

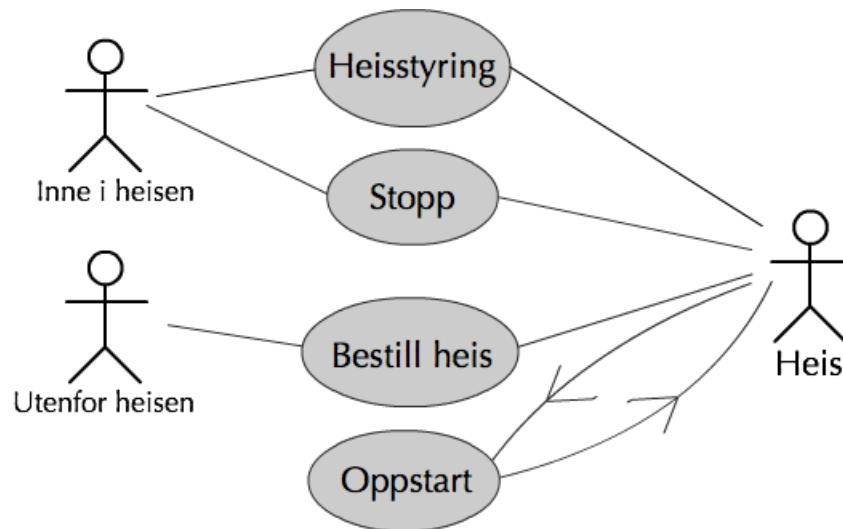


INSTITUTT FOR TEKNISK KYBERNETIKK

HEISPROSJEKT — TTK4125

Systemdokumentasjon

Marius Thoresen
Erik Wilthil



Figur 1: Usecase-diagram

1 Usecases

Bestillingsknapp: Knappene opp og ned som befinner seg i hver etasje.

Etasjeknapp: Knappene inne i heisen som bestemmer hvor heisen skal gå

Oppstart

Precondition: Ingen

Trigger: Program startes

Suksessscenario:

1. Sjekk etasjesensor
2. Kjør heis opp til den kommer til en etasje
3. Stopp heisen

Utvidelser:

- 1a. Hvis heis er i etasje, hopp til punkt 3.

Suksessgaranti: Heis i ro i kjent etasje

Minimal garanti: Samme som suksessgaranti

Stopp

Precondition: Heis ikke i oppstartsmodus

Trigger: Stopp-knapp trykkes eller obstruksjon aktiveres når heisen er i bevegelse

Suksessscenario:

1. Heisen stoppes
2. Alle bestillinger slettes fra bestillingsliste
3. For å komme ut av stopp-modus må bestilling legges inn av etasjeknapp og hvis det er obstruksjon må denne fjernes.

Suksessgaranti: Heisen står stille og bestillingene slettes

Minimal garanti: Ingen

Bestill Heis

Precondition: Heis ikke i stoppmodus eller oppstartsmodus

Trigger: Bestillingsknapp trykkes inn

Suksessscenario:

1. Bestilling i etasje med retning blir lagt til i ordreliste
2. Heisen fortsetter med andre ordre. En bestilling vil slettes etter at den blir ekspedert
3. Hvis heisen ikke har flere bestillinger videre i retningen den sist kjørte, vil den snu og ekspedere ordre i motsatt retning.
4. Når heisen kommer til riktig etasje, stopp heisen
5. Fjern bestilling fra bestillingslisten, åpne døren
6. Start nedtelling 3 sekunder
7. Når nedtelling når 0, lukk døren

Utvidelser:

- 2a. Hvis heisen kommer til bestilt etasje i samme retning som bestillingen, hopp til punkt 4.
- 6a. Hvis obstruksjon er aktivert, start nedtelling på nytt.

Suksessgaranti: Heis kommer til bestilt etasje. Døren åpnes og holdes åpen i minimum 3 sekunder, den vil holdes åpen så lenge det er obstruksjon.

Minimal garanti: Ingen

Bestilling fra etasjeknapp

Precondition: Heis ikke i oppstartsmodus

Trigger: Etasjeknapp trykkes

Suksessscenario:

1. Bestilling i etasje blir lagt til i bestillingsliste
2. Heisen fortsetter med andre bestillinger. En bestilling vil slettes etter at den blir ekspedert.
3. Hvis heisen ikke har flere bestillinger videre i retningen den sist kjørte, vil den snu og ekspedere bestillinger i motsatt retning.
4. Når heisen kommer til riktig etasje, stopp heisen
5. Fjern etasjen fra bestillingslisten, åpne døren
6. Start nedtelling 3 sekunder
7. Når nedtelling når 0, lukk døren

Utvidelser:

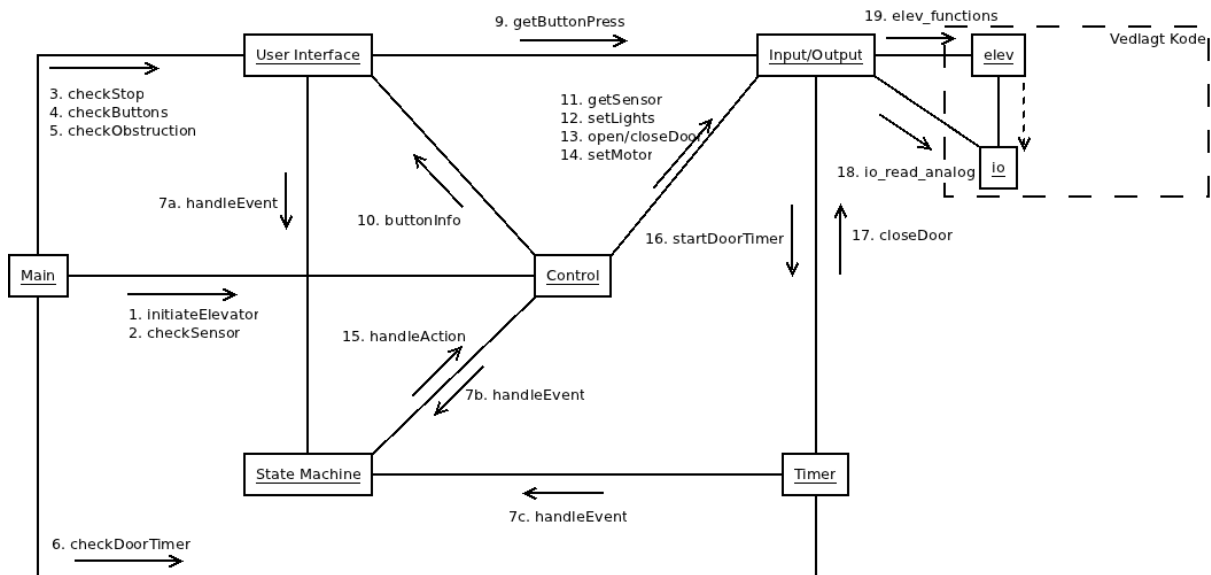
- 2a. Hvis heisen kommer til bestilt etasje, hopp til punkt 4.
- 6a. Hvis obstruksjon er aktivert, start nedtelling på nytt.

Suksessgaranti: Heis kommer til bestilt etasje. Døren åpnes og holdes åpen i minimum 3 sekunder, den vil holdes åpen så lenge det er obstruksjon.

Minimal garanti: Samme som suksessgaranti

2 Systemarkitektur

1. initiateElevator kalles hver gang heisprogrammet starter, og utfører oppstartsprosedyren beskrevet i Usecase-diagrammet.
2. - 6. check-funksjonene kalles kontinuerlig i main, og registrerer hendelser for tilstandsmaskinen.



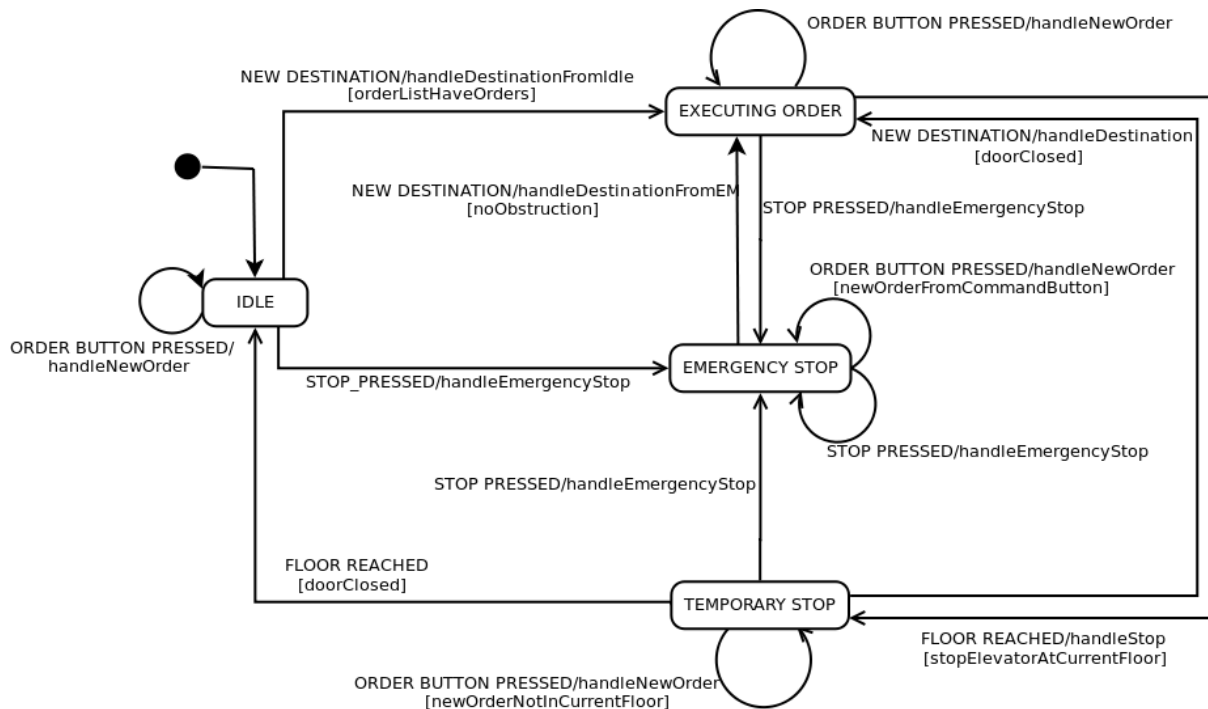
Figur 2: Overordnet systemarkitektur

7. `handleEvent` kalles av `check`-funksjonene i tre forskjellige klasser. `handleEvent` kalles med et av argumentene `NEW_DESTINATION`, `FLOOR_REACHED`, `STOP_PRESSED` og `ORDER_BUTTON_PRESSED` avhengig av hvilke hendelser `check`-funksjonene registrerer.
8. `handleAction` er et samlebegrep for `handle`-funksjonene som kalles av tilstandsmaskinen.
9. betingelser er funksjoner som returnerer 0 eller 1. Disse må være oppfylt (returnerer 1) for å få en transisjon i tilstandsmaskinen der disse er satt som betingelse.
10. `getUiSignals` er et samlebegrep for funksjonene som henter signaler fra knappene og obstruksjon. Funksjonene kalles av `check`-funksjonene.
11. Siste bestillingsknapp som trykkes, lagres i `ui`-klassen, og `getLastOrder` returnerer denne til kontroll-klassen.
12. `getSensor` er et samlebegrep for funksjoner som henter informasjon fra etasjesensorene.
13. `controlLights` er et samlebegrep for funksjoner som skrur av og på lys.
14. `openDoor` og `closeDoor` håndterer døra.
15. `controlMotor` er et samlebegrep for funksjoner som starter og stopper motoren.
16. `startDoorTimer` kalles når døra åpnes for å telle ned tre sekunder.
17. `closeDoor` kalles fra timer-klassen og lukker døra når timeren har telt ned.
18. `io.read_analog` er en funksjon som returnerer målt motorhastighet.
19. `elevFunctions` er funksjonene i det ferdige grensesnittet mot heisen.

3 Tilstandsdiagram

4 Klassediagrammer

Heisstyringa er delt opp i klasser. Måten vi har implementert dette på er at alle variabler er deklartert static for å holde de internt i en klasse/fil, og innføre set- og getfunksjoner der det trengs. Grunnen til



Figur 3: Tilstandsdiagram
Tilstandsmaskinen er implementert som en tilstandstabell i klassen statemachine.

denne oppdelingen er selvfølgelig å få en logisk oppdeling, hvor hver klasse har et veldefinert ansvarsområde. For hver klasse bør det gå greit fram hva de forskjellige variablene og funksjonene gjør, men det er gitt utdypende forklaring ved de mest sentrale.

4.1 Tilstandsmaskinen

Tilstandsmaskinen holder orden på hvilken tilstand heisen står i.

handleEvent

Bestemmer, sammen med tilstandsvariablen, nestetilstand og handling for heisen når den genererer en hendelse.

4.2 Kontroll

Kontrollklassen inneholder alle funksjoner og variabler som håndterer logikken og den fysiske funksjonen i heisen, som å bestemme retning, at motoren skal startes, hvilke lys som skal tennes, holde orden på bestillingene, og lignende.

Handlinger

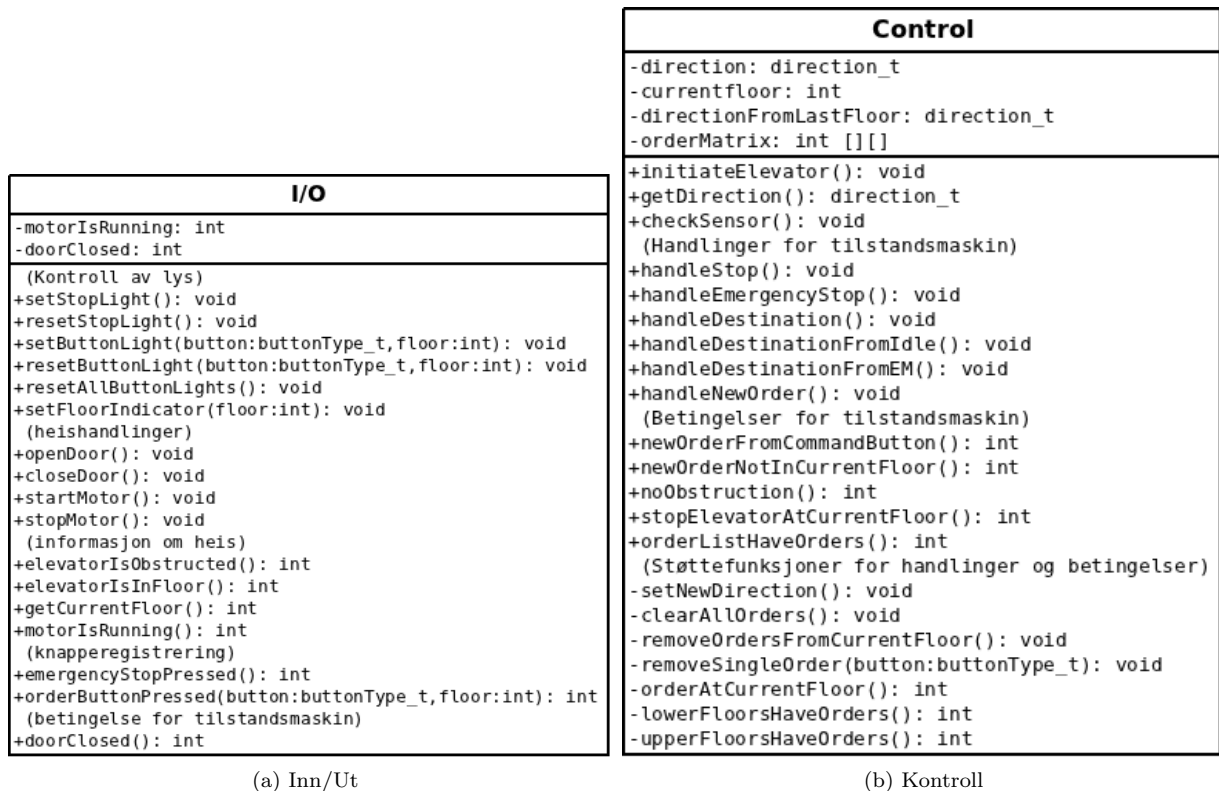
De forskjellige handlingene (`handle...`) kalles av tilstandsmaskinen, og beskriver den overordnede oppførselen til heisen. Grunnen til tre forskjellige `handleDestination`-funksjoner er at man i nødstop- og idletilstand trenger litt annerledes rutiner for å bestemme neste mål for heisen.

Sperre? Vakt?

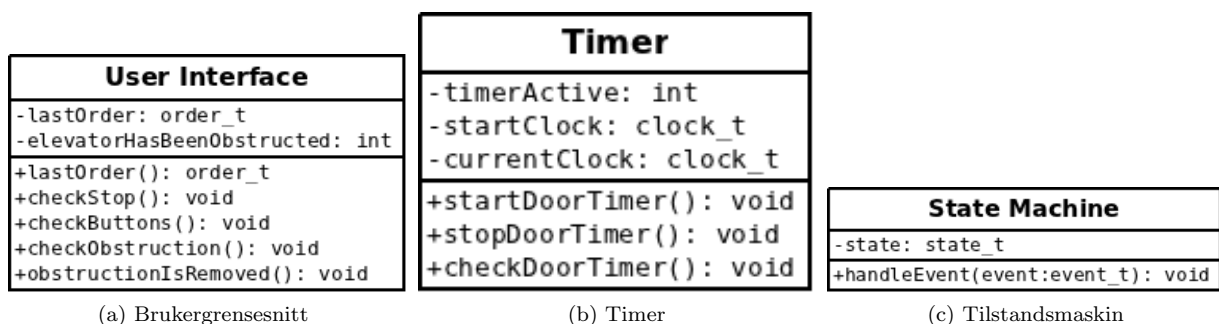
`newOrderFromCommandButton`, `newOrderNotInCurrentFloor`, `noObstruction` og `stopElevatorAtCurrentFloor` er ?sperrer? tilstandsmaskinen bruker for å avgjøre om den skal utføre handlingene.

Checksensor

En lyttefunksjon som står og går i main-funksjonen for å avgjøre hvilken etasje heisen er i, og gi en handling når den ankommer en etasje.



Figur 4: Klassesdiagrammer



Figur 5: Klassesdiagrammer

Heisstyring

setNewDirection, clearAllOrders, setLightsAtElevatorStop, removeOrdersFromCurrentFloor og removeSingleOrder er funksjoner som påvirker retning, ordre og lys for heisen.

Sammeligning

De fire nederste funksjonene er hjelpefunksjoner som bestemmes av ordrene heisen har. Brukes for å bedre lesbarhet og for å unngå for mye nøsting.

4.3 User Interface

Denne klassen sørger for å håndtere input fra brukeren, som bestillingsknapper, stoppknapp og obstruksjonsføleren i døra. Den sørger for å sende bestillinger til kontroll.

Sjekkfunksjoner

checkButtons, checkStop og checkObstruction løper i main-funksjonen for å for å håndtere input fra brukeren. De genererer også hendelser for tilstandsmaskinen.

LastOrder

Denne kalles av kontroll-klassen for å returnere siste ordreknap som ble trykket inn og legge bestilling i ordrelisten.

4.4 Timer

Timeren er en enkel klasse for å telle ned tre sekunder hver gang døren åpnes i en etasje. Den har to variabler fra time-biblioteket for å telle opp til 3 sekunder, og et heltall som holder orden på om den er aktiv eller ikke.

checkDoorTimer

Denne står også og løper i main-funksjonen, og sjekker om nedtellingen er ferdig eller ikke.

4.5 I/O

I/O klassen er en egenlagt driver for å styre heisen. Dette er vår mest maskinnære klasse, som inneholder funksjoner for alle hardware-relaterte operasjoner som å skru av og på lys, gi signal fra sensoren og sette motorhastigheten. Disse funksjonene implementeres ved hjelp av medfølgende funksjoner i elev.c og io.c

Lysfunksjoner

Det er funksjoner for å skru lys av og på. Bestillingslampene kan skrues av en og en, eller alle på en gang som for en nødstop. Etsjelyset har ingen resetfunksjon, da kun et skal tennes av gangen, og derfor kan det ordnes i setfunksjonen.

Heisaksjoner

Funksjoner for dør- og motorstyring. Det er egen motorstop for nødsituasjoner, **hvorfor?**

Informasjon om Heis

Funksjoner som registrerer fysisk informasjon om heisen

Knapperegistreringer orderButtonPressed blir kalt fra kontroll hver gang man trykker en knapp, og legger etasje og knappetype inn i diagrammet.

vakt? sikring?

doorClosed er vakt? sikring? som brukes av tilstandsmaskinen.