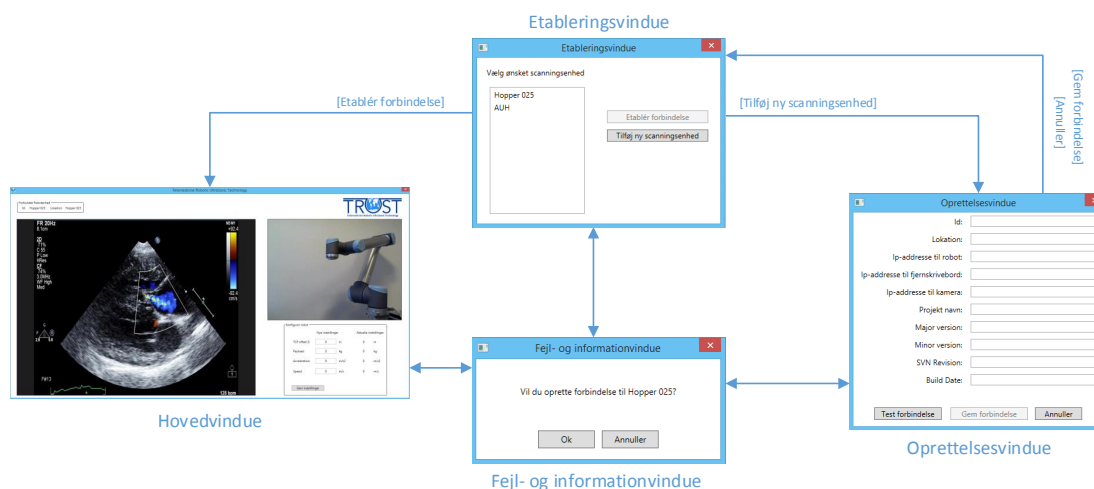
Telemedicinsk Robotstyret Ultralydsscanner er et hjælpemiddel til at udføre Telesonografi (Se figur 4.1). Via to enheder har Operatøren mulighed for at foretage en ultralydsscanning på en Patient uden at være fysisk tilstede. For at foretage en ultralydsscanning af Patienten anvender Operatøren et GMT-joystick, der er forbundet til PC applikationen som styrer UR10 med en påmonteret Ultralydsscanner. Under ultralydsscanningen er der en Assistent til rådighed for at supportere Operatøren og give Patienten trygge rammer. Da ultralydsscanningen foregår uden Operatøren er fysisk tilstede, har denne mulighed for at orientere sig om UR10's position via et videofeed fra Webcam, samt se og interagere med Ultralydsscanneren via fjernskrivebord i PC applikationen. Ydermere har Operatøren mulighed for at sende konfigurationer til UR10, som bl.a. indstille acceleration og hastighed. Forbindelserne mellem disse elementer er sikret via et fejlhåndteringssystem i PC applikationen, der informerer Operatøren om fejl, der er forekommet, og giver Operatøren en mulighed for at handle herefter. Der er vedlagt en videodemonstration af systemet i Bilag 14.



Figur 4.1. Rigt billede af Telemedicinsk Robotstyret Ultralydsscanner, der beskriver systemets opbygning og hvorledes de enkelte elementer interagerer.

4.1.1 PC applikation

PC applikationens vinduer og flow mellem vinduerne kan ses på figur 4.2. PC applikationen åbner i *Etableringsvindue*, hvor der bliver præsenteret en liste over robotenheder, som der kan oprettes forbindelse til. Hvis den ønskede Robotenhed (RE) findes i listen kan denne vælges og der oprettes forbindelse ved at trykke på [Etablér forbindelse], ellers kan en ny RE tilføjes ved tryk på [Tilføj ny scanningsenhed], hvorefter *Oprettelsesvindue* åbner. Når der trykkes på [Etablér forbindelse], åbnes et *Fejl- og informationvindue*, hvor operatøren skal verificere, at det er den korrekte RE, der oprettes forbindelse til. Hvis RE verificeres, startes en proces der etablerer forbindelse til UR10, GMT, fjernskrivebord og videofeed. Hvis der kan oprettes forbindelse, åbnes *Hovedvindue*, ellers bliver der informeret om den pågældende fejl i *Fejl- og informationvindue*.



Figur 4.2. Beskrivelse af vinduerne i PC applikationen og flowet mellem disse. PC applikationen åbner i *Etableringsvindue*, hvor der kan oprettes forbindelse til en RE og *Hovedvinduet* åbner ellers tilføjes en ny RE ved at åbne *Oprettelsesvindue*. Alle vinduer er forbundet til *Fejl- og informationvindue*, da vinduet håndterer visualisering af fejl og information.

I *Oprettelsesvindue* kan der tilføjes en ny RE ved at indskrive påkrævede informationer og herefter trykke på [Test forbindelse]. Herved testes de indtastede informationer ved at oprette forbindelse til UR10, GMT, fjernskrivebord og videofeed. Hvis der kan oprettes forbindelse, bliver de forbundne enheders information verificeret ift. indtastede værdier. Er der overensstemmelse, informeres der om at forbindelsen er godkendt i *Oprettelsesvindue* og [Gem forbindelse] aktiveres for knaptryk.

Hovedvinduet åbner efter der er oprettet forbindelse i *Etableringsvinduet* og der bliver i vinduet opdateret et videofeed af UR10's positionering ift. patient, et fjernskrivebord fra ultralydsscanneren og konfigurationer for UR10. Via fjernskrivebordet kan der indstilles på ultralydsscanneren og via et panel kan der indtastes nye konfigurationer til UR10 som sendes ved at trykke på [Gem indstillinger].

Når systemet kører, er der mange forbindelser, der kan mistes. Her er der implementeret et fejlhåndteringssystem, der opdager fejlen og informerer operatøren i *Fejl- og informationvindue*.

Fejl- og informationvindue viser og beskriver en fejl, der er opstået og giver mulighed for at reagere på fejlen. Vinduet viser desuden information i specificerede tilfælde, eks. når RE skal verificeres.

4.1.2 Afgrænsning

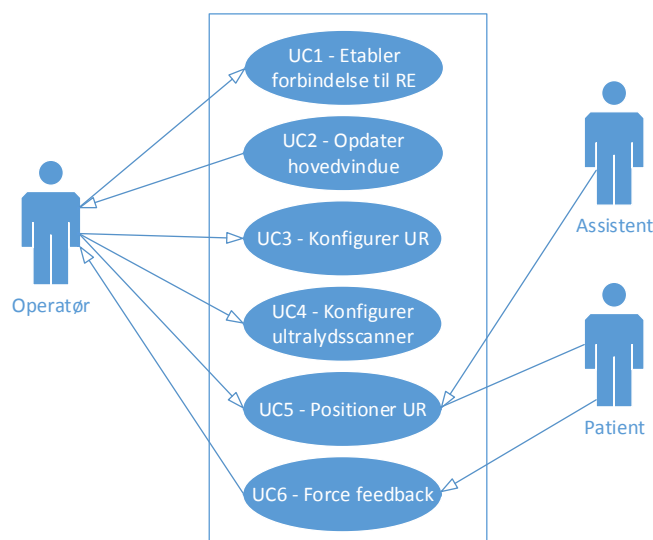
Projektet har ikke haft en ultralydsscanner til rådighed under udvikling og implementering. Dog blev der stillet en ultralydsscanner til rådighed ved den udførte brugertest. Dette betød, at udviklingen og implementering af fjernskrivebord til ultralydsscanneren tog udgangspunkt i, at ultralydsscanneren er en computer med Windows 7. Ultralydsscanneren er derfor blevet simuleret af en anden computer med Windows 7.

Under designfasen blev det besluttet at begrænse løsningen til at arbejde på et lokalt netværk, da målet var at skabe en konceptuel løsning.

I løbet af projektet besluttede gruppen at afgrænse positioneringen af UR10 ved at henlægge rotationen af UR10's hoved og kun bevæge den i et kartesisk koordinatsystem, da andre dele af projektet blev prioriteret.

4.2 Funktionelle krav

De funktionelle krav for Telemedicinsk Robotstyret Ultralydsscanner er defineret ved hjælp af use cases. Use case diagrammet illustreret på figur 4.3 viser de identificerede use cases og aktører, samt deres indbyrdes relation. Der er identificeret seks use cases, der vurderes til at have betydning for udførelsen af telesonografi. Operatøren udvælger en RE og opretter forbindelse (*UC1*), hvorefter hovedvinduet bliver opdateret med ultralydsscanning, videofeed der viser positioneringen af UR10 ift. patienten og konfigurationer for UR10 (*UC2*). Operatøren har i hovedvinduet mulighed for at sende nye konfigurationer til UR10 (*UC3*) og konfigurere ultralydsscanneren (*UC4*). Operatøren kan i hovedvinduet flytte UR10 ved at bevæge GMT (*UC5*) og blive udsat for force feedback i GMT (*UC6*).



Figur 4.3. Use case diagram for Telemedicinsk Robotstyret Ultralydsscanner, der viser systemets funktionaliteter og aktørernes relation hertil.

Metoder og processer 5

5.1 Samarbejdsproces

5.1.1 Projektstyring

Gruppen startede projektforløbet med at tage udgangspunkt i Scrum som projektstyringsmetode. Hertil valgte gruppen at anvende Podio [6] som værktøj til at administrere tidsplan, backlogs, sprints og møder. Da samarbejdet foregik i et meget tæt miljø og det var en naturlig del af arbejdsgangen, at der løbende blev snakket sammen om den individuelle proces, gav det ikke samme værdi for projektet at følge Scrum i det omfang, det var tiltænkt. Gruppen valgte derfor i stedet at benytte sig af milestones (Se Milestones i dokumentet *Projektadministration*) med en varighed på to uger indeholdende de opgaver og mål, der skulle opnås i den pågældende periode. Længden af milestones kunne variere ift. hvilke opgaver, der skulle udføres og de deadlines, der har været opsat i tidsplanen. Tidsplanen er blevet benyttet til at identificere deadlines for elementer og give bedre overblik over, hvor langt gruppen var i processen. Den oprindelige tidsplan er løbende blevet udvidet og opdateret som gruppen har tilegnet sig ny viden omkring projektets elementer og hvordan disse skal udføres. Dette fremgår af den opdaterede tidsplan i dokumentet *Projektadministration*.

Hver fredag blev der skrevet logbog for at få noteret den forgangne uges fremgang, evt. opståede problemer og få fastlagt den efterfølgende uges forventede arbejde. Det fremgår af de konkrete logbøger i dokumentet *Projektadministration*. Gruppen har udarbejdet en samarbejdsaftale, som er blevet anvendt til at sikre enighed om bl.a. forventninger og forpligtelse overfor hinanden og til projektet, som findes i dokumentet *Projektadministration*.

I starten af fasen *Elaboration* (se afsnit 5.2.2) da projektets elementer var identificeret, blev der lavet en arbejdsfordeling således, at medlemmerne fik ansvaret for hver deres område. Samarbejdet i gruppen foregik på tværs af ansvarsområderne, hvor der blev udført sparring og vidensdeling mellem medlemmerne.

	Ansvarsområde
Karsten	<ul style="list-style-type: none"> - Programudvikling af UR10 - PC applikation
Rasmus	<ul style="list-style-type: none"> - Programudvikling af GMT - Programudvikling af fjernskrivebord
Andreas	<ul style="list-style-type: none"> - Programudvikling af videofeed - Medicinsk godkendelse

Tabel 5.1. Fordeling af ansvarsområder for de enkelte medlemmer i projektgruppen. Udarbejdelsen af de enkelte ansvarsområder er forgået på tværs af gruppen.

5.1.2 Møder

Igennem hele projektforsløbet har der været afholdt et ugentligt møde med vejleder Michael Alrøe, hvor projektets status blev gennemgået og problemstillinger, som gruppen stødte på, blev diskuteret. Undervejs i forløbet skulle der tages diverse beslutninger, bl.a. mht. samarbejde med eksterne parter, hvor Michael Alrøe har vejledt gruppen i beslutningsprocessen. Mødereferater kan ses i dokumentet *Projektadministrationen*.

5.1.3 Eksternt samarbejde

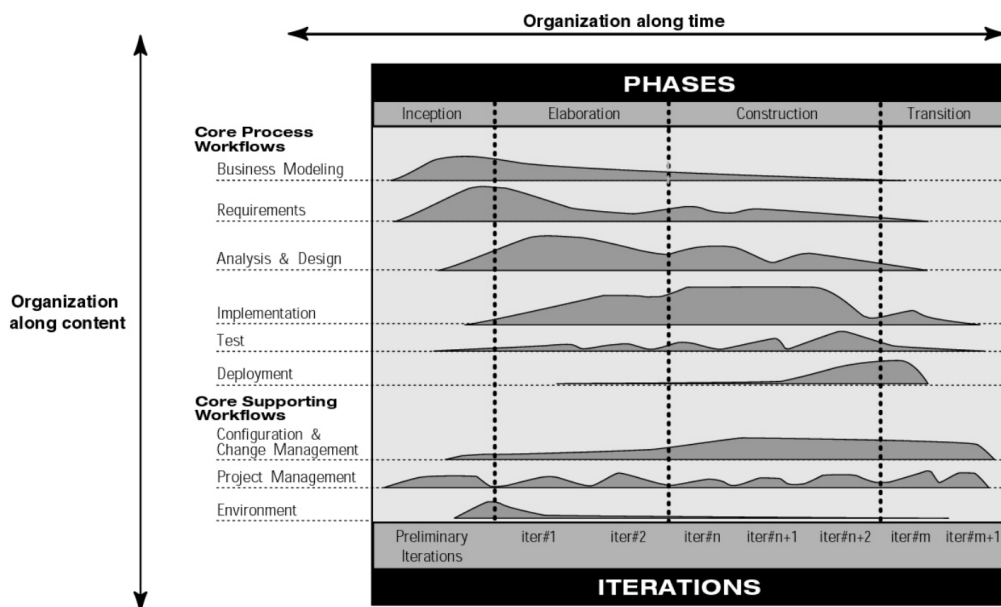
På baggrund af specialet, der har ført til dette projekt, som omtalt i baggrundsafsnittet (Se afsnit 3.1), har Søren Holm Pallesen efterfølgende været igang med at starte egen virksomhed, hvor den telemedicinske robotstyrede ultralydsscanning er omdrejningspunktet. Søren Holm Pallesen har derfor haft en stor interesse i projektet, hvilket har ført til et samarbejde hvor parterne har fungeret som sparring for hinanden. Søren Holm Pallesen har ikke fungeret som kunde, men der har været en naturlig fællesinteresse for det endelige resultat. Medlemmer af projektgruppen har derfor deltaget i enkelte møder arrangeret af Søren Holm Pallesen, som en del af hans opstart af virksomhed. Det drejer sig om et møde med en IT-konsulent i Hobro, der muligvis kunne hjælpe med at løse et problem for gruppen ang. manglende force feedback fra UR10. Derudover var der et møde med Alexandra Instituttet, hvor de deltagende gruppemedlemmer primært fungerede som teknisk støtte for Søren Holm Pallesen.

Gruppen var yderligere repræsenteret til Dansk Ultralyddiagnostisk Selskabs (DUDS) årsmøde, hvor Søren Holm Pallesen og Johannes Hollensberg præsenterede deres speciale og efterfølgende viste løsningen frem med GMT som nyt tiltag til deres eksisterende løsning fra specialet.

Søren Holm Pallesen har også haft stor del i af at få gennemført den brugertest, der blev afholdt i afslutningen af projektet. Igennem hans speciale og videre arbejde har han skabt kontakter, der gjorde det muligt at anskaffe en ultralydsscanner med tilhørende probe og en speciallæge til at udføre selve brugertesten.

5.2 Udviklingsproces

Udviklingsprocessen af Telemedicinsk Robotstyret Ultralydsscanner har fulgt Rational Unified Process (RUP)[7]. Denne proces blev valgt på baggrund af, at der ved projektstart var begrænset domænekendskabet indenfor: UR10, GMT, videofeed og fjernskrivebord. Det har derfor været nødvendigt at arbejde iterativt og med flere sideløbende processer. Figur 5.1 viser hele udviklingsprocessen delt op i fire faser, der hver dækker over flere arbejdsprocesser.



Figur 5.1. Illustration af Rational Unified Process som udviklingsproces for projektet[7].

5.2.1 Inception - Opstartsfasen

I den første fase af udviklingen defineres produktets omfang og afgrænsninger på baggrund af de problemstillinger, der er opstillet i projektoplægget. Dette gøres bl.a. vha. en MoSCoW-analyse, hvor der i samarbejde med interessenter prioriteres hvilke krav og funktionalteter, der bør indgå i denne version af produktet. I den forbindelse laves første udkast af de primære use cases, hvor aktørerne bliver identificeret. Det er ligeledes i denne fase, at de relevante medicinske standarder bliver identificeret.

5.2.2 Elaboration - Uddybelsesfase

På baggrund af MoSCoW-analysen bliver der i denne fase lavet første udgave af udviklingsdokumentationen, hvor produktet vha. UML deles op i realiserbare blokke i en domænemodel, BDD og IBD. Herfra laves kravspecifikationen for at få specificeret produktets funktionelle og ikke-funktionelle krav. De funktionelle krav angives ved fully-dressed use cases. Yderligere udarbejdes en accepttest der definerer hvordan produktets krav skal testes, således der sikres overensstemmelse mellem kravene og den løsning, der udarbejdes. I forbindelse med specificeringen af use cases er der anvendt risikohåndtering til at identificere potentielle farer i produktet og sikre disse gennem krav opstillet i use cases. *Kvalitetssikringssystemet* og *Godkendelse af medicinsk software* er ligeledes dokumenter,

der bliver arbejdet på i denne fase, således det sikres, at der er overensstemmelse mellem de krav som standarderne sætter og de valg/løsninger, der vælges.

Sideløbende med dette begyndes analyseprocessen af interfacet mellem de realiserbare blokke fra BDD og IBD således, at der kan laves arkitektur og design, der definerer de elementer, der skal implementeres. Systemdesignet udføres vha. SysML, hvor der laves pakke- og klassediagrammer og PC applikationens adfærd beskrives via sekvensdiagrammer.

Implementeringen igangsættes delvist i denne fase, da det er en naturlig del af analyseprocessen, hvor der udvikles løsninger for mindre dele af problemstillingen, som testes løbende.

5.2.3 Construction - Konstruktionsfase

Det primære fokus i konstruktionsfasen er udviklingen og implementeringen af arkitekturen og designet fra den foregående fase. Her opdeles udviklingen af systemet i mindre segmenter på baggrund af de realiserbare blokke UR10, GMT, Webcam og Ultralydsscanner, der blev identificeret i analyse- og designprocessen. Denne konstruktionsfase gennemføres vha. flere iterationer, hvor der gennem processen udvikles diverse funktionaliteter til det samlede produkt. Efterhånden som delelementerne bliver udviklet og testet op imod kravspecifikationen, implementeres de i den samlede løsning, hvor der udføres integrationstests.

Til slut valideres produktet ved, at der i første omgang udføres accepttest internt for at sikre, at alle krav er opfyldt, før slutbrugeren inddrages i en brugertest.

5.2.4 Transition - Overgangsfase

I overgangsfasen modnes produktet til slutbrugeren. Dette gøres ved, at der udføres brugertest med fagpersoner inden for det område som produktet henvender sig til. Det gøres for at undersøge brugervenligheden således, at produktet kan finjusteres. Dette skal i sidste ende føre til den første udgivelse af produktet. Detaljer om den udførte brugertest findes i afsnit 5.3. I denne fase er der ligeledes blevet udført ekstern accepttest. Da der ikke er en direkte kunde i projektet, foruden gruppen selv, er denne accepttest blevet udført med vejleder Michael Alrøe som ekstern interessent og repræsentant for ASE.

I projektets forløb har faserne *Elaboration* og *Construction* været mere sammenhængende end RUP angiver dem til, hvorimod arbejdsprocesserne for arkitektur, design og implementering er foregået mere sideløbende. I udviklingsprocessen kom den endelige arkitektur til sent, da der forelå et stort analysearbejde af de forskellige interfaces (Se figur 7.5 s. 22) før en endelig løsning på problemstillingerne blev fundet. Dette skyldes det manglende domænekendskab i starten af forløbet, som gjorde det problematisk at lave arkitektur og design af software. Igennem alle udviklingsfaser er der blevet anvendt interne og eksterne reviews af dokumenter og løsninger. De eksterne reviews er lavet i samarbejde med en anden bachelorgruppe, hvor grupperne har udført reviews på hinandens dokumenter. Det er ikke alle dokumenter, der har været til ekstern review, mens alle dokumenter har været til intern review minimum én gang. Detaljer for review er ligeledes beskrevet i dokumentet *Kvalitetssikringssystem*.

5.3 Brugertest

Brugertesten, der er anvendt til test af Telemedicinsk Robotstyret Ultralydsscanner, bygger på en testopsætning, der giver et virkelighedstro scenarie for telesonografi. Testpersonen er svarende til en slutbruger i form af en specialist indenfor ultralydsscanning. Testen foregår i to adskilte rum, hvor testpersonen er i rummet med SE, mens figuranten vil simulere en patient og være placeret i det andet rum, hvor RE er opstillet. På baggrund heraf vil testpersonen forsøge at foretage en telemedicinsk robotstyret ultralydsscanning. Evalueringen af brugertesten er foretaget i form af kvalitative interviewspørgsmål (Se Bilag 7), hvor testpersonen har mulighed for at komme med sin vurdering og mening om systemet og oplevelsen ved at bruge det.

5.4 Medicinsk godkendelse af software

Den medicinske godkendelse af software er udført med baggrund i de opstillede metoder og processer i standarden DS/EN 62304:2006. Denne standard har ligeledes krav om, at processer fra de to standarder for risikohåndtering DS/EN ISO 14971:2012 og kvalitetssikring DS/EN ISO 13485:2012 anvendes. Foruden de to nedenstående afsnit kan der findes detaljer omkring de anvendte processer i dokumenterne *Risikohåndtering*, *Kvalitetssikringssystem* og *Godkendelse af medicinsk software*.

5.4.1 Identificering af farer

De identificerede farer i forbindelse med Telemedicinsk Robotstyret Ultralydsscanner er identificeret på baggrund af en kombination af metoderne: fishbone-analyse[2] og brainstorm. Der er foruden analyserne undersøgt dokumentation for andre løsninger og hvordan lignende medicinsk udstyr er vurderet. Både fishbone-analyse og brainstorm bygger på at få identificeret en masse farer, hvorefter de mest relevante senere kan udvælges. Disse farer danner baggrund for den videre analyse af systemets risici.

5.4.2 Failure Mode Effect Analysis

Der er i forbindelse med den medicinske godkendelse af softwaren blevet udarbejdet et risikohåndteringsdokument, hvor Failure Mode Effect Analysis (FMEA) er blevet anvendt til at estimere faren og effekten af de identificerede farer for Telemedicinsk Robotstyret Ultralydsscanner. Metoden er designet til at identificere potentielle fejltilstande for det udviklede produkt, fejltilstande i forbindelse med udviklingen af produktet, estimere farerne og prioritere dem efter vigtighed. Yderligere giver metoden anledning til at identificere, hvordan en given fare kan håndteres[1].

Medicinsk godkendelse

6

Det blev besluttet, at medicinsk godkendelse skulle være en del af projektet efter samtaler med Johannes Hollensberg og Søren Holm Pallesen. Gruppen vurderede, at medicinsk godkendelse passede godt ind i projektet, da der er mange overlappende dele, hvor godkendelsen eks. underbygger nogle af de valg af dokumenter, der skal udarbejdes. Det vil desuden være et krav iht. gældende lovgivning, hvis produktet på sigt skal sælges som medicinsk udstyr. Det er på tværs af gruppen blevet besluttet at fokusere på en medicinsk godkendelse af softwaren pga. en stor del af projektet har software som omdrejningspunkt og standarden DS/EN 62304:2006 blev derfor valgt. I denne standard er der desuden krav om, at der udarbejdes en risikohåndtering (DS/EN ISO 14971:2012) og et kvalitetssikringssystem (DS/EN ISO 13485:2012). Selvom den medicinske godkendelse primært er fokuseret på software, tager risikohåndtering hele Telemedicinsk Robotstyret Ultralydsscanner i betragtning.

6.1 Lovgivning

I Medical Device Direktive: 93/42/ECC (MDD) (se Bilag 5) er medicinsk udstyr defineret som følgende:

«»Medicinsk udstyr«: Ethvert instrument, apparat, udstyr, software, materiale eller anden genstand anvendt alene eller i kombination, herunder software, som af fabrikanten er beregnet til specifik anvendelse til diagnostiske eller terapeutiske formål, og som hører med til korrekt brug heraf, og som af fabrikanten er beregnet til anvendelse på mennesker med henblik på:

- a) Diagnosticering, forebyggelse, overvågning, behandling eller lindring af sygdomme,*
- b) diagnosticering, overvågning, behandling, lindring af eller compensation for skader eller handicap,*
- c) undersøgelse, udskiftning eller ændring af anatomen eller en fysiologisk proces, eller*
- d) svangerskabsforebyggelse,*

og hvis forventede hovedvirkning i eller på det menneskelige legeme ikke fremkaldes ad farmakologisk, immunologisk eller metabolisk vej, men hvis virkning kan understøttes ad denne vej.»

Med udgangspunkt i denne definition vil Telemedicinsk Robotstyret Ultralydsscanner kategoriseres som medicinsk udstyr pga. punkt a) og b). Det vil ligeledes være MDD der er den gældende lovgivning for systemet, da systemet går under bekendtgørelsen om medicinsk udstyr. De valgte standarder for risikohåndtering og kvalitetssikring er begge af typen ISO, hvilket sikrer overensstemmelse med MDD's krav.

6.2 CE-Mærkning

Medicinsk udstyr kræver en CE-mærkning for, at det frit kan sælges og markedsføres i hele EU. En sådan mærkning opnås ved at opfylde rammedirektivet MDD. En måde at opfylde kravene heri er ved anvendelse af en harmoniseret standard. Opnåelse af en CE-mærkning forudsætter, at det pågældende udstyr har et tilstrækkeligt højt sikkerhedsniveau, er egnet til dets formål og fordelene ved brug af udstyret overstiger ulemperne. Der er ligeledes krav til udstyret i form af kvalitetssikring (eks. materialesikkerhed og mærkning af udstyr).

Tildeling af CE-mærke sker ved anvendelse af et bemyndiget organ, som er udpeget af myndighederne til at sikre at producenten bag et givent udstyr har den påkrævede tekniske dokumentation og kvalitetssikring. Det bemyndigede organ erklærer, at der på baggrund af dokumentationen er overensstemmelse med gældende lovgivning og påsætter CE-mærkning. Heraf vil det bemyndigede organ fremgå i form af et registreringsnummer, således det altid er muligt at spore, hvem der har været det bemyndigede organ. Det er dermed ikke en "godkendelse" af udstyret, når det modtager en CE-mærkning, men blot en erklæring om, at det ud fra dokumentation lever op til gældende lovgivning. En CE-mærkning kan og må ikke tolkes som et kvalitetsstempel.

6.3 Klassificering

Medicinsk udstyr klassificeres i følgende klasser: I (Is, Im), IIa, IIb og III. Denne klassificering sker på baggrund af udstyrets tilsigtede anvendelse såsom funktionalitet og risici forbundet med brug. Klassificering af det medicinske udstyr er ligeledes afhængig af, om det er et invasivt eller ikke-invasivt udstyr, aktivt eller ikke aktivt udstyr, om det er tiltænkt kortvarig, midlertidig eller langvarig anvendelse, har diagnostisk eller terapeutisk formål.

Telemedicinsk Robotstyret Ultralydsscanner klassificeres som en klasse IIa, da den er tilsluttet en ultralydsscanner, som er kategoriseret som en "Aktiv anordning", hvilket altid vil høre under klasse IIa, medmindre specielle tilfælde er gældende (se MDD Bilag III Regel 10). På baggrund af klassificering er det påkrævet at opfylde Bilag IV, V, VI og VII fra MDD. I stedet for disse bilag kan der med fordel anvendes en fuld kvalitetssikring (Bilag II fra MDD), hvilket ofte er at fortrække for producenten, da det giver en god sikkerhed i det udviklede produkt. Det sikrer overensstemmelse med alle opstillede krav i MDD og hvis der ikke anvendes Bilag II, skal producenten kunne argumentere herfor. I begge tilfælde skal dokumentationen verificeres af det bemyndigede organ inden der kan påsættes CE-mærke på produktet. For Telemedicinsk Robotstyret Ultralydsscanner er der valgt at anvende standarden DS/EN ISO 13485:2012, som sikrer fuld kvalitetssikring af produktet ift. Bilag II fra MDD. Dette er gjort med visse forbehold, da det eks. ikke er muligt at etablere den markedsovervågning, der er påkrævet, når det ikke er et produkt til salg og markedsføring, der udvikles i bachelorprojektet.

Der er for medicinsk software andre regler end ved almindelig medicinsk udstyr. Softwaren kræver sin egen klassificering, som bygger på nogle andre kriterier end de overnævnte. Medicinsk softwareklassificering er som følgende:

- A - Ingen mulighed for skade/sundhedsfare
- B - Ikke-alvorlig skade er mulig
- C - Død eller alvorlig skade er mulig

Softwaren for Telemedicinsk Robotstyret Ultralydsscanner kan igangsætte hændelser, der vil medføre skade i form af tryk/slag. Den vurderes derfor til at være en klasse B. Da forskellen i det dokumentationsarbejde, der ligger bag klasse B og C, er lille, vil der arbejdes ud fra en klassificering på niveau C.

Produktudvikling 7

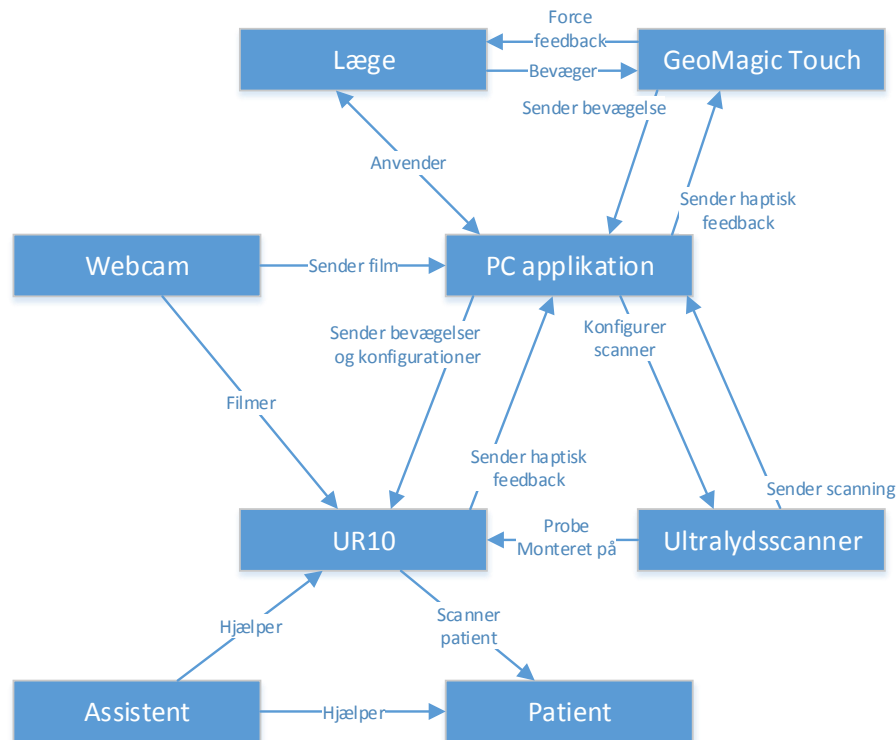
7.1 Specifikation

Specifikationer af produktet blev bestemt på baggrund af de opstillede krav i projektbeskrivelsen udleveret af ASE, gruppens egne ønsker og samtaler med Søren Holm Pallesen og Johannes Hollensberg. Igennem denne process blev der først anvendt MoSCoW-metoden til prioritering af krav og ideer til projektet og dets udarbejdelse. Den udarbejdede MoSCoW-analyse kan ses nedenfor i tabel 7.1.

Must have	<ul style="list-style-type: none">- Positionering af UR10 via TCP/IP med GMT- Haptisk Feedback fra UR10 til GMT- Integrere en fjernskrivebordsløsning til live scanningsbillede i programmet- Integrere live videofeed fra webcam i programmet
Should have	<ul style="list-style-type: none">- Bruger kan vælge hvilken RE, der ønskes forbindelse til fra en liste- Justering af UR10's indstillinger direkte på Hovedvinduet. Eks. hastighed og acceleration- Medicinsk godkendelse af software
Could have	<ul style="list-style-type: none">- I Hovedvinduet skal det være muligt at forstørre de enkelte vinduer til videofeed og fjernskrivebord
Won't have	<ul style="list-style-type: none">- Design af ultralydsprobemodell med knapper, som kan fastgøres på GMT joystick

Tabel 7.1. Resultat af udarbejdede MoSCoW-analyse, hvor *Must have* er de elementer som har højest prioritet, mens *Won't have* har laveste prioritet igennem projektet.

Med afsæt i MoSCoW-analysen, specielt områderne *Must have* og *Should have*, blev der udarbejdet en domænemodell til at illustrere på et overordnet niveau, hvordan det udviklede produkt skal fungere, se figur 7.1.



Figur 7.1. Konceptuel beskrivelse af Telemedicinsk Robotstyret Ultralydsscanners funktionalitet i form af en domænemodel, som anvendes i forbindelse med specificering af systemet.

Domænemodellen beskriver på et konceptuelt niveau, hvilke funktioner og blokke det udviklede produkt skal indeholde. Disse funktionaliteter er blevet yderligere behandlet og opstillet i form af use cases i kravspecifikationen, hvor figur 4.3 viser systemets use case diagram. Hver enkelt use case er defineret i form af fully dressed tabeller. Der er på figur 7.2 vist et eksempel for use case *UC1 - Etabler forbindelse til RE*, som omhandler oprettelse af forbindelse mellem RE og SE.

UC1 - Etabler forbindelse til RE	
Goal	Skabe forbindelse mellem RE og SE
Initiation	<i>Operatør</i>
Actors and Stakeholders	<i>Operatør</i>
References	Figur 4.1, 4.2 og 4.4.
Number of concurrent occurrences	1
Precondition	<ul style="list-style-type: none"> - GMT er tilsluttet PC på SE - RE er opstartet
Postcondition	Systemet er opstartet.
Main Scenario	<ol style="list-style-type: none"> 1. <i>Operatør</i> opstarter styringsapplikation på SE. 2. Systemet åbner etableringsvindue, se Figur 4.1. 3. <i>Operatør</i> vælger ønsket RE fra RE-listen. 4. Systemet aktiverer [Etabler forbindelse] for knaptryk. 5. <i>Operatør</i> trykker på [Etabler forbindelse]. 6. Systemet åbner bekræftigelsesvinduet, se Figur 4.4. 7. <i>Operatør</i> trykker på [OK]. 8. Systemet indikerer oprettelsesprocessen og opretter forbindelse til RE. 9. Systemet lukker bekræftigelses- og etableringsvinduet og åbner hovedmenu, se Figur 4.3.

Figur 7.2. Eksempel på hvordan fully dressed use cases er benyttet i kravspecifikationen til specificering af systemets funktionaliteter. (*Kravspecifikation s. 7*).

I forbindelse med specificering af use cases er der yderligere anvendt viden opnået gennem risikoanalysen og estimeringen. Disse er anvendt til at sikre at systemet har høj sikkerhed ved brug og er ligeledes et krav i forbindelse med medicinsk godkendelse af software. Risikoanalysen tager udgangspunkt i analysemetoden FMEA, som anvendes til at identificere og vurdere farer ved et givent system. De fundne farer blev evalueret ift. en RPN-værdi (Alvorlighed x Sandsynlighed x Detektering) og deres kritikalitet (Alvorlighed x Sandsynlighed). For hele analysen henvises der til dokumentet *Risikohåndtering*, som indeholder den udarbejdede risikohåndtering. Med udgangspunkt i den fundne kritikalitet blev det vurderet hvilke farer, der skulle have højest prioritet. I figur 7.3 nedenfor ses en risikoevalueringsmatrix indeholdende alle de identificerede farer på baggrund af deres kritikalitet. Inddelingen i evalueringsmatrixen er valgt på baggrund af gruppens egne erfaringer og viden omkring risikoestimering.

Sandsynlighed for fare	Alvorlighed af fare					
		Ubetydelig (1-2)	Mindre (3-4)	Alvorlig (5-6)	Kritisk (7-8)	Katastrofal (9-10)
	Hyppig (9-10)				H2.1	
	Sandsynligvis (7-8)				H5.2	
	Lejlighedsvis (5-6)		H1.2, H3.5, H3.7	H1.4, H3.2, H3.4, H4.2	H5.4	H3.1
	Ringe (3-4)		H1.10	H1.7, H1.9, H4.4, H6.2	H1.3, H2.5, H3.6, H3.8, H3.9, H3.10, H5.1, H5.3, H6.3, H6.5, H6.6	H1.1, H1.5, H3.11
	Usandsynlig (1-2)		H1.6, H1.8	H4.3, H4.5, H6.1	H2.2, H2.3, H2.4, H2.6, H4.1, H4.6, H6.4	H3.3

Uacceptabel risiko

Delvis acceptabel risiko

Acceptabel risiko

Figur 7.3. Risikoevalueringsmatrix indeholdende de fundne farer ift. deres kritikalitet. Rød er uacceptable farer, gul er delvis acceptable farer og grøn er acceptable farer for systemet (se *Risikohåndtering* s. 24).

Evalueringsmatrixen er inddelt i tre niveauer (rød, gul og grøn), som er svarende til, hvor kritisk faren er. Derfor er det primært farer fra det røde område, der er håndteret, mens der kun er håndteret enkelte fra det gule område. Det er bl.a. på baggrund af disse farer, at visse krav i kravspecifikationens use cases er opstillet, eks. at operatøren skal holde begge knapper på GMT inde for at kunne positionere UR10, svarende til fare H3.1.

7.2 Analyse

Med udgangspunkt i kravspecifikationen er der fundet områder, som kræver yderligere analyse for at kunne implementeres i systemet. Det er elementer som videofeed, fjernskrivebord, kommunikation mellem PC applikation og UR10 og mellem PC applikation og GMT. Analysen og den fundne løsning af de pågældende elementer er beskrevet i detaljer nedenfor.

7.2.1 Kommunikation mellem PC applikation og UR10

Den indledende gennemgang af Universal Robots[9] dokumentation af UR10 ledte til, at kommunikation med og styring af UR10 kunne foregå via et secondary client interface på port 30002 på UR10's server. Her igennem er det muligt at overvåge UR10's tilstand, bl.a. dens position, da dette konstant streames på denne port. Her er det også muligt ved brug af URScript (Se Bilag 11) at sende kommandoer, der kan få UR10 til at bevæge sig. Starten af projektet gik derfor med at forsøge at opnå systemets funktionaliteter udelukkende vha. secondary client interfacet. På denne måde kan al kontrol af UR10 ligge på PC applikationen uden, at der skal køre et program på UR10.

Under denne udviklings-/analyseproces viste det sig, at det ikke var muligt at hente force feedback ud på port 30002 selvom, der blev holdt møde med en IT-konsulent og ydet support fra Universal Robots supportafdeling. Der skulle derfor findes en anden løsning og det ledte til anvendelsen af modbusserveren på port 502[3]. Modbusen består af en række registre, der kan læses fra og skrives til, hvor det er muligt vha. et program på UR10 at skrive forcen ind i registeret, som PC applikationen derefter kan aflæse.

Kommunikationen foregår derfor som en kombination af secondary client interface og

modbusregistre. UR10's sikkerhedstilstand og aktuelle position aflæses fra data streamen via secondary client interfacet på port 30002. Nye koordinator, som UR10 skal flytte sig hen til, sendes til modbusregistret og den force der er blevet registreret aflæses.

7.2.2 Kommunikation mellem PC applikation og GMT

I projektet skal der anvendes GMT til at styre UR10, hvilket kræver at PC og GMT kan kommunikere. PC applikationen, der skal kommunikere med GMT, er udviklet i C# og Open Haptic Toolkit API til GMT er udviklet i C++. Det var derfor nødvendigt at finde en løsning, der sikrer kommunikation mellem managed code og unmanaged code. Her blev muligheden for at anvende et client-server interface mellem C# og C++ undersøgt. Løsningen, der blev valgt, var Platform Invocation Services(PInvoke), som tillader managed code at lave kald til unmanaged code via et Dynamic link library(DLL). Open Haptic Toolkit består af Haptic Device API(HDAPI) og Haptic Library API(HLAPI). I projektet er der anvendt HDAPI, da denne tillader direkte indstilling af forcen for GMT og HLAPI er blevet fravalgt, da denne anvendes til at udvikle grafiske applikationer.

7.2.3 Videofeed

Med udgangspunkt i Søren Holm Pallesen og Johannes Hollensbergs speciale (se Bilag 2), der har været med til at danne baggrund for projektet, blev der for videofeed'et først undersøgt mulighederne for implementering af Skype, da dette var den løsning, der blev anvendt ved specialet. I specialet blev Skype anvendt som en ekstern applikation. Det blev hurtigt fastslået, at Skype ikke var en mulighed, når visningen af videofeed skulle være en integreret del af PC applikationen, da Skype's API ikke tillader direkte integration, men vil åbne Skype-forbindelsen i egen applikation.

Derfor blev det undersøgt hvilke teknologier og protokoller, der lå bag streaming af video. Gennem denne undersøgelse blev det besluttet at anvende Real Time Streaming Protocol (RTSP), som bygger på Real-time Transport Protocol (RTP), da disse er typiske og generelt anerkendte protokoller indenfor videostreaming i realtid[8]. Yderligere blev det besluttet, at videofeed'et skulle bygge på et client-server forhold, hvor et USB-webcam er tilkoblet en server, som clienten kan tilslutte og afspille fra.

Mange IP-kameraer bygger på RTSP og det vil derfor være muligt i en fremtidig løsning at anvende IP-kameraer istedet for webcams uden, at afspilleren i PC applikationen vil have behov for markante ændringer. Den første løsning, der blev forsøgt anvendt, byggede på Microsoft Expression encoder 4, som både kunne agere server og client. Denne løsning havde den begrænsning, at videofeed'et fik en forsinkelse på 18-20 sekunder, hvilket var en uacceptabel løsning. Derfor blev der undersøgt flere løsninger indtil det blev valgt, at serveren skulle laves ved anvendelse af et bibliotek fra Ozeki[5], som både gør det muligt at streame enkelte webcams, men med mulighed for at flere webcams kan kobles på serveren, således at clienten kan afspille videofeeds fra flere webcams. Desuden er dette et gratis bibliotek, hvor mange af de andre muligheder kræver betaling. Til afspilning af videofeed i PC applikationen er der valgt en løsning, som bygger på open source projektet VLC-player fra videoLan. Denne løsning blev valgt på baggrund af, at det er en afspiller, der har mulighed for at afspille mange forskellige medietyper eks. RTSP. Der anvendes biblioteket nVLC[10], som muliggør implementering af VLC-players funktioner i .Net frameworket, da VLC som udgangspunkt er C++.

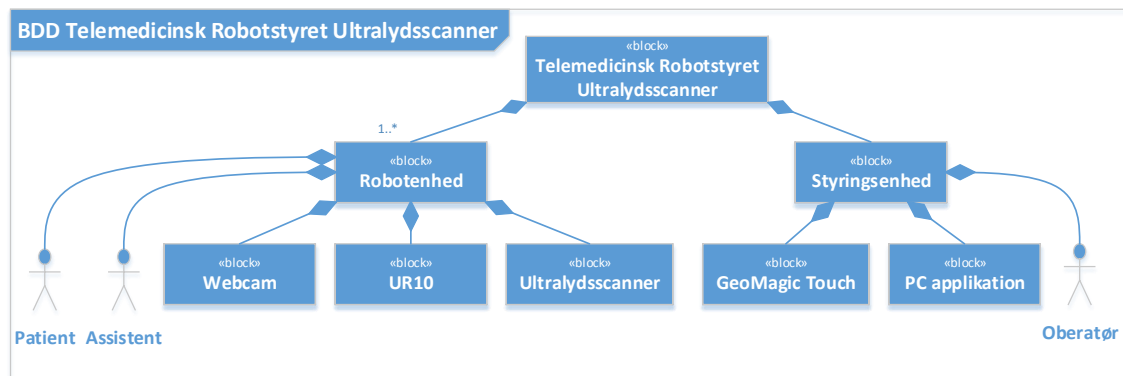
7.2.4 Fjernskrivebord

Løsningen til at dele ultralydsscannerens skrivebord var i Søren Holm Pallesen og Johannes Hollensbergs speciale (se Bilag 2) implementeret i form af fjernskrivebordsløsningen TeamViewer. Derfor var det den første løsning, der blev undersøgt af gruppen. Det viste sig, at denne løsning kan integreres i et Windows Presentation Foundation(WPF) program, men vil åbne i sit eget vindue. På denne baggrund blev løsningen fravalgt, da dette ikke stemte overens med kravene i kravspecifikationen.

Igennem yderligere analyse blev Windows Desktop Sharing API[11] valgt som løsning i projektet, da API'en anvender Windows egne biblioteker og tillader deling af et skrivebord, som kan tilgås via en integreret del af PC applikationen.

7.3 Systemarkitektur

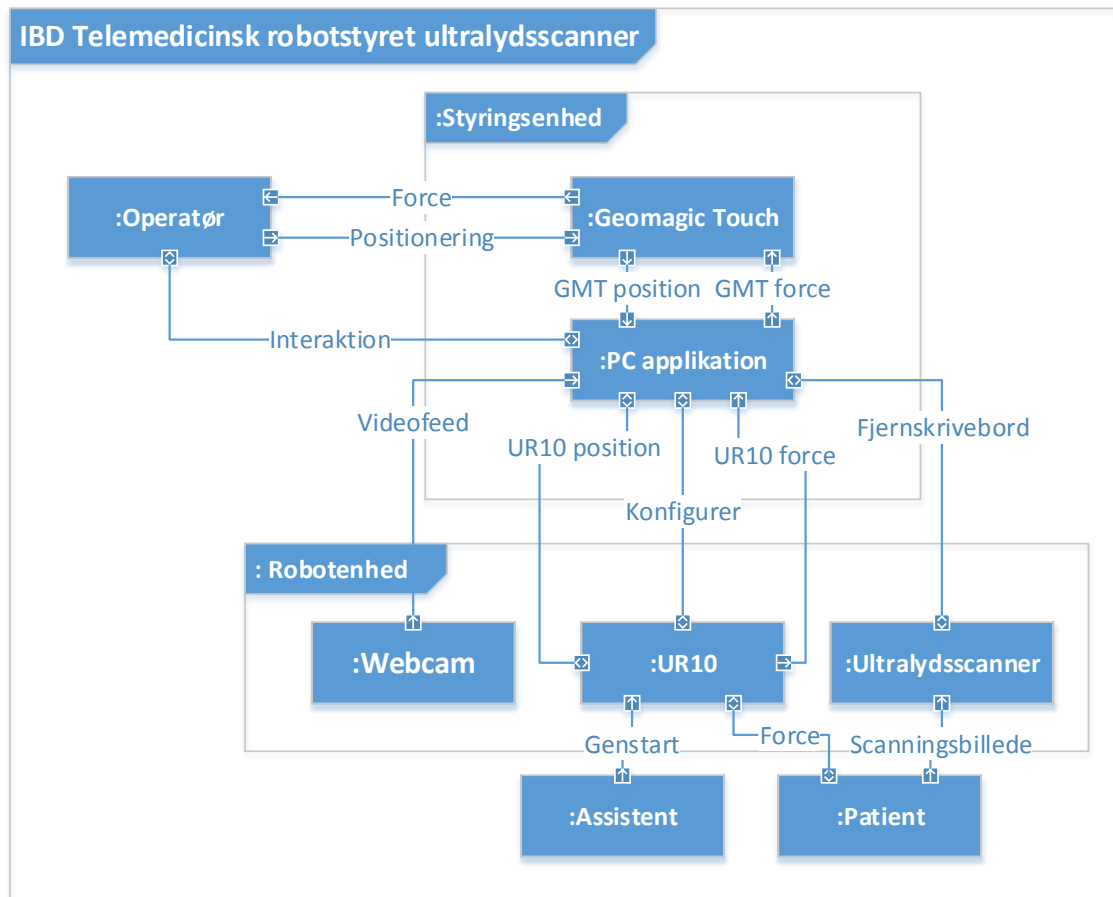
På baggrund af de specificerede krav og den opnåede viden gennem analyse er der for systemets arkitektur udarbejdet forskellige diagrammer, som har til formål, at dele systemet op i mere realiserbare dele.



Figur 7.4. BDD over Telemedicinsk Robotstyret Ultralydsscanner indeholdende systemets opdeling i realiserbare blokke. Figuren viser at systemet består af to hovedenheder i form af Robotenhed og Styrsenhed (*Udviklingsdokumentation s. 2*).

Telemedicinsk Robotstyret Ultralydsscanner er opstillet i ovenstående BDD, hvor det ses, at systemet indeholder to enheder, Robotenhed og Styrsenhed. Robotenheden består af de dele, operatøren ikke har fysisk kontakt med, det gælder blokkene Webcam, UR10 og Ultralydsscanner. Styrsenhed er den del af systemet, operatøren benytter i forbindelse med brug af systemet og består af blokkene GeoMagic Touch og PC applikation.

Detaljer omkring interaktionen mellem de enkelte blokke er beskrevet i figur 7.5. Dette er gjort med et IBD, hvor det kan ses, at PC applikation er den centrale blok for det udviklede system. Denne blok står for al kommunikation mellem de enkelte blokke og har ansvar for at interagere med Operatøren, både i form af at sende force feedback til GMT på baggrund af UR10's force, vise videofeed og vise fjernskrivebordet fra ultralydsscanneren. Webcam's opgave er udelukkende at filme UR10's bevægelser og sende det via et live videofeed til PC applikationen. Det samme gør sig gældende for blokken ultralydsscanner, som står for at sende et live billede af scanningen og gøre det muligt at interagere med scannerens indstillinger via et fjernskrivebord.



Figur 7.5. IBD der beskriver den interne interaktion og interfaces på tværs af de enkelte blokke. PC applikation er ansvarlig for kommunikation mellem Robotenhed og Styringsenhed (*Udviklingsdokumentation s. 3*).

På IBD'et er der foruden de realiserbare blokke vist, hvordan aktørerne i systemet interagerer med de relevante blokke.

7.3.1 System interfaces

På baggrund af systemarkitekturen er der identificeret fire interfaces mellem PC applikation og GeoMagic Touch, UR10, Ultralydsscanner samt Webcam (Se figur 7.5). PC applikationen fungerer som en client, der forbinder til fire servere, en for hver ekstern enhed, hvor den kan sende og/eller modtage data.

Kommunikation mellem PC applikation og UR10

Styring af og kommunikation med UR10 håndteres på flere måder. Selve UR10 programmeres i sproget URScript, som bruges til at styre UR10 på script-niveau. I URScriptet er der flere indbyggede metoder, som bruges til positionering af UR10. Dette gøres ved, at UR10 ses fra dens Tool Center Point(TCP), hvor den kan positioneres i sit Tool-space ift. basen og den kraft der påvirker dens TCP kan registreres. Til positionering af TCP benyttes der følgende funktion:

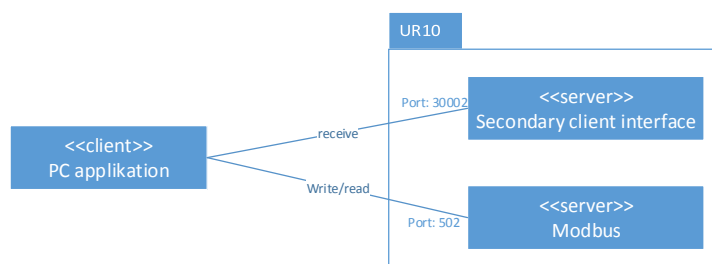
move(pose, a=1.2, v=0.3)	
Parameter	Beskrivelse
pose	Ønskede position
a	Tool acceleration (m/s ²)
v	Tool hastighed (m/s)

Tabel 7.2. Funktionen der anvendes i URScript-programmet til positionering af UR10. En *pose* er givet ved $p[x,y,z,Rx,Ry,Rz]$ og værdierne for acceleration og hastighed er standardværdier, såfremt andet er opgivet. (Se Bilag 11 s. 13).

Funktionen *move* bevæger TCP lineært i Tool-space hen til den ønskede *pose*. En *pose* er givet ved $p[x,y,z,Rx,Ry,Rz]$, hvor x,y,z er positionen af TCP og Rx,Ry,Rz er orienteringen af TCP i en axis-angle rotation (Se Bilag 11 s. 5). Værdierne for acceleration og hastighed angivet i tabel 7.2 er standardværdier såfremt andet ikke er angivet.

Funktionen *get_tcp_force()* returnerer kraftdrejningen i TCP, som er beregnet på baggrund af den fejl, der måles i moment på hver joints for at holde den ønskede bane og det forventede moment. Kraften returneres som en *pose* og er målt i Newton og Newton/Radian.

Ved positionering af UR10 og konfigurerings heraf benyttes der to porte på dens server, når der kommunikeres fra PC applikationen. Kommunikationen på de to porte har hver deres struktur, som er beskrevet nedenfor.



Figur 7.6. Diagram over PC applikationens forbindelse til to servere på UR10. På port 30002 modtages der en data stream med oplysninger om UR10's tilstand. På port 502 læses og skrives der værdier i modbusregistrene. (Udviklingsdokumentation s. 8).

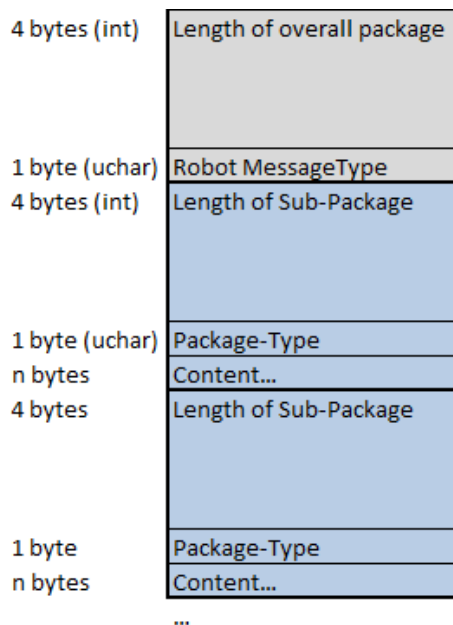
Secondary client interface

UR10 controlleren leverer hele tiden data omkring robotens version og tilstand med bl.a. position etc på port 30002. På denne data stream sendes der to typer beskeder/pakker.

Pakke nr.	Pakketype
16	Robottilstand
20	Versionsbesked

Tabel 7.3. Oversigt over de to beskedtyper der modtages på port 30002. Robottilstand informerer om den tilstand UR10 er i, eks. den aktuelle position. Versionsbeskeden indeholder informationer omkring den installerede software på UR10. (Udviklingsdokumentation s. 8).

De pakker, der modtages, følger en bestemt struktur, hvilket kan ses på figur 7.7. Der modtages en overordnet pakke, der kan indeholde ingen eller flere underpakker. Som figuren viser, afgør de første fem bytes, hvor stor den modtagne pakke er og hvilken af pakketyperne i tabel 7.3, der er modtaget. Længden af pakkerne, der er angivet i de respektive fire bytes, er inklusiv sig selv.



Figur 7.7. Datastruktur over data stream modtaget fra port 30002 (Bilag 6, worksheet DataStreamFromURController).

Når der oprettes forbindelse til UR10, vil den første pakke, der modtages på streamet altid, være versionsbeskeden (pakke nr. 20), hvilket informerer om den software, der er installeret og hvornår den sidst er blevet opdateret. Disse oplysninger skal anvendes sammen med IP-adressen til at kontrollere, at det er den rigtige robot der forbindes til, når der skal testes en ny forbindelse.

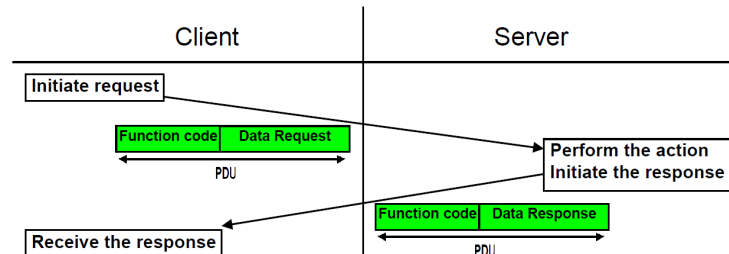
De efterfølgende pakker på streamet er af pakketypen Robottilstand (pakke nr. 16), som består af ti underpakker. Ud af disse ti pakker anvendes der tre specifikke pakker ifm. positionering af UR10. De tre pakker er:

- Pakke nr. 0 - Robot Mode Data
- Pakke nr. 3 - Masterboard data
- Pakke nr. 4 - Cartesian info

Robot Mode Data leverer informationer omkring UR10's tilstand, om den er tændt, kørende osv. Sammen med oplysninger om UR10's sikkerhedstilstand fra pakken Masterboard Data skal disse anvendes til at overvåge om UR10 er i den rette tilstand til at kunne positionere. Positioneringen skal ske på baggrund af UR10's aktuelle position, som kan læses ud fra pakken Cartesian info. Indholdet i denne pakke og de resterende kan ses i Bilag 6, worksheet DataStreamFromURController.

Modbus

Modbusserveren er på port 502 og der kommunikeres via en TCP-forbindelse. Modbusen består af en række registre, der læses fra og skrives til vha. requests. Modbus protokollen definerer en Protocol Data Unit (PDU), der gør at ved en request skal der oprettes en header, der indeholder en funktionskode og oplysninger om de registre, der ønskes at skrive til eller læse fra. Udvekslingen af en request kan ses på figur 7.8.



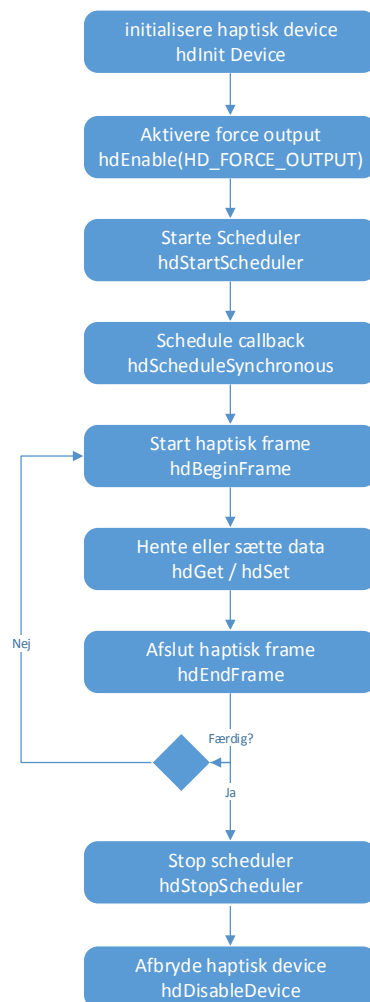
Figur 7.8. Diagram der viser en modbus-transaktion, hvor clienten sender en request i form af en Protocol Data Unit (PDU) med en specifik funktionskode og serveren der responderer med en ny PDU. (Bilag 10 s. 4).

Der anvendes to funktionskoder til kommunikation med modbusserveren, *Read port register* og *Write multiple register*.

Kommunikationen mellem PC applikationen og modbusserveren fungerer ved, at der anvendes et general purpose 16 bits register på modbusen, som ligger fra adresse 128-255. Her skriver PC applikationen UR10's ønskede position ind, som et URScript-program på UR10 læser koordinaterne fra. Ligeledes skriver programmet på UR10 den aktuelle force ind i registeret, som PC applikationen kan læse fra. Derudover er der afsat registre til diverse konfigurationer på UR10. En oversigt kan ses i Bilag 8.

Kommunikation mellem PC applikation og GMT

Kommunikationen med GMT foregår gennem en del af Open Haptic Toolkit[4], som er et redskab udviklet af Geomagic til udvikling af simple haptiske applikationer. I projektet er der anvendt HDAPI, der har en specifik sekvens for kommunikation med GMT, som kan ses på figur 7.9.



Figur 7.9. Specifik sekvens for kommunikation med HDAPI, hvor der kan hentes og sættes værdier ved at kommunikere med en høj prioritets tråd (*Udviklingsdokumentation s. 5*).

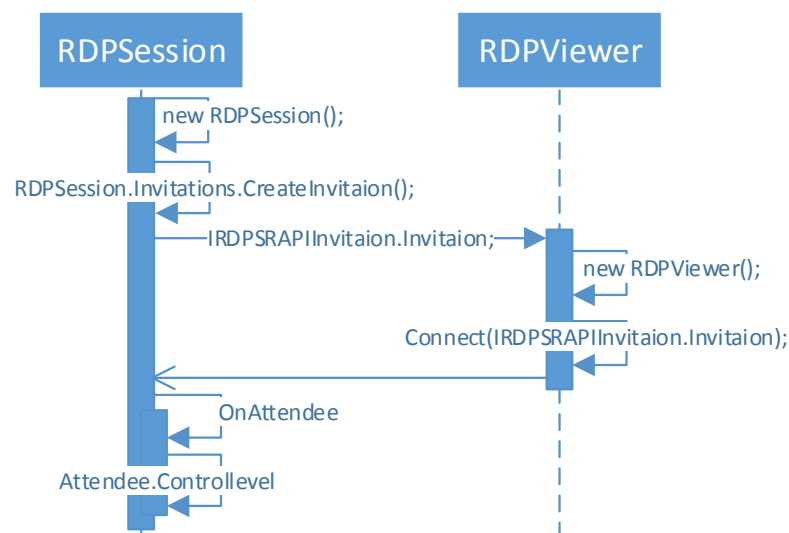
Først initialiseres det haptiske device, hvilket leverer en handle til det tilsluttede device. Herefter aktiveres force output for det tilsluttede device, som tænder alle motorer på GMT. Der startes en Scheduler, som håndterer en højprioritets tråd, der kører med 1000 Hz. For at kommunikere med denne tråd arrangeres der et callback, som starter et nyt frame, hvor det er muligt at hente eller sætte data for GMT, hvorefter frameet afsluttes. Hvis der ønskes at kommunikere yderligere startes et nyt frame ellers stoppes Scheduleren og der afbrydes forbindelse til devicet.

Kommunikation mellem PC applikation og Webcam

Videofeedfunktionen er opbygget med et client-server forhold mellem PC applikationen (client) og en server, hvor der er tilkoblet et webcam. Forbindelsen mellem de to initieres via RTSP, hvorefter selve overførslen af video vil ske via RTP. Serveren benytter et bibliotek fra Ozeki, som kan encode billedet fra webcam og streame det videre. Videoafspilleren i PC applikationen anvender nVLC biblioteket. Det er derfor indbygget i selve afspilleren at decode det modtagne billede fra serveren.

Kommunikation mellem PC applikation og Ultralydsscanner

Kommunikationen mellem PC applikationen og Ultralydsscanneren foregår via et fjernskrivebord. For at opnå denne funktionalitet anvendes Windows Desktop Sharing API. Denne API består af to dele, en RDPSession og en RDPViewer, hvor sekvensen for kommunikationen mellem disse kan ses på figur 7.10. På baggrund af en RDPSession oprettes der et forbindelsesID. Dette forbindelsesID sendes til en RDPViewer, som anvender det til at oprette forbindelse til den specifikke RDPSession. Adgangsniveauet for RDPViewer kontrolleres af RDPSession, hvilket bestemmer om det skal være muligt kun at se eller også at interagere med skrivebordet.



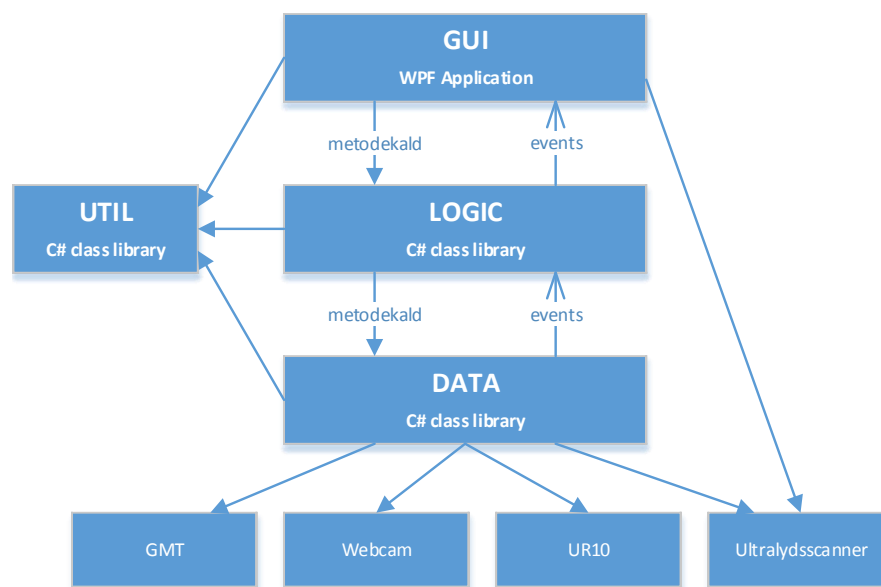
Figur 7.10. Sekvensdiagram der viser sekvensen mellem RDPSession og RDPViewer. RDPSession opretter et forbindelsesID der sendes til RDPViewer, som derved kan forbinde til den pågældende RDPSession (*Udviklingsdokumentation s. 13*).

7.3.2 Softwarearkitektur

Der er blevet anvendt en trelags-model til strukturering af PC applikationens arkitektur, se figur 7.11. Denne arkitektur er valgt på baggrund af de identificerede blokke og deres funktioner, som stemmer overens med filosofien bag modellen. Modellen består af tre lag: GUI, LOGIC og DATA. Hvert lag har et specifikt ansvarsområde i softwaren. GUI håndterer interaktionen med brugeren eks. i form af knaptryk og visning af videofeed. LOGIC er ansvarlig for de logiske operationer i softwaren. Her udføres der bl.a. tjek af, om det er den korrekte enhed, der forbindes til og beregninger for positionering af UR10. DATA håndterer forbindelser til enheder udenfor PC applikationen eks. læses UR10 og GMT data fra de pågældende enheder i dette lag. Yderligere står DATA laget for fejlhåndtering af forbindelser til eksterne enheder. For at sikre lav kobling og høj samhørighed i arkitekturen kan lagene kun kalde metoder nedad, mens kommunikation den anden vej foregår ved events.

En enkel afvigelse fra modellens opdeling er, når RDPVieweren skal forbinde til RDPSessionen, sker dette direkte fra RDPVieweren i GUI-laget.

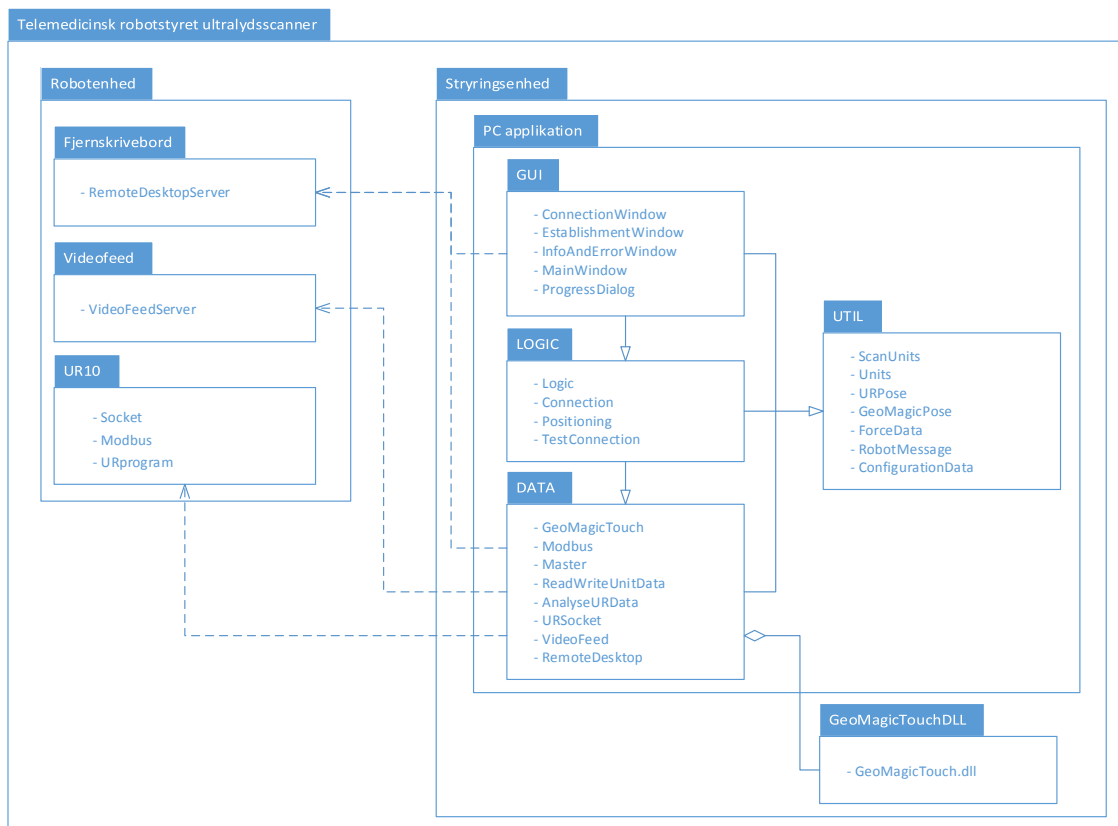
Der anvendes desuden en UTIL klasse til DTO'er for at sende objekter mellem lagene.



Figur 7.11. Illustration af hvordan trelags-modellen er anvendt i softwarearkitekturen. De tre lag har hvert et ansvarsområde: GUI er brugergrænseflade, LOGIC er de logiske beregninger og DATA er kommunikation til eksterne elementer som eks. UR10 (*Udviklingsdokumentation s. 17*).

7.4 Softwaredesign

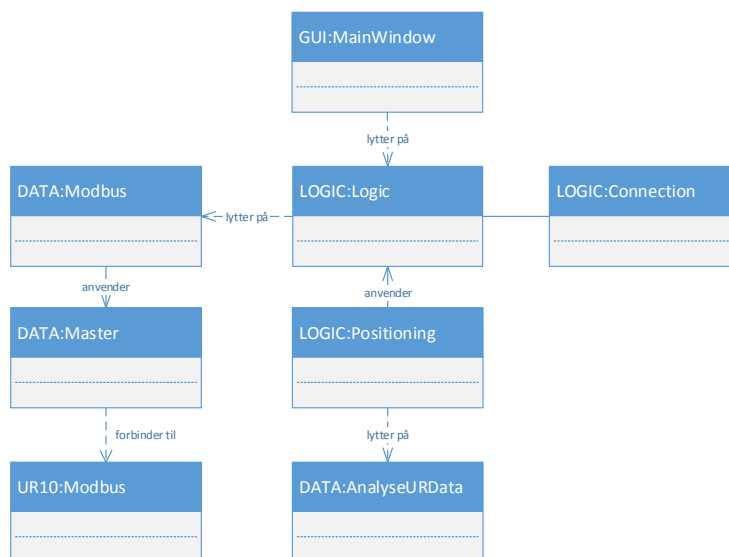
På baggrund af det beskrevne system og softwarearkitekturen er der udarbejdet et pakkediagram, som nedbryder softwaren i realiserbare pakker og klasser, Se figur 7.12.



Figur 7.12. Pakkediagram indeholdende de pakker som Telemedicinsk Robotstyret Ultralydsscanner består af. Hver pakke nedbrydes til pakker, som indeholder klasser, der er implementeret på Stryingsenhed eller Robottenhed (*Udviklingsdokumentation s. 19*).

Pakkediagrammet for Telemedicinsk Robotstyret Ultralydsscanning består af pakkerne Robottenhed og Stryingsenhed. Robottenhed nedbrydes i mindre pakker for hver element i RE, som indeholder realiserbare klasser. Stryingsenhed nedbrydes i pakkerne PC applikation og GeoMagicTouchDLL. PC applikationen nedbrydes i mere specifikke pakker, som er defineret ift. softwarearkitekturen og der identificeres realiserbare klasser for hver pakke. GeoMagicTouchDLL pakken er en DLL, som er udviklet til kommunikation mellem GMT og PC applikationen. Det er værd at bemærke, elementerne i pakken UR10 ikke er deciderede klasser, men er medtaget for at skabe et overblik over softwarens opbygning. Socket svarer til secondary client interfacet, Modbus er modbusregistrene og URprogram er URScript-programmet på UR10.

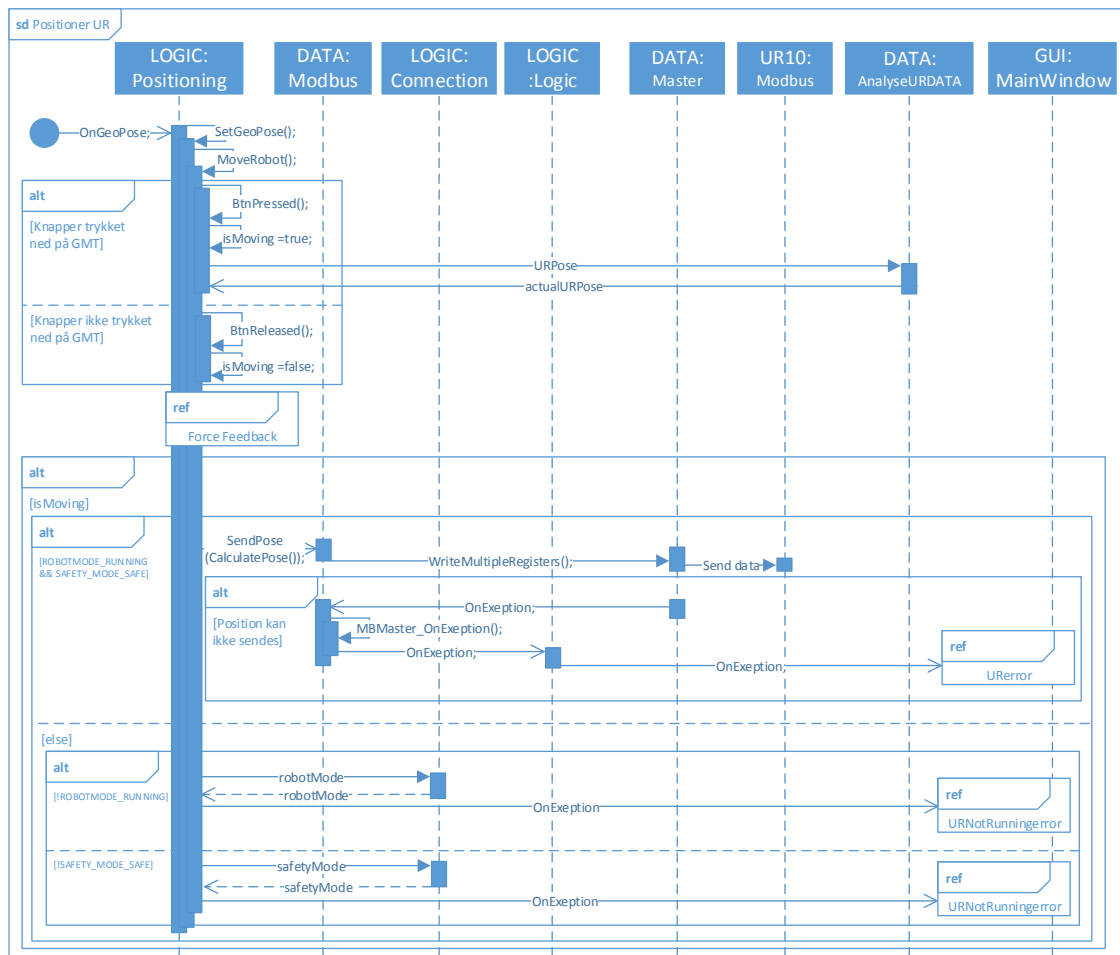
På baggrund af de identificerede klasser i pakkediagrammet på figur 7.12 udarbejdes et klassediagram til at identificere forholdet mellem klasserne. På figur 7.13 ses et udsnit af klassediagrammet og i *Udviklingsdokumentation s. 20* kan det fulde klassediagram ses.



Figur 7.13. Et udsnit fra det komplette klassediagram, der findes i dokumentet *Udviklingsdokumentation s. 20*, som viser klasserne, der har indflydelse på UC5 - Positioner UR.

På baggrund af relationerne, der er defineret mellem klasserne og use cases, bliver der udarbejdet sekvensdiagrammer, der beskriver sekvenserne i systemet. Sekvensdiagrammerne beskriver kald mellem objekter i systemet og anvendes bl.a. til at definere attributter og metoder for de enkelte klasser.

Nedenfor vises et eksempel på et sekvensdiagram, som viser sekvensen for UC5 - Positioner UR. Først tjekkes om knapperne er nede på GMT, hvorefter sekvensen Force Feedback håndterer force feedback mellem UR10 og GMT. Hvis knapperne er trykket ned på GMT hentes den aktuelle position for UR10, som anvendes til at beregne en ny position, der sendes til UR10:Modbus. Hvis der ikke kan sendes en position, opstår der en fejl, som håndteres af URError. Herefter tjekkes om UR10's robotmode og safetymode kan godkendes ellers opstår en fejl, som håndteres af URNotRunningerror. URError og URNotRunningerror håndterer fejl, der opstår i systemet og er beskrevet i sekvensdiagrammer for sig selv.



Figur 7.14. Sekvensdiagram for UC5 - Positioner UR hvor Operatøren har mulighed for at trykke knapper ind på GMT og sende positioner til UR10 (*Udviklingsdokumentation s. 39*).

Sekvensdiagrammerne danner grundlag for at lave detaljerede specifikationer af allerede identificerede klasser. Her noteres Fields, Properties, Methods, Events og Nested types for hver klasse. Nedenfor er der vist et eksempel på et detaljeret klassediagram, som viser LOGIC:Positionering.



Figur 7.15. Detaljerede specifikationer af klassen LOGIC:Positionering, der viser Fields, Properties, Methods, Events og Nested types (*Udviklingsdokumentation s. 47*).

7.5 Udviklingsværktøjer

I projektet er der blevet anvendt Visual Studio 2013 som udviklingsmiljø, der tillader udvikling af applikationer i forskellige sprog som C++ og C#. Gruppen har tidligere erfaring med udviklingsmiljøet.

Til design af brugergrænseflader på PC applikationen er der blevet anvendt WPF, da gruppen har tidligere erfaring med dette. I projektet er der anvendt to Windows Forms komponenter, hvilket medfører, at projektet med fordel kunne være udviklet i Windows Forms.

7.6 Implementering

Implementering af UR10, GMT, videofeed, fjernskrivebord blev udviklet i selvstændige løsninger, hvor der blev foretaget unittest før delene blev implementeret i den endelige løsning. Kommenteret kildekode er anvendt som dokumentation for implementeringen og kan findes som en del af de afleverede udviklingsdokumenter.

Implementeringen af PC applikationen foregik trinvist, hvor et delkomponent blev integreret og godkendt før der kunne arbejdes videre med integrationen af næste delkomponent. Dette fremgår af den opdaterede tidsplan i dokumentet *Projektadministration*.

Til at teste det samlede system blev der foretaget interne accepttest, der ledte til mindre rettelser, før der blev foretaget ekstern accepttest af produktet. Den eksterne accepttest

ledte til, at accepttesten kunne delvist godkendes med ubetydelige anmærkninger eller fejl. Den delvise godkendelse skyldes, at nedenstående punkter ikke kunne godkendes:

- Extension UC3-3a - Systemet kan ikke sende konfiguration til UR10.
- Extension UC3-3a-2a - Operatør trykker på [Annuller] i fejlmeddelelse.

Disse to punkter blev ikke godkendt, da det ikke var muligt at få systemet i et scenarie, hvor fejlen opstod. Dette skyldes, at fejlen der testes, håndteres andensteds af programmet og derved ikke vil blive aktuel.

7.6.1 Brugertest

Den endelige test af Telemedicinsk Robotstyret Ultralydsscanner var en brugertest (Se afsnit 5.3), hvor overlæge på Radiologisk Afdeling, Århus Universitetshospital, Lars Bolvig testede produktet. På figur 7.16 ses opsætningen af de to enheder, hvor Lars Bolvig bruger SE til at styre RE, der foretager en ultralydsscanning af figuranten.



Figur 7.16. Opstilling af brugertest med Lars Bolvig. SE ses til venstre og er tilsluttet til RE til højre via internettet.

På baggrund af testen udtrykte Lars Bolvig stor tilfredshed med oplevelsen. Han kommenterede, at han uden store problemer kunne anvende GMT til at positionere UR10 og at han som normalt kunne styre scanningen ud fra ultralydsscaningsbilledet. Lars Bolvig blev også anvendt af det foregående speciale til at teste deres løsning og nævnte her at den nye løsning er *"Fantastisk meget bedre, kan slet ikke sammenlignes"*, hvilket understøtter, at der er udviklet en mere intuitiv styring og applikation. Til selve styringen og anvendelsen af PC applikationen udtalte Lars Bolvig, at det var *"Næsten som at sidde med ultralydsscanneren i hånden"*. Lars Bolvig ytrede dog også mangler i produktet ift. rotation af UR10-hovedet, da han så dette som essentielt for videreudvikling af produktet. Derudover manglede han en mere realistisk sammenhæng mellem forcen han blev udsat for med GMT og den reelle force i UR10. Lars Bolvig kommenterede, at det var en nødvendighed at have flere kameraer, der viser UR10 fra forskellige vinkler således, at ultralydsscanneren kan placeres korrekt.

Da Lars Bolvig har en personlig interesse i projektet, kan det ikke udelukkes, at der er en bias forbundet med hans udtalelser.

Resultater 8

Projektets resultater kan opdeles i tre områder: Styring af UR10 med GMT, PC applikation og medicinsk godkendelse.

Styring af UR10 med GMT

Det er muligt at styre UR10 ved at anvende joysticket GMT. Her er det muligt at positionere UR10 i et kartesisk koordinatsystem. Det har ikke været muligt at konvertere rotationen fra GMT til en rotation af hovedet på UR10. Der er skabt en flydende bevægelse af UR10 med mindre rystelser. Bevægelser i UR10 sker forsinket, hvis GMT bevæges unaturlig hurtigt. Derudover er det implementeret, at operatøren aktivt skal holde knapper inde på GMT for at positionere UR10.

PC applikation

I projektet er der udviklet en PC applikation, der giver brugeren mulighed for at oprette forbindelse til en RE eller tilføje en ny. Når brugeren har oprettet forbindelse, åbnes et vindue, hvor fjernskrivebord, videofeed og konfigurationer af UR10 er en indbygget del. PC applikationen tillader brugeren at se og justere en ultralydsscanner via fjernskrivebordet. PC applikationen viser UR10's positionering via et videofeed på 150ms delay. PC applikationen tillader desuden brugeren at stille på konfigurationer for UR10. Der er implementeret et fejlhåndteringssystem i PC applikationen, der håndterer fejl relateret til forbindelserne mellem PC applikationen og eksterne enheder.

Medicinsk godkendelse

Der er udført en risikoanalyse, estimering, og håndtering af potentielle farer i forbindelse med det udviklede produkt. Dette er primært gjort på udvalgte farer, som har en speciel høj kritikalitet. Dermed er alle fundne farer ikke håndteret.

Det udarbejdede kvalitetssikringssystem er blevet anvendt i det omfang, hvormed det har givet værdi for projektet. Dele, der omhandler markedsovervågning, køb af eksterne produkter, produktion og serviceydelser, er ikke medtaget. Medicinsk godkendelse af software anvender og opfylder størstedelen af de elementer, som standarderne foreskriver. For alle tre standarder er der udarbejdet separate dokumenter, hvori det er beskrevet, hvordan de pågældende krav er implementeret i projektet. Det udviklede produkt lever endnu ikke op til de krav, der stilles for, at produktet kan godkendes til medicinsk brug.

Diskussion 9

I resultatafsnittet er projektet opdelt i tre afsnit. Det er på baggrund af disse afsnit det følgende diskussionsafsnit er udarbejdet.

Gennem projektet er der opnået løsninger for alle krav, der er defineret i kravspecifikationen. Det udviklede system er begrænset, da der ikke er rotation af UR10's hoved. Det blev pointeret i den gennemførte brugertest, at netop rotationen vil være altafgørende, hvis systemet skal tages i brug. Rotationen i UR10's hoved endte med at blive fravalgt pga., at bachelorprojektet ikke udelukkende bygger på det udviklede produkt, men også processerne bag. Foruden den manglende rotation er det på baggrund af resultaterne muligt at foretage en faktisk ultralydsscanning uden den lægefaglige specialist er tilstede i samme lokale som patienten. Dette blev bekræftet i den gennemførte brugertest.

Gruppen har valgt at fokusere primært på at løse funktionelle krav for systemet. De ikke-funktionelle krav er derfor blevet specificeret i et begrænset omfang. Her kunne der eks. have været specificeret krav til præcisionen for positionering af UR10 og et acceptabelt delay på videofeed og fjernskrivebord. Eksempler som disse bør derfor specificeres i en evt. videreudvikling.

Det har gennem projektet vist sig, at den force feedback som UR10 registrerer er meget svingende, når den udsættes for kræfter under 50 newton. Derfor er der anvendt en midling af den registrerede force, for at mindske de svingninger, der er i målingerne. Den anvendte løsning er en midling over 100 punkter. En 100 punkts midling er nødvendigvis ikke nok til at fjerne alle de svingninger der er, men den har givet en mere flydende bevægelse af GMT og derved også UR10.

Gennem PC applikationen er det muligt for brugeren at se UR10's position ift. patienten, men der er behov for yderligere kameravinkler end den ene, der er nu. Dette vil give en bedre oplevelse og følelse af kontrol over robotens position, når den fjernstyres. Der er som nævnt blevet anvendt en normal PC med Windows 7 til simulering af ultralydsscanneren. Dette viste sig som et mindre problem, da der blev udført brugertest, hvor det udviklede fjernskrivebord blev forvrænget ift. det faktiske skrivebord på ultralydsscanneren. Selvom det ikke er optimalt, var selve scanningsbilledet ikke udpræget påvirket af forvrængningen. Det var derimod de omkringliggende knapper og indstillingsmuligheder, der blev svære at interagere med.

Anvendelse af medicinske standarder har igennem projektforløbet haft blandet succes. Det har fungeret rigtig godt med udførelse af risikohåndteringen, da den indeholder mange elementer, som passer godt ind i projektet. Dette er specielt ved udarbejdelsen af use cases, hvor der netop er medtænkt risikokontrol som en del af disse use cases. Indførelsen af at bruge knapperne på GMT bygger eks. på en identificeret fare.

Derimod er det ikke lykkedes at anvende kvalitetssikring til fulde som ellers var hensigten. Dette skyldes at eks. salg af produktet og efterproduktion ligger uden for projektets rammer. Derfor er der mange af elementerne i kvalitetssikring, der ikke giver mening i projektet. Ligeledes har det været svært at implementere kvalitetssikringssystemet, da det er blevet udviklet sideløbende med udviklingen af systemet, hvorved overensstemmelse mellem de to processer er svær at overholde. Der kunne med fordel have været undersøgt og etableret et kvalitetssikringssystem inden projektet startede og dermed være sikker på, at de processer, der var beskrevet, også var en del af projektets processer fra opstarten. Den medicinske godkendelse af softwaren er naturligvis påvirket af, at netop kvalitetssikringen er mangelfuld og produktion kan derfor ikke på nuværendes tidspunkt godkendes som medicinsk udstyr. Dog er der stadig anvendt mange af de processer som standarden forskriver og den dokumentation, der kræves af standarden, er ligeledes blevet udført. Derfor er det generelt for alle de standarder, der er fulgt, at hvis der tidligere havde været fokus på disse og deres krav, ville der formentlig kunne have været udviklet et produkt, der lå tættere på en medicinsk godkendelse, end det er tilfældet.

De anvendte samarbejdsprocesser har medført visse udfordringer i forbindelse med udarbejdelsen af projektet. Specielt de dele, der har omhandlet Scrum og milestones. Her har gruppen ikke været gode nok til at anvende værktøjerne i en sådan grad, at det har givet værdi til projektet. Der er ingen i gruppen, der tidligere har arbejdet med Scrum som udviklingsværktøj i et projekt af denne størrelse. Milestones virkede som en god idé og som en proces, der ville give værdi til projektstyringen. Implementeringen og anvendelsen af milestones har ikke været god nok, da de ofte har været mangelfulde og det faktum, at der ikke er blevet fulgt ordentligt op på de enkelte opgaver defineret i dem.

Udviklingsprocessen med RUP har derimod fungeret godt og har givet værdi til de enkelte udviklingsfaser, som projektet har gennemgået.

Konklusion 10

Der er skabt én PC applikation, hvorigennem en speciallæge har mulighed for at udføre en telemedicinsk ultralydsscanning. Denne styring forgår via et GMT joystick således, at operatøren kan positionere UR10. Projektet har skabt en intuitiv løsning ved at inkludere et fjernskrivebord, hvor ultralydsscanningen kan håndteres, et videofeed, der viser UR10's position samt et haptisk feedback i joysticket på baggrund af UR10's bevægelser. Gennem en brugertest er det blevet bekræftet, at det er en mere intuitiv løsning end de, der allerede findes på markedet.

Det haptiske feedback, som kan registreres i UR10, er dog ikke præcist nok til, at der kan laves en løsning, hvor der et 1:1 forhold mellem den faktiske påvirkning på UR10 og det feedback, som brugeren føler i GMT. Det blev ved brugertesten påpeget, at den manglende rotation i UR10's hoved er en essentiel funktion for at kunne fortage en fyldestgørende ultralydsscanning.

Det implementerede videofeed er lavet med udgangspunkt i ét kamera. Det er efter brugertest kommet frem, at der er behov for yderligere kameravinkler end denne ene. Yderligere viste brugertesten et problem med det udviklede fjernskrivebord, der muliggør interaktion og visning af ultralydsscanningen, da billedet fra ultralydsscanneren blev forvrænget i en sådan grad, at det var til gene for brugeren.

Der er anvendt medicinske standarder ift. medicinsk godkendelse i det omfang, hvor det har givet værdi og mening for projektet. Flere elementer fra kvalitetssikringssystemet har ikke kunnet implementeres, hvorimod risikohåndtering er blevet anvendt i langt større grad. På den baggrund kan standarden for medicinsk software ikke opfyldes, hvilket medfører, at det udviklede produkt ikke opfylder de krav, der er nødvendige for at opnå en medicinsk godkendelse.

En vigtig videreudvikling for produktet er at få tilføjet rotation i UR10's hoved. Problemstillingen består i at få lavet en transformationsmatrix, der skal beregne en rotation i joysticket på baggrund af vinklerne i GMT's led. Disse beregninger skal derefter transformeres til en axis-angle rotation, der passer på leddene i UR10 således, at GMT og UR10 får en identisk rotation og dermed øger operatørens bevægelsesfrihed.

For at operatøren skal opnå et fuldt overblik over UR10's orientering ift. patienten, kræver det, at der kommer mere end én kameravinkel på det videofeed, operatøren får via PC applikationen. På nuværende tidspunkt kan operatøren komme i en position, hvor det er nødvendigt, at assistenten hjælper operatøren videre. Med et bedre overblik kan operatøren blive mere selvstændig i denne proces og potentielt øge effektiviteten af den enkelte scanning.

Der skal udvikles en bedre midling af de kraftmålinger, der modtages fra UR10 end den midling, der anvendes i den nuværende løsning. Her kan der eks. anvendes *weighted moving average* således, at målinger i det ustabile område får en mindre vægtning i den resulterende force.

Yderligere kan der overvejes at benytte et ekstra modul, der kan monteres på UR10's hoved i form af en *Robotiq force torque sensor*[12]. På den måde kommer man udenom UR10's egen registrering af force og kan opnå mere præcise målinger.

I forlængelse af den manglende fornemmelse af den reelle force åbnede Lars Bolvig for ideen med at implementere et force-barometer i brugergrænsefladen, som skal bidrage til operatørens oplevelse af den force feedback, der allerede registreres i joysticket. Barometeret skal angive den kraft, eks. mellem 0-10 kg, der trykkes med på patienten og vha. farveskala indikere graden af denne kraft.

Lars Bolvig informerede om, at lægerne allerede benytter sig af barometre under elastografi, når der i kroppens væv skal skelnes mellem maligne tumorer og normalt kropsvæv. Derfor vil det være et naturligt værktøj for speciallægerne at bruge denne form for indikation.

Systemet har potentiale til at række over internationale, nationale og regionale grænser således, at en lokal placeret specialist kan række ud med sin ekspertise i et større område. Dette kræver dog, at systemet videreudvikles til at kunne kommunikere på tværs af netværk.

Når systemet skal tages i brug, kan det være afgørende, at tilgangen til systemet fra operatøren bliver så intuitiv som muligt. Derfor anbefales det at der designes og udvikles et nyt håndtag til GMT således, at betjeningen af GMT kan ske på samme måde, som operatøren er vant til.

Referenceliste

- [1] Failure mode and effect analysis. <http://www.weibull.com/basics/fmea.htm>. [Online; accessed 13-10-2015].
- [2] Fishbone (ishikawa) diagram. <http://asq.org/learn-about-quality/cause-analysis-tools/overview/fishbone.html>. [Online; accessed 22-09-2015].
- [3] Modbus server - 16377. <http://www.universal-robots.com/how-tos-and-faqs/how-to/ur-how-tos/modbus-server-16377/>. [Online; accessed 09-10-2015].
- [4] Open haptic toolkit developer edition. <http://www.geomagic.com/en/products/open-haptics/overview>. [Online; accessed 06-09-2015].
- [5] Ozeki camera sdk, version: 1.2.0. http://www.camera-sdk.com/p_13-download-onvif-standard-ozeki-camera-sdk-for-webcam-and-ip-camera-developments-onvif.html. [Online; accessed 29-10-2015].
- [6] Podio. <https://podio.com/site/da/use-cases/project-management>. [Online; accessed 19-08-2015].
- [7] Rational unified process, best practices for software development teams. https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf. [Online; accessed 25-08-2015].
- [8] Real time streaming protocol (rtsp). <https://www.ietf.org/rfc/rfc2326.txt>. [Online; accessed 03-11-2015].
- [9] Universal robot, remote control via tcp/ip. <http://www.universal-robots.com/how-tos-and-faqs/how-to/ur-how-tos/remote-control-via-tcpip-16496/>. [Online; accessed 09-09-2015].
- [10] Vlc library til .net, version: 3.10.2013. <http://www.codeproject.com/Articles/109639/nVLC>. [Online; accessed 31-10-2015].
- [11] Windows desktop sharing. [https://msdn.microsoft.com/en-us/library/windows/desktop/bb968809\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/bb968809(v=vs.85).aspx). [Online; accessed 23-10-2015].
- [12] Zacobria. <http://www.zacobria.com/robotiq-zacobria-universal-robot-force-torque-sensor.html>. [Online; accessed 13-09-2015].
- [13] Alessandro Di Cerbo et al. Narrative review of telemedicine consultation in medical practice. <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4298290/>. [Online; accessed 08-12-2015].
- [14] Iris Hossmann et al. Europe's demographic future. http://www.berlin-institut.org/fileadmin/user_upload/Studien/Europa_e_Kurzfassung_sicher_o_B.pdf. [Online; accessed 08-12-2015].

- [15] L. Pian et al. Potential use of remote telemonitoring as a transformational technology in underresourced and/or remote settings. <http://www.hindawi.com/journals/emi/2013/986160/>. [Online; accessed 08-12-2015].
- [16] Danmarks Statistik. Øgede kommunale udgifter til sundhedsfremme. <http://www.dst.dk/pukora/epub/Nyt/2013/NR283.pdf>. [Online; accessed 08-12-2015].
- [17] Sundhedsregisteret. http://www.esundhed.dk/sundhedsregistre/LPR/Sider/LPR04_Tabel.aspx. [Online; accessed 08-12-2015].
- [18] Sundhedsstyrelse. Brug af telemedicin mellem rigshospitalet og bornholms hospital. <http://www.kora.dk/media/529713/dsi-2242-sammenfatning.pdf>. [Online; accessed 08-12-2015].
- [19] Kronisk sygdom. <http://sundhedsstyrelsen.dk/da/sundhed/folkesygdomme/kronisk-sygdom>. [Online; accessed 07-12-2015].

- 1 - 15138 Telemedicinsk robotstyret ultralydsskanning
- 2 - Feasibilitetsstudie
- 3 - OpenHaptics_ProgGuide
- 4 - OpenHaptics_RefGuide
- 5 - Medical Device Direktiv 93-42-EØF - 2007
- 6 - client_interface
- 7 - Interview Sonograf
- 8 - ModBus Register Overview
- 9 - ModBus server data
- 10 - Modbus_Application_Protocol_V1_1b3
- 11 - scriptmanual_en
- 12 - software_manual_en_polyscope
- 13 - UR10_User_Manual_da_Global
- 14 - Videodemonstration
- 15 - Komplet klassediagram
- 16 - Tidsplaner
- 17 - Milestones template