

Hejsan Henrik!

Det kan mycket väl vara något.. men eftersom jag inte har någon aning om vad en simple cruise controller simulering skulle innebära för något så behöver jag din ide lite mer förklarad... min gissning är att man har en satt hastighet som man ökar och minskar med två knappar (+/-) och ett läge på och av.... och det skulle ju vara lite för enkelt.. men om du förklarar det hela bättre för mig.. med t.ex. hur ett körning eller gui skulle kunna fungera så kanske det går bra...

Min bild av du vill göra ser ut så här:

**Programmed speed: 70 km/h**

**Status: on**

**1. increase speed**

**2. decrease speed**

**3. Toggle on off**

**0. exit**

>1

**Programmed speed: 75 km/h**

**Status: on**

**1. increase speed**

**2. decrease speed**

**3. Toggle on off**

**0. exit**

>2

**Programmed speed: 70 km/h**

**Status: on**

**1. increase speed**

**2. decrease speed**

**3. Toggle on off**

**0. exit**

>3

**Programmed speed: 70 km/h**

**Status: off**

**1. increase speed**

**2. decrease speed**

**3. Toggle on off**

**0. exit**

>0

och då skulle det vara lite väl enkelt.. en variabel speed som kan ökas och minskas... och en på eller av variabel...

Så förklara gärna närmare.

/Micke

Respons Henrik Axelsson

Hej Mikael!

Jag tror jag förstår hur du tänker och mitt svar blir i form av en bild och diskussion kring denna. Misstänker att jag var alltför kortfattad i min beskrivning av projektet.

Börjar med bilderna, vilken beskriver de tillståndsmaskiner som jag tänker mig implementera (i form av switch case satser) som beskriver beteendet hos systemet. Därefter kommer en bild som illustrerar den logiska strukturen med olika delar (implementerar jag i separata c- och h-filer samt lämpliga includes).

GUIT är uppbyggt av vårt kära menysystem, där man kan lägga till händelser till en kö (vilket bygger på alla inlämningsuppgifterna). Därefter väljer man att starta exekveringen av simuleringen. När en simuleringsrunda är klar (kön är tom), så återkommer man till kösystemet och kan starta en ny vända om så önskas. Annars avsluta via menyval.

När simuleringen körs, så kommer lämpliga utskrifter göras i kommandofönstret så att man kan följa exekveringen av maskinen beroende på vad som injiceras.

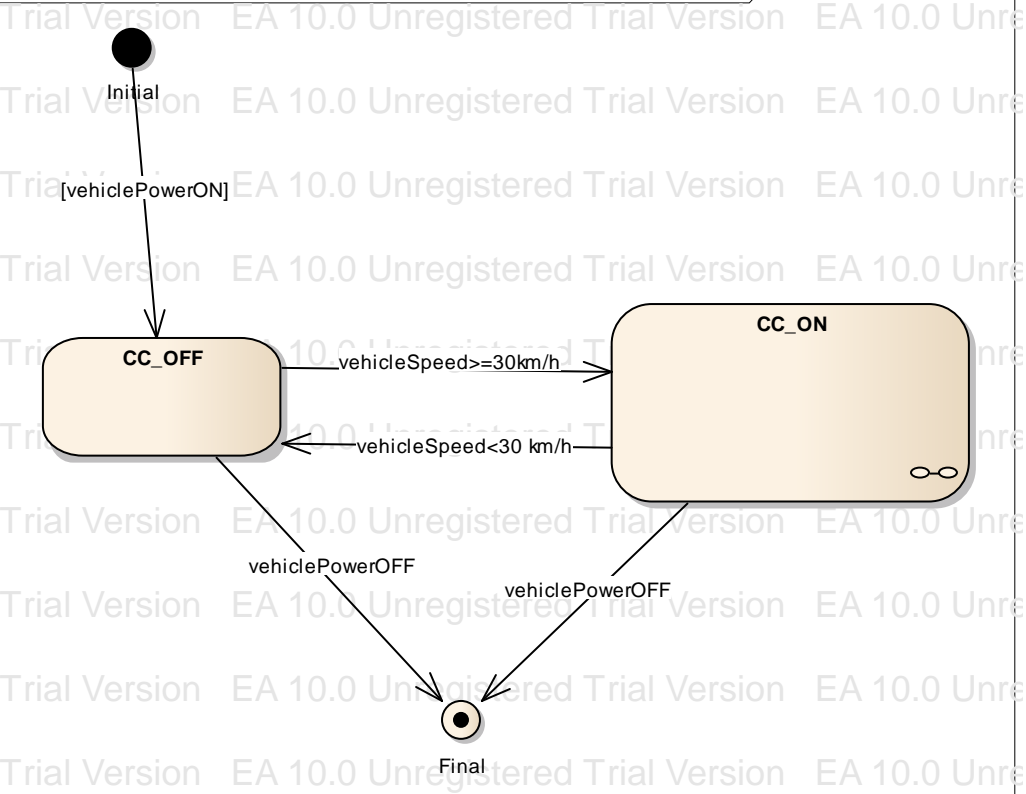
Systemet och implementationen är skalbar beroende på hur komplext och verklighetstroget man vill göra systemet.

Återkom gärna med feedback på detta!

Mvh Henrik

Yttre maskin

stm [Package] Use Case State Diagrams [Use Case Main State Machine]



## Inre maskin för CC\_ON

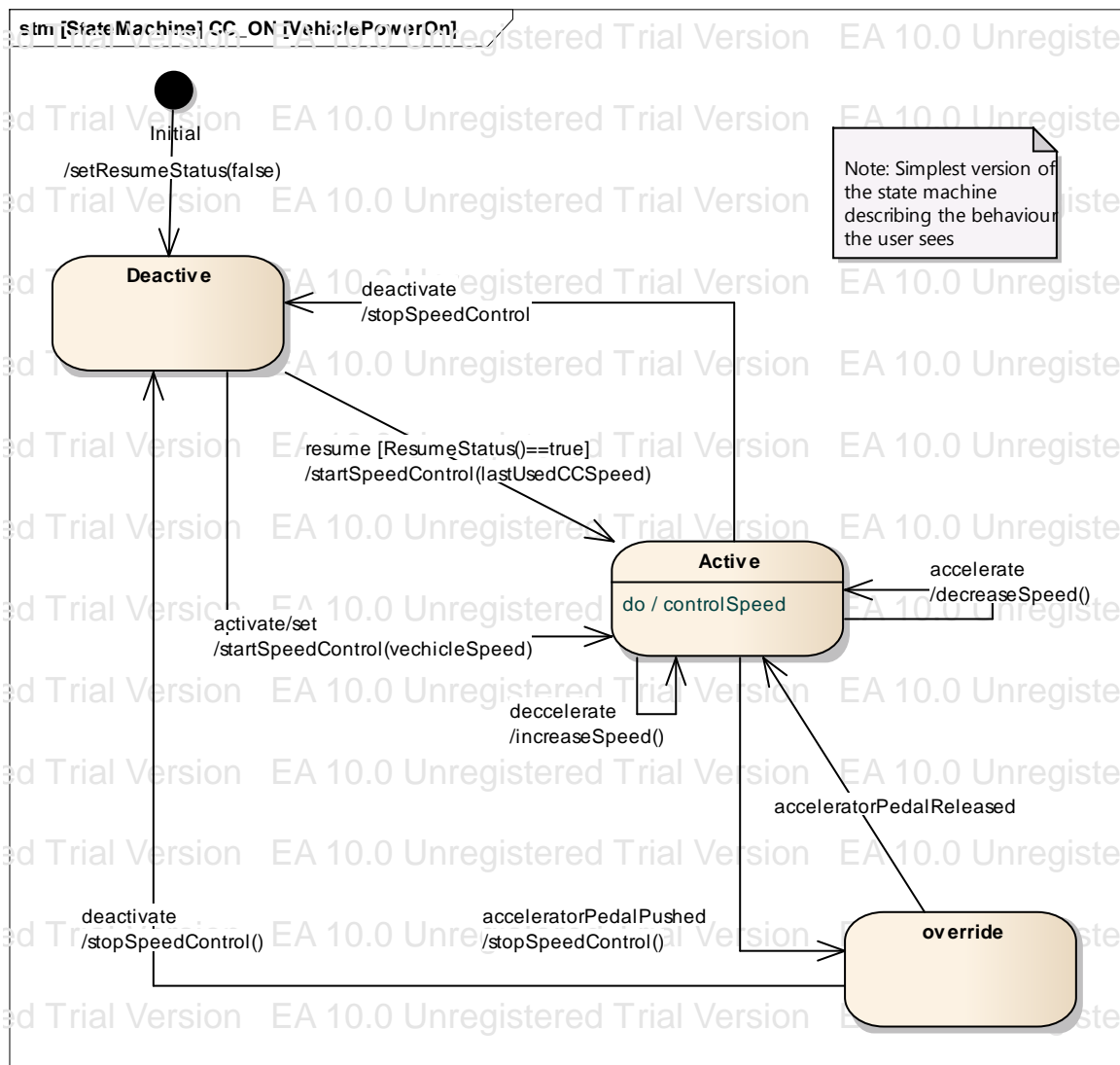


Diagram som visar den statiska strukturen för det hela (OBS! Detaljerna är inte färdiga, detta är konceptet):

