
Titel:

Systemarkitektur for Sorting Industrial Robot

Versionshistorik

Ver.	Dato	Initialer	Beskrivelse
0.1	02-03-12	RHT	Første udkast lavet fra sidste projekt.

Indholdsfortegnelse

Introduktion.....	5
Formål.....	5
Referencer.....	5
Definitioner.....	5
Dokumentstruktur og læsevejledning.....	5
Dokumentets rolle i en iterativ udvikling.....	6
System oversigt.....	7
System kontekst.....	7
System introduktion.....	7
Systemets grænseflader.....	7
Grænseflader til person aktører.....	7
Fører.....	7
Tekniker.....	7
Grænseflader til eksterne system aktører.....	7
Grænseflader til hardware aktører.....	7
Grænseflader til software aktører.....	7
Use case view.....	7
Oversigt.....	7
Logisk view.....	7
Oversigt.....	7
Use case realiseringer.....	8
Use-case X: XXX.....	8
Beskrivelse.....	8
Proces view.....	8
Oversigt over processer.....	8
Implementering.....	8
Kommunikation og synkronisering.....	8
Procesbeskrivelser.....	8
Deployment view.....	8
Oversigt over systemkonfigureringer.....	8
Systemkonfigureringer.....	8
Node beskrivelser.....	8
Development view.....	9
Oversigt.....	9
Komponentbeskrivelser.....	9
Generelle designbeslutninger.....	9
Arkitektur mål og begrænsninger.....	9
Arkitektur mønstre.....	9
Generelle brugergrænsefladeregler.....	9
Fejlhåndtering.....	9
Implementeringssprog og værktøjer.....	9
Implementeringsbiblioteker.....	9
Størrelse og ydelse.....	9
Kvalitet.....	9
Oversættelse	10
Oversættelse-hardware.....	10
Oversættelse-software.....	10
Oversættelse og linkning.....	10
Installation.....	10

Kørsel.....	10
Kørsels-hardware.....	10
Kørsels-software.....	10
Start og stop.....	10
Informationsdisplay.....	10
Bilag.....	10

Introduktion

Formål

Formålet med dette systemarkitektur-dokument er at dokumentere designet af SIR.

De væsentlige aspekter af designet er specificeret heri, og man kan ved at læse dette dokument opnå et overblik over designet.

Referencer

[1] Kravspecifikation

[2] Doxygen generet kode dokumentation.

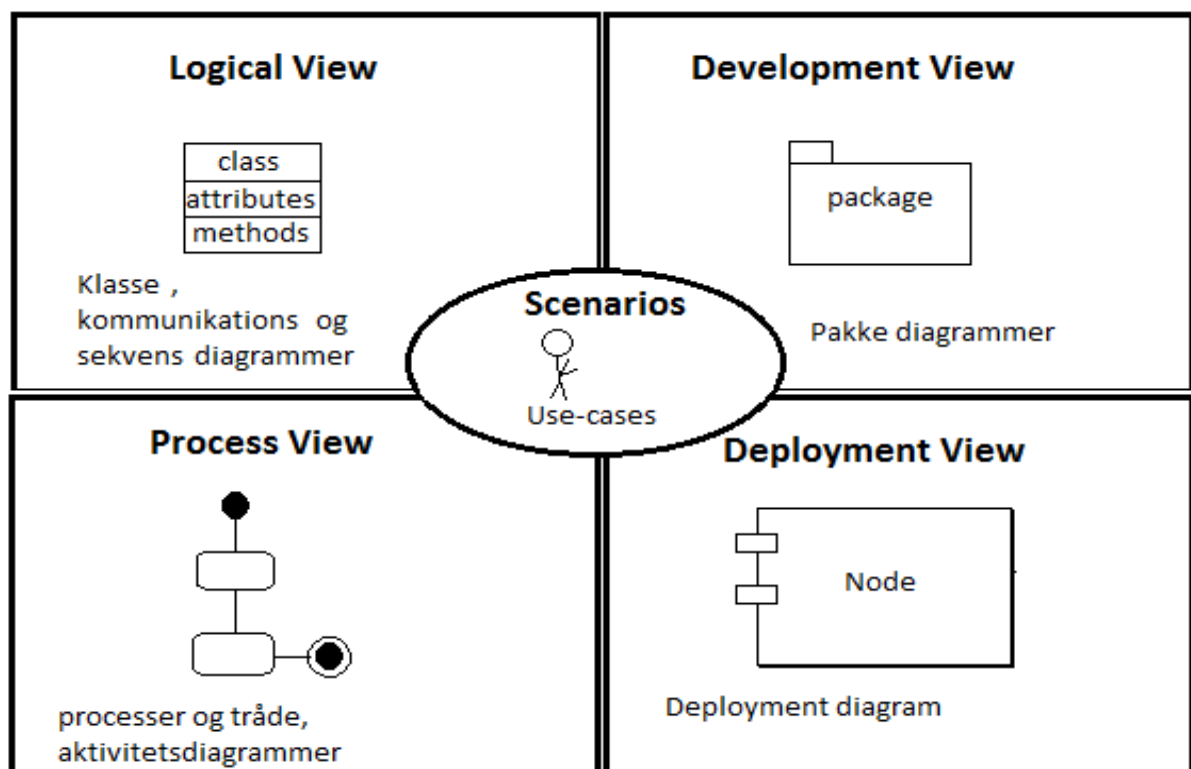
Definitioner

Tekst.

Dokumentstruktur og læsevejledning

Vi har taget udgangspunkt i 4+1 modellen, som illustrerer forskellige måder at vise softwarearkitekturen på.

Illustration 1: 4+1 modellen



Dokumentets rolle i en iterativ udvikling

Selve dokumentet består af dokumentationen fra de fire iterationer.

I iterationerne er der arbejdet med:

- Udarbejdelse af use-cases i kravspecifikation.
- Design af use-cases.
 1. Sekvensdiagrammer.
 2. Klassebeskrivelser (Doxygen).
 3. Klassediagrammer.
- Implementering af use-cases.
- Unit test og integreringstest.
- Accepttest udarbejdet ud fra kravspecifikationen.
- Udarbejdelse af projektrapport.

System oversigt

System kontekst

Tekst.

System introduktion

Tekst.

Systemets grænseflader

Grænseflader til person aktører

Tekst.

Fører

Tekst.

Tekniker

Tekst.

Grænseflader til eksterne system aktører

Tekst.

Grænseflader til hardware aktører

Tekst.

Grænseflader til software aktører

Tekst.

Use case view

Oversigt

Se use cases under Kravspecifikationen.

Logisk view

Oversigt

Tekst.

Drivere:

Ingeniørhøjskolen i Århus

Tekst.

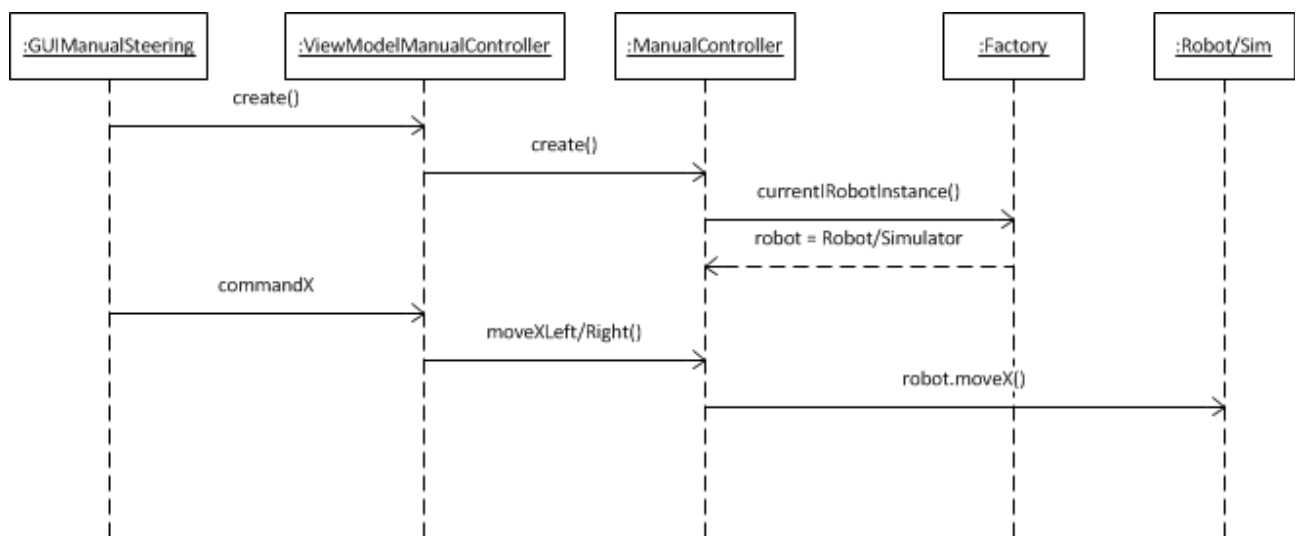
Use case realiseringer

Tekst.

Use-case X: XXX

Noget der forklarer hvad der sker i use casen for eksempel med hjælp af et sekvensdiagram.

Use-case 4: Manuelt Styre



Figur X

Beskrivelse:

Denne er lavet i tre dele:

- View(GUIManualSteering)
- ViewModel(ViewModelManualSteering)
- Model(ManualController med forbindelse videre til IRobot)

I View-delen er der designet, så man har adgang til knapper for de forskellige funktionaliteter for at bevæge robotten. Det er enten ved at dreje en akse til den ene eller anden side eller ved at ændre på en af robotternes koordinater. Den kan også åbne og lukke for kloen.

GUIManualSteering er forbundet med ViewModelManualSteering ved hjælp af normal databinding for hastigheden af bevægelserne, og bevægelsesfunktionerne er blevet implementeret ved hjælp af Commands, der er i ViewModelManualSteering. Der er på denne måde undgået code-behind i View-delen af designet.

ViewModelManualSteering har så simple funktioner som muligt, så der kunne forbindes direkte fra View til ViewModel uden ekstra argumenter. Dette betyder, at der for eksempel er en funktion til at bevæge basen mod højre og en for at bevæge den til venstre.

ManualController har så mere generelle funktioner for bevægelse af robotten, som er forbundet videre til Factory's "currentRobotInstance". Dette betyder, at manuel styring kan ved kodeeksekvering skifte mellem at blive kørt på Robotten og Simulatoren ude fra styresystemet.

Beskrivelse

Tekst.

Proces view

Oversigt over processer

Tekst.

Implementering

Tekst.

Kommunikation og synkronisering

Tekst.

Procesbeskrivelser

Tekst.

Deployment view

Oversigt over systemkonfigureringer

Tekst.

Systemkonfigureringer

Tekst.

Node beskrivelser

Tekst.

Development view

Oversigt

Tekst.

Komponentbeskrivelser

Se dokumentation på CD'en for kode dokumentation("kode autogen dokumentation.pdf").

Generelle designbeslutninger

Dette afsnit beskriver de beslutninger vi har taget om arkitekturdesign.

Arkitektur mål og begrænsninger

Tekst.

Arkitektur mønstre

Vi har valgt at bruge MVVM pattern til at implementere programmet.

Dette er blevet valgt da C# med WPF giver relativ nem mulighed for at implementere med MVVM pattern. Ting som DataBinding og Commands.

Andre patterns der er blevet brugt er Singleton og Factory sammen med Indirection for at fjerne afhængighed fra forskellige komponenter i systemet samt at flere klasser havde brug for at deles om de samme klasse instanser.

Generelle brugergrænsefladeregler

Tekst.

Fejlhåndtering

Tekst.

Implementeringssprog og værktøjer

Tekst.

Implementeringsbiblioteker

Tekst.

Størrelse og ydelse

Tekst.

Kvalitet

Tekst.

Oversættelse

Oversættelse-hardware

Tekst.

Oversættelse-software

Tekst.

Oversættelse og linkning

Tekst.

Installation

Tekst.

Kørsel

Kørsels-hardware

Tekst.

Kørsels-software

Tekst.

Start og stop

Tekst.

Informationsdisplay

Bilag

Tjek medhørende CD for at se bilagene.