



Giotto - A Time-Triggered Language for Embedded Programming

Freie Universität Berlin

Institut Informatik

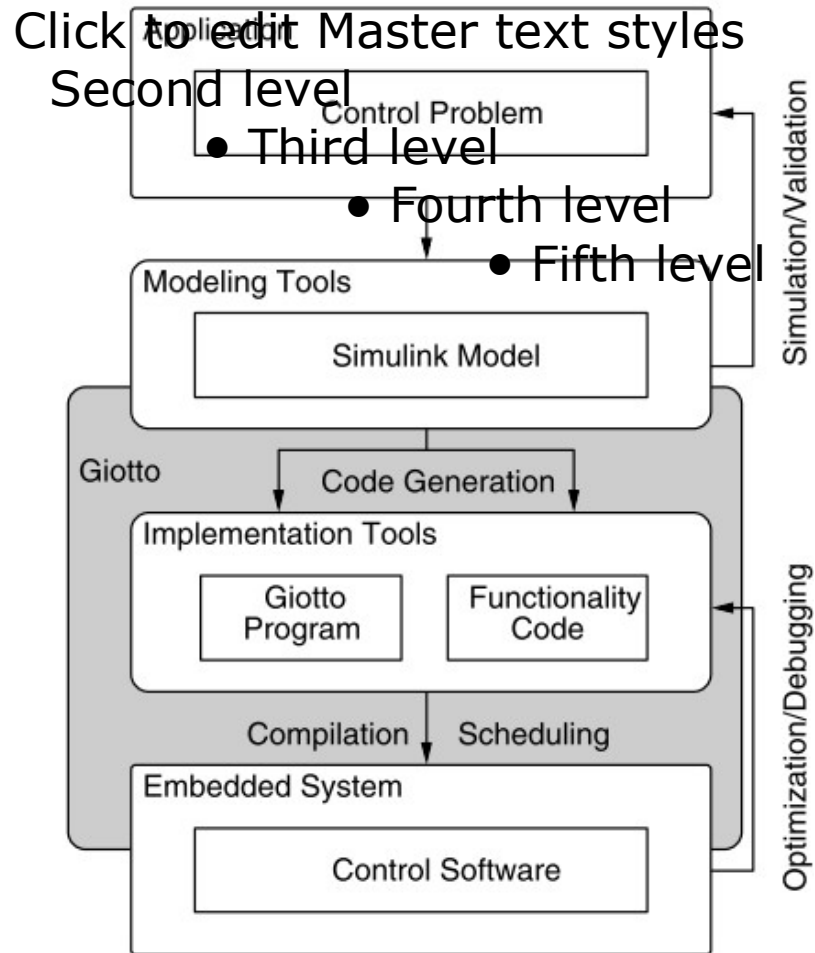
Wintersemester 2010/2011

Seminar Synchrone Programmiersprachen

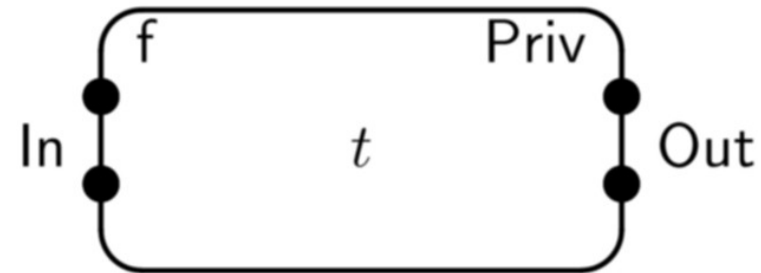
Robert Fehrmann

Datum: 25.02.2013

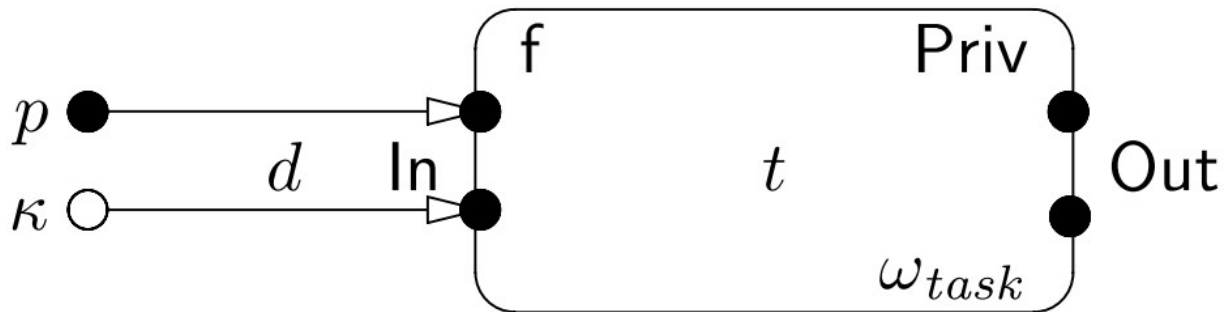
Vorgehen und Werkzeugeinsatz beim Giotto-based Entwicklungsansatz



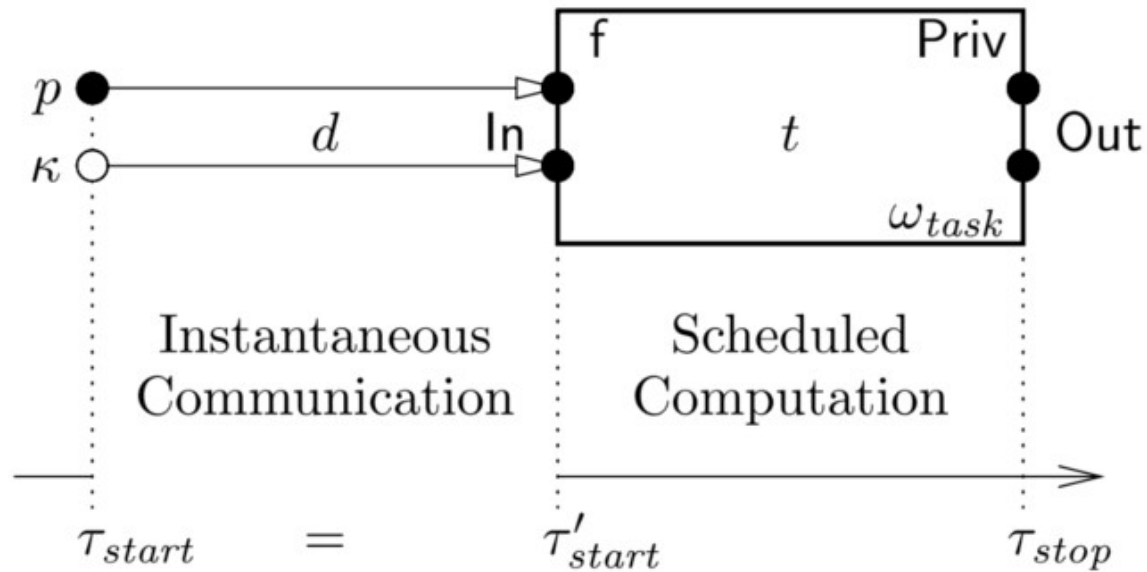
Informelle Beschreibung - *ports* & *tasks*

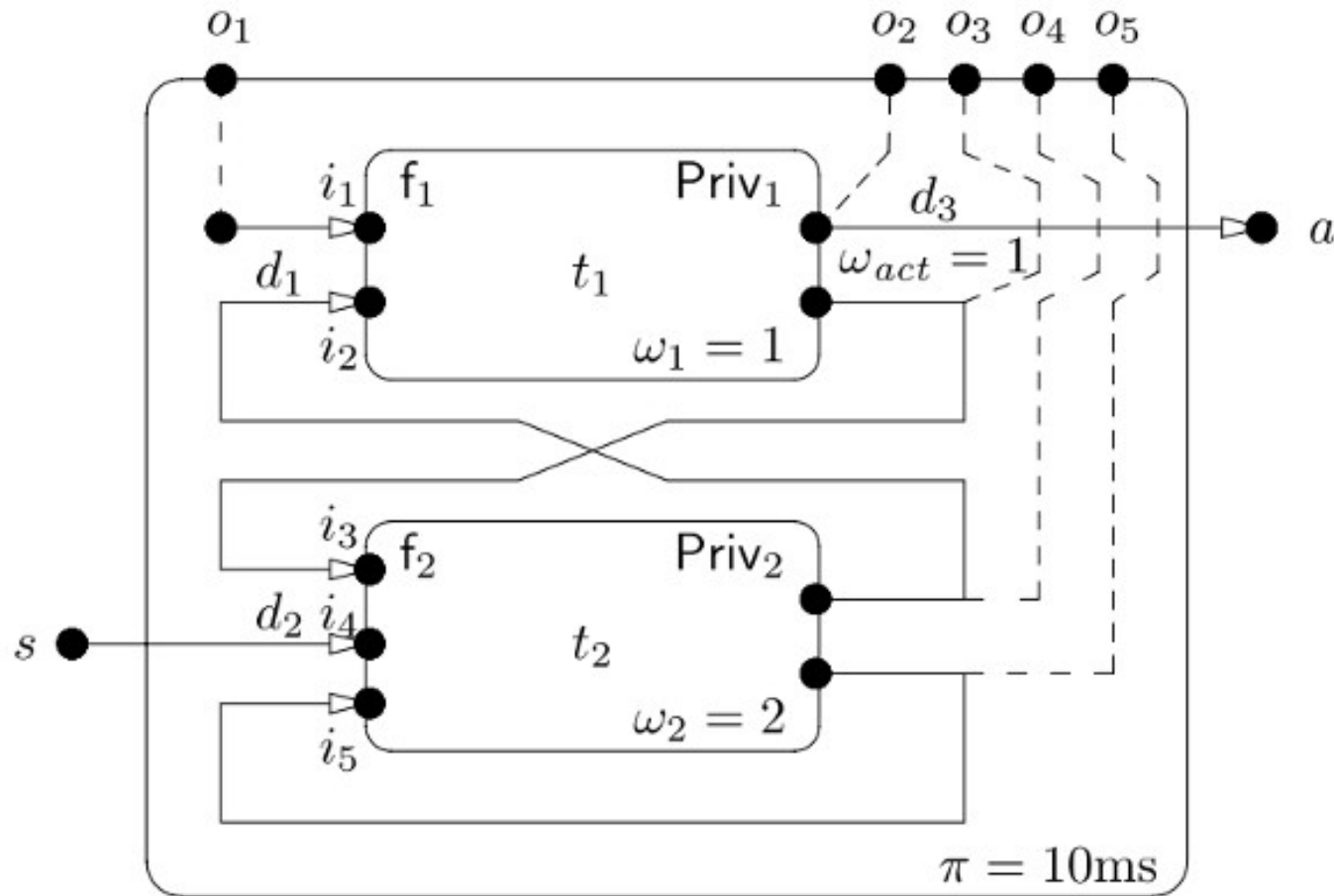


Informelle Beschreibung - task invocation: driver & frequency

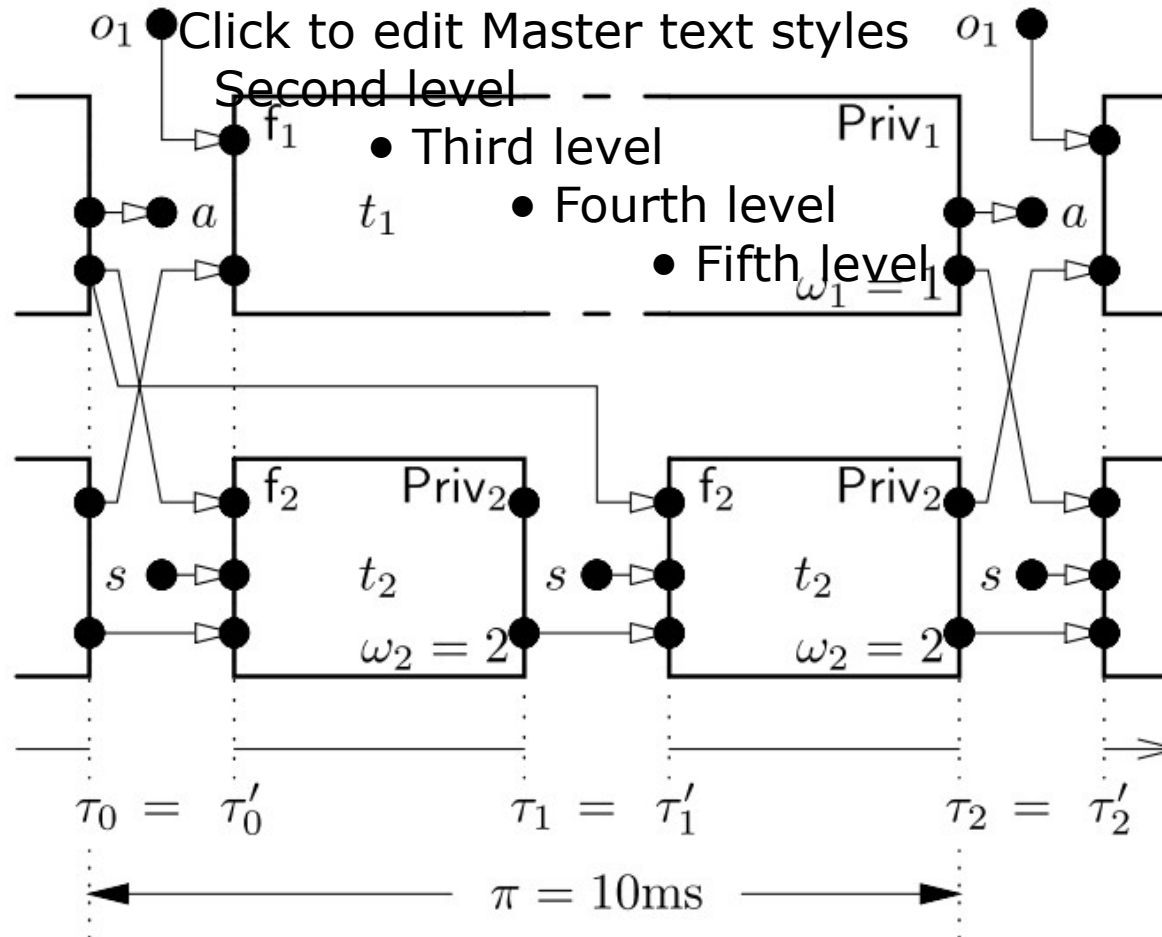


invocation:
communication phase % schedule
phase

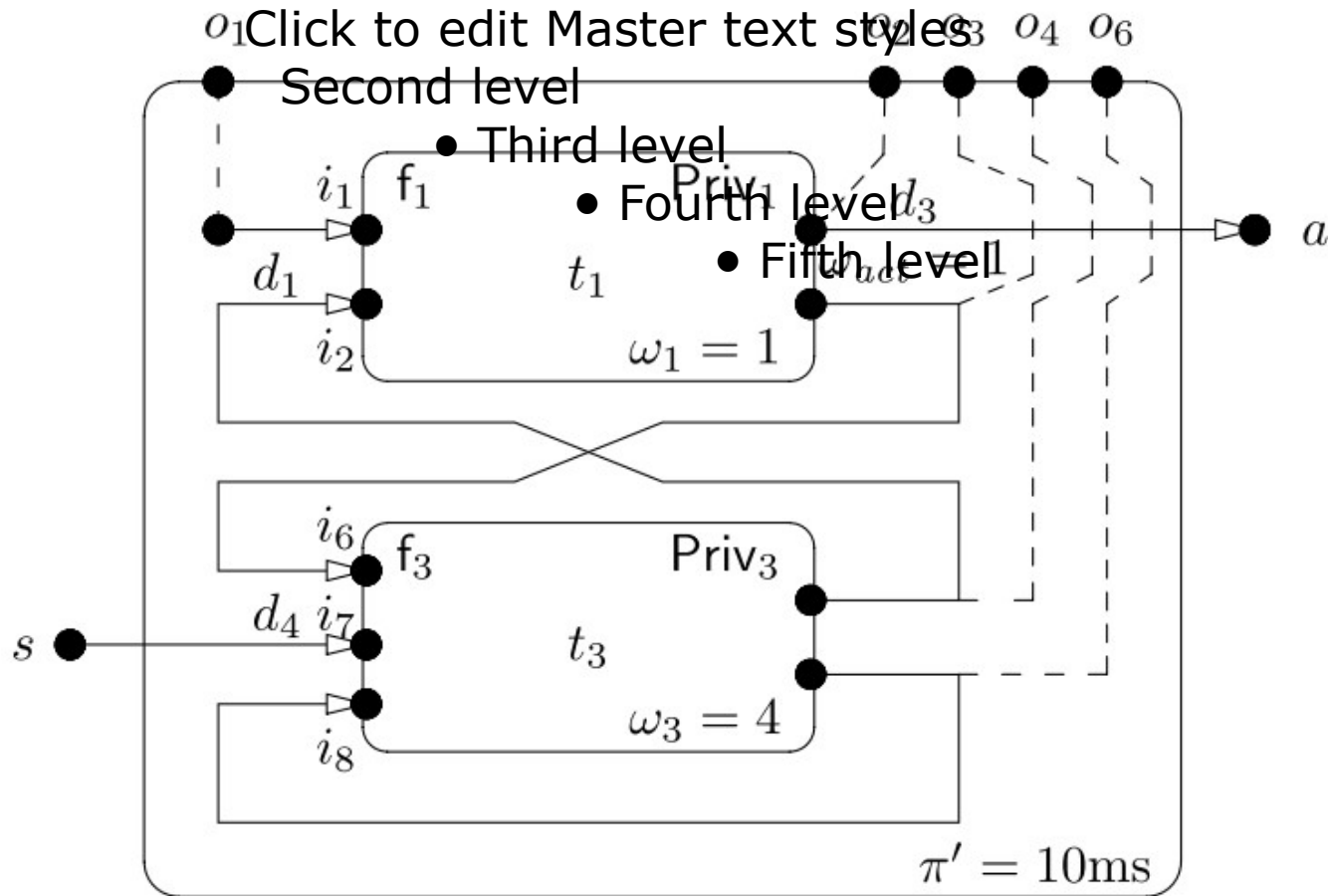




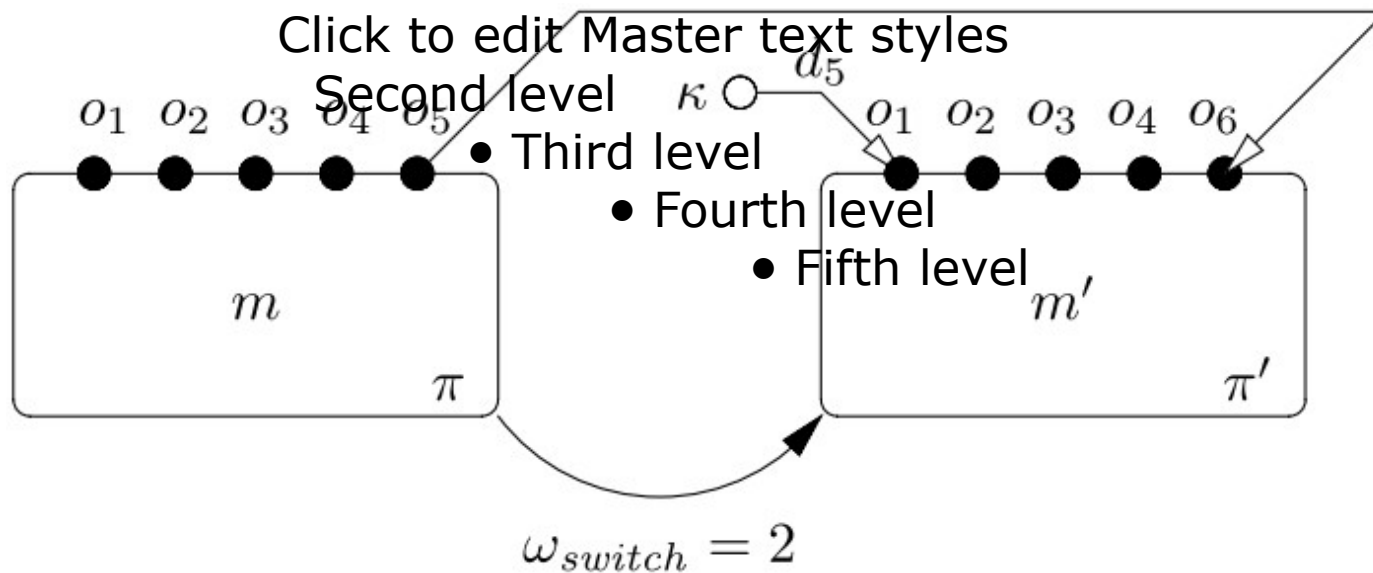
Informelle Beschreibung - mode invocation



Informelle Beschreibung - mode switch: zweiter mode m'



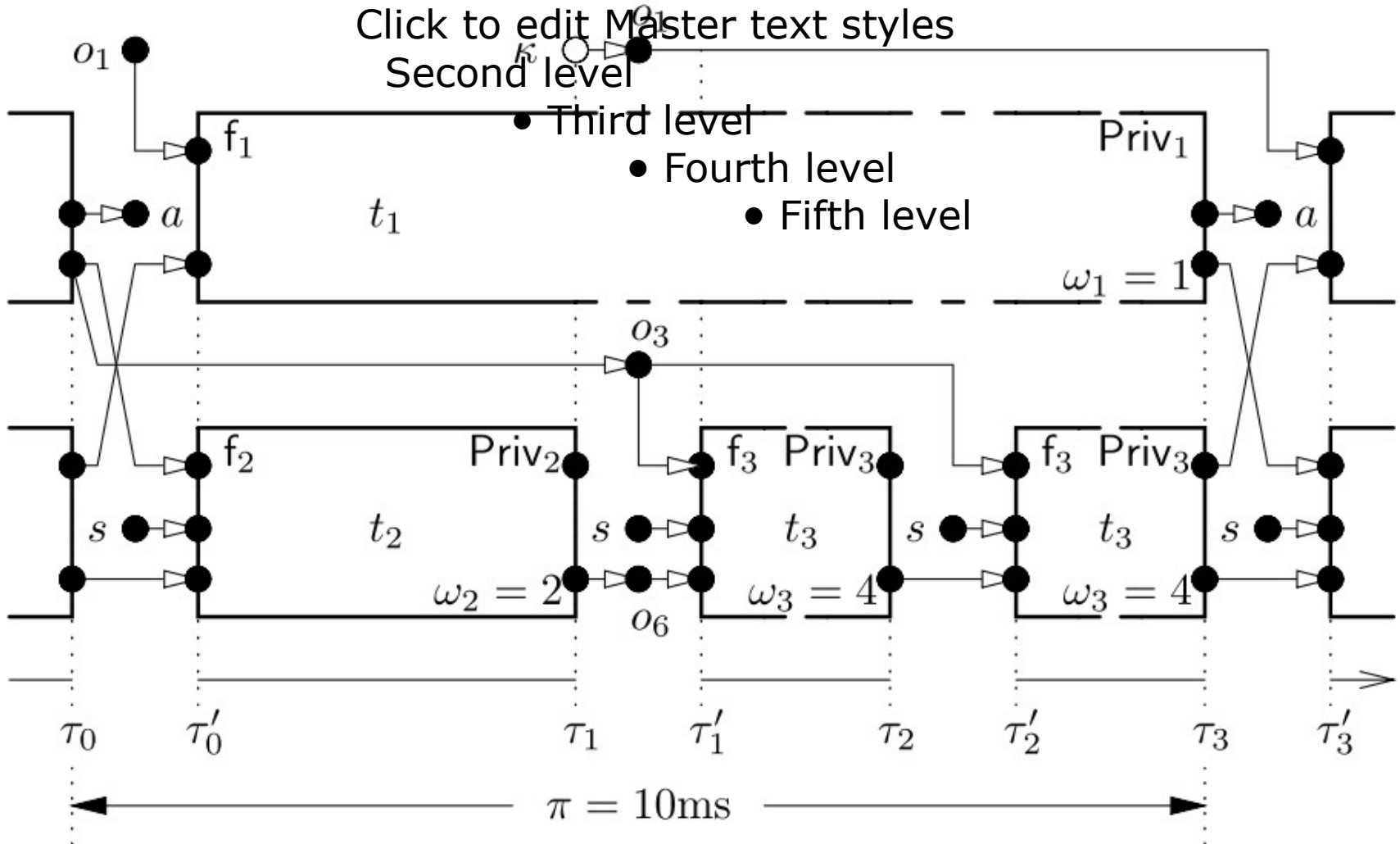
Informelle Beschreibung - mode switch: $m \leftrightarrow m'$ (Überblick)



Informelle Beschreibung - mode

switch:

m \square m' (Ausführung bei 5ms)



- Semantik von Giotto wurde operationell beschrieben
 - Aufstellen einer *configuration* $C = (\tau, m, u, v, \sigma_{active})$
- Aktualisieren einer *configuration*
 - Erfolgt in 9 Schritten
- *Giotto Programm* besteht aus einer unendlichen Folge:
 - der *program configuration* mit:
 - (1) $(0, start, 0, v, leer)$
 - (2) Nachfolgekongfiguration für alle $i \geq 0$

C_1, C_2, \dots

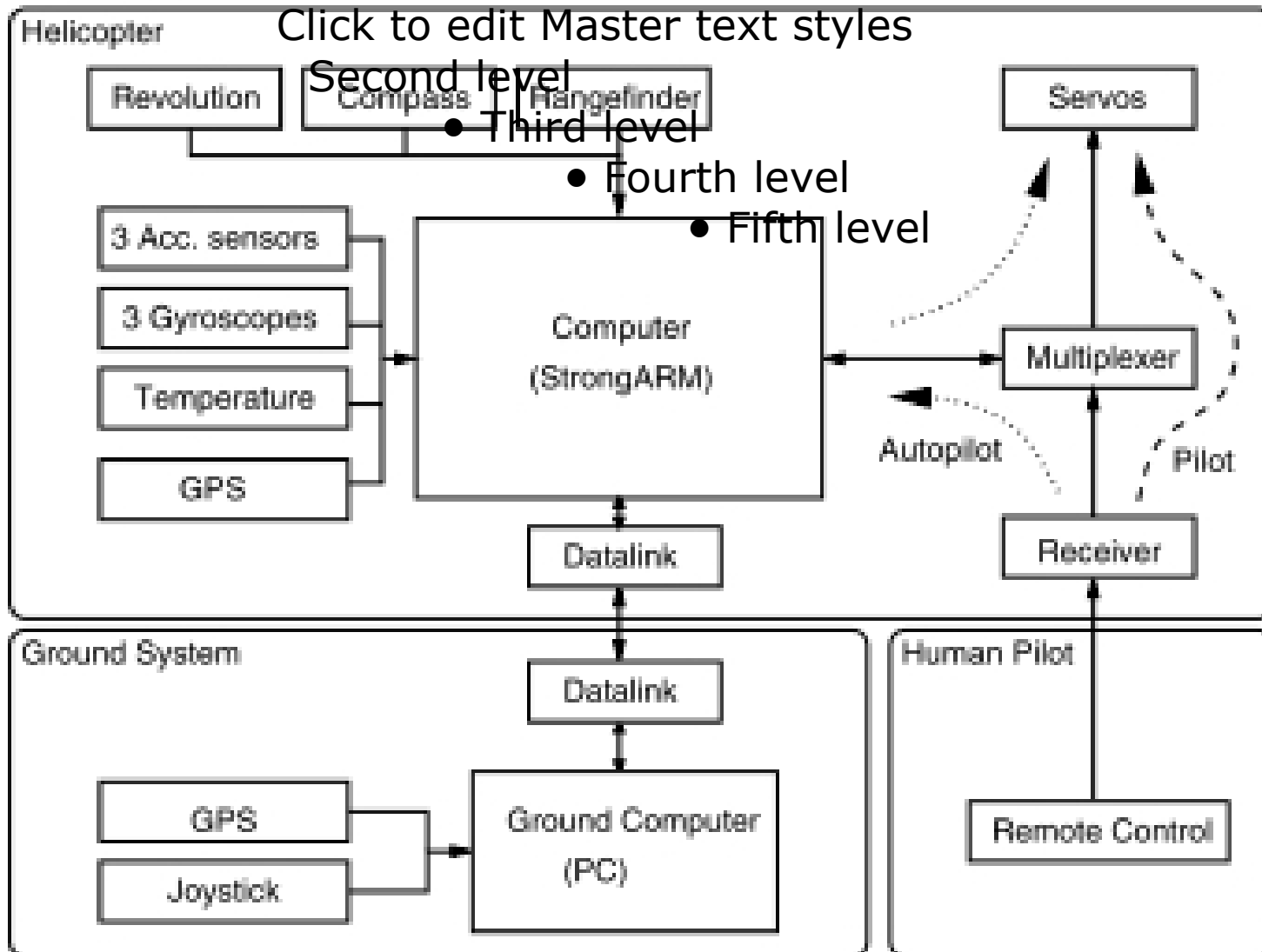
C_i

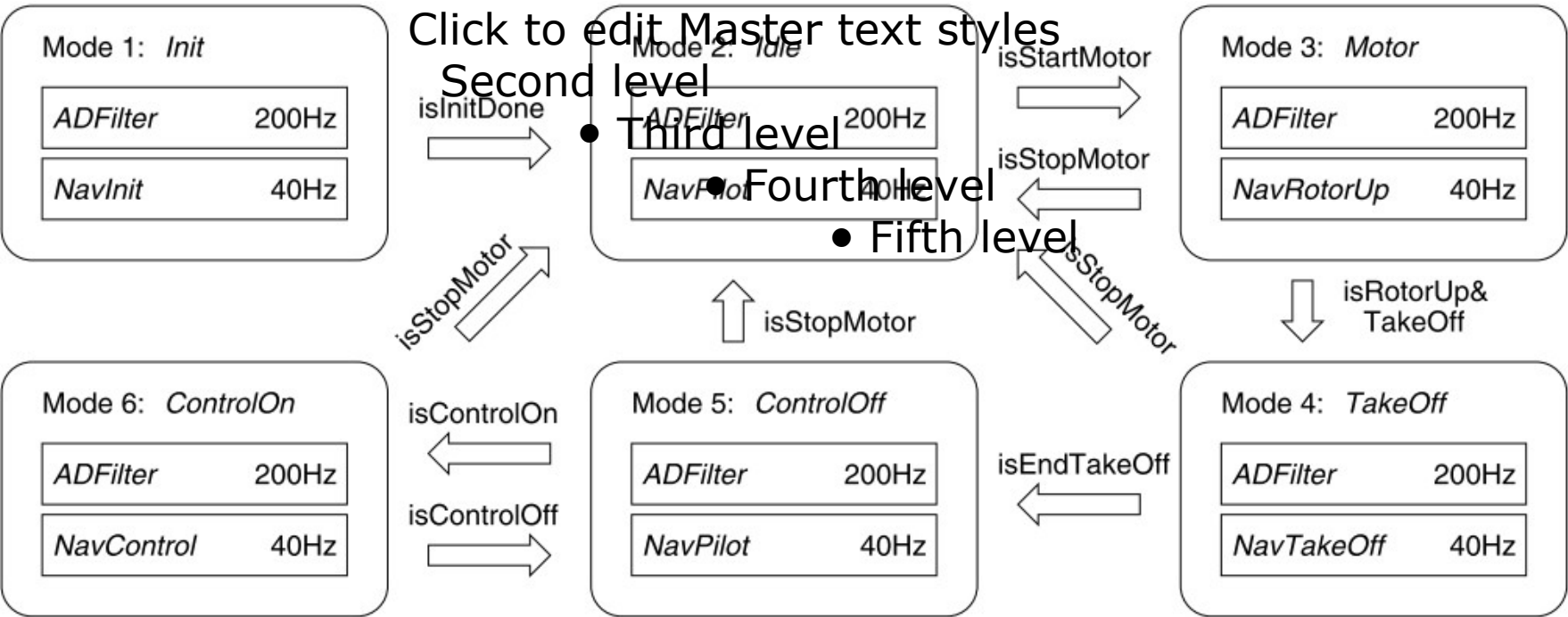
$C_0 =$

$C_{i+1} =$

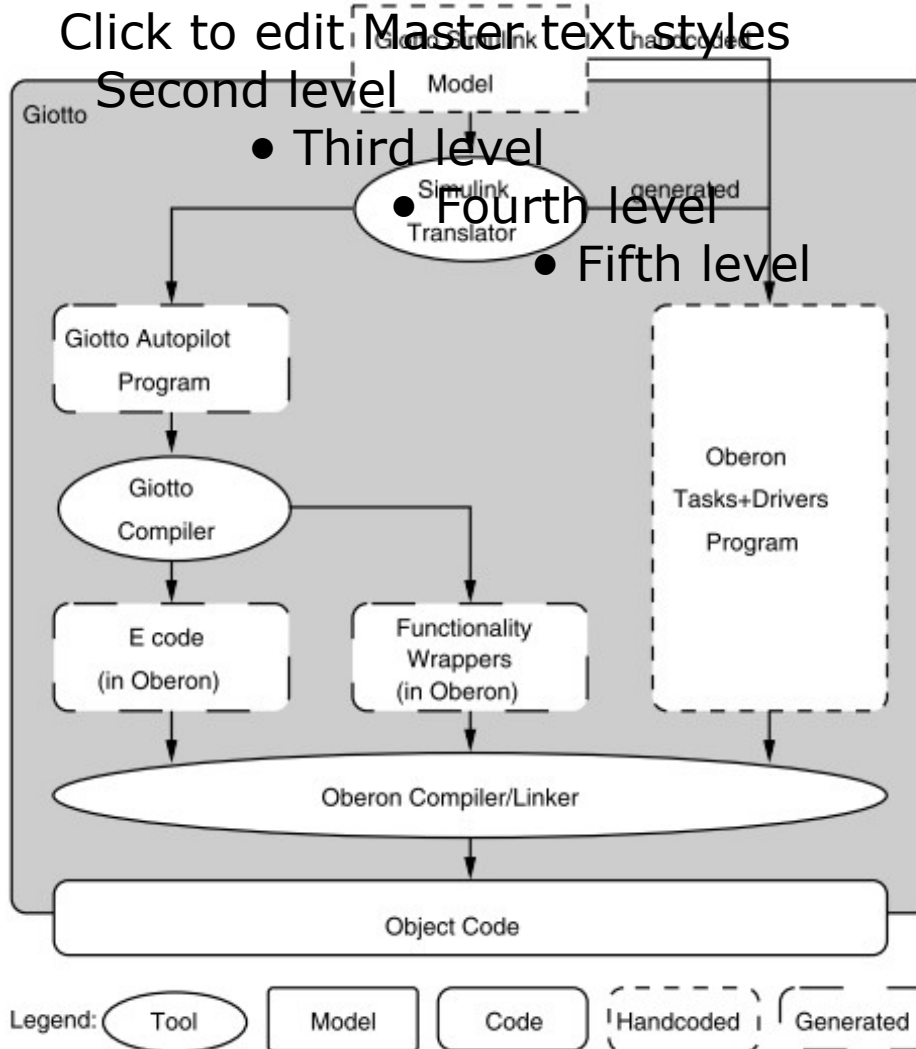
Ein Giotto Program - Ausschnitt für einen Autopilot des OLGA Helikopters



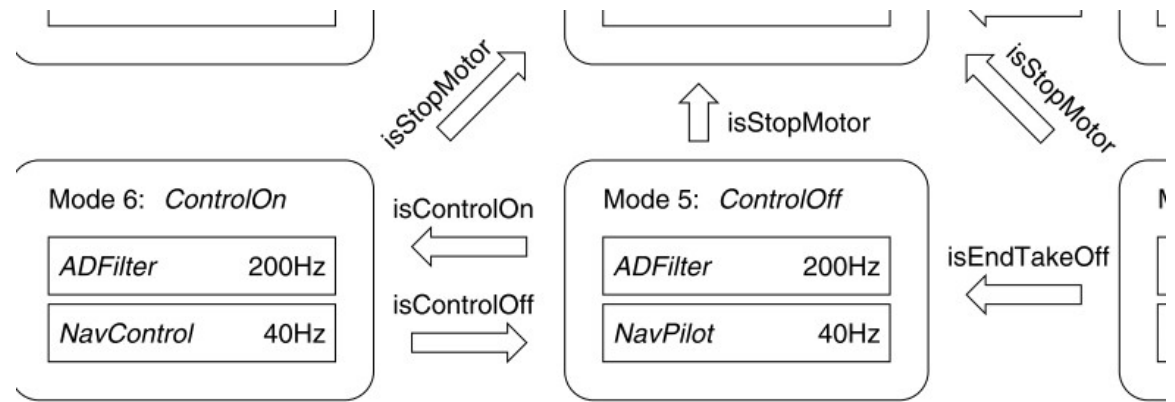




Vom Modell bis zum real-time Code: Überblick



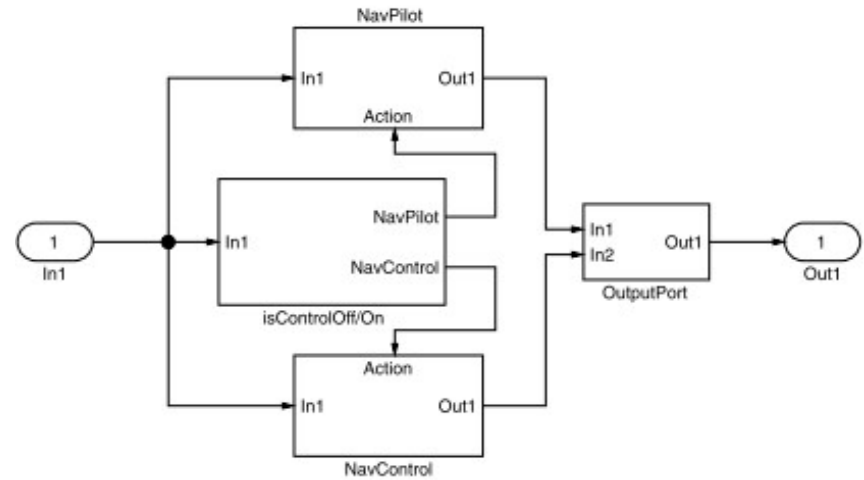
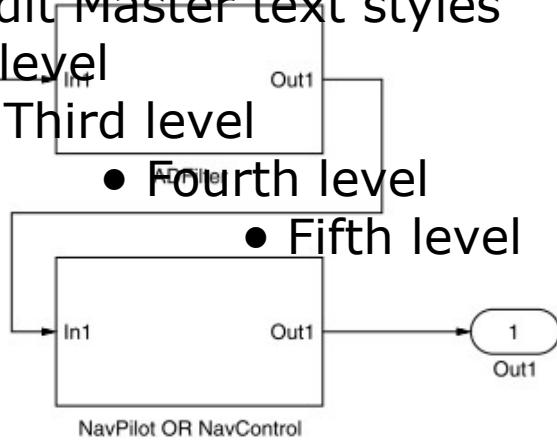
Vom Modell bis zum real-time Code: Simulink Modell



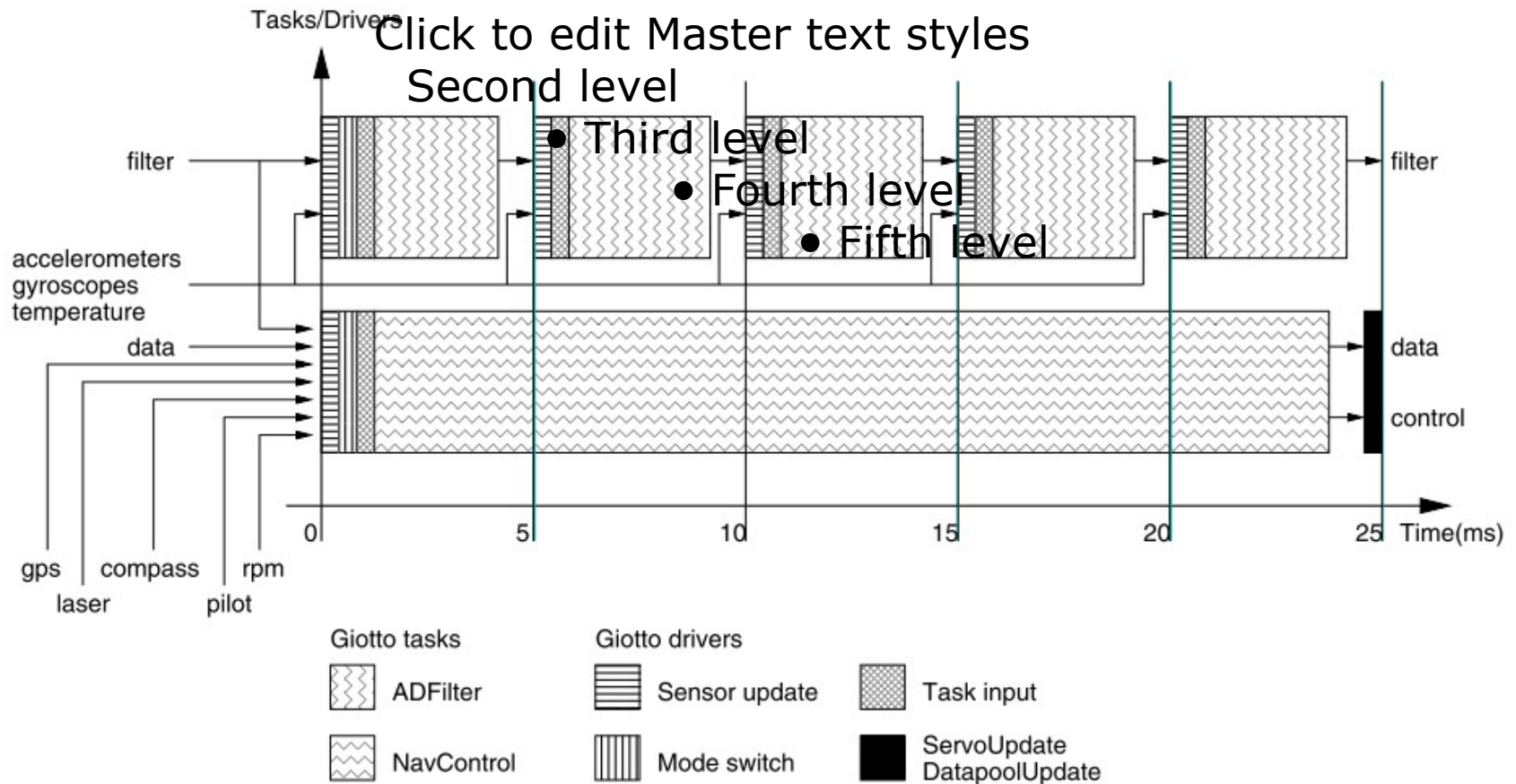
Click to edit Master text styles

Second level

- Third level
- Fourth level
- Fifth level



Giotto Programm: Was wird benötigt ?



```
mode ClickControlMaster text styles 25 {
  actfreq 1 do ServoUpdate;
  actfreq 1 do DataPoolUpdate;
  exitfreq 1 do ControlOn;
  taskfreq 5 do ADFilter;
  taskfreq 1 do NavPilot;}
mode ControlOn() period 25 {
  actfreq 1 do ServoUpdate;
  actfreq 1 do DataPoolUpdate;
  exitfreq 1 do ControlOff;
  taskfreq 5 do ADFilter;
  taskfreq 1 do NavControl;}
```

Das Giotto Programm: sensor port Definition

sensor	Click to edit Master text styles	
GPSPort	Second level	GPSGet ;
LaserPort	laser • Third level	LaserGet ;
CompassPort	compass • Fourth level	CompassGet ;
RPMPort	rpm uses • Fifth level	RotorGet ;
ServoPort	pilot uses	ServoGet ;
AnalogPort	accelerometers uses	AccGet ;
AnalogPort	gyroscopes uses	GyrosGet ;
AnalogPort	temperature uses	TempGet ;
BoolPort	startswitch uses	StartSwitchGet ;
BoolPort	stopswitch uses	StopSwitchGet ;

Das Giotto Programm: mode driver & sensor update Definition

```
driver ControlDef MasterTextStyles output () {  
    switch isControlOff(stopswitch) }
```

- Second level
 - Third level
 - Fourth level
 - Fifth level

output

```
AnalogPort    filter    :=  FilterInit;  
ServoPort     control   :=  ServoInit;  
DataPoolPort  data      :=  DataPoolInit;
```

Das Giotto Programm: task & actuatorDefinition

```
task ADFilter(accelerometers, gyroscopes, temperature, filter)
  output (filter) {
    schedule ADFilterImplementation(accelerometers, gyroscopes,
    temperature, filter)}
task NavControl(gps, laser, compass, filter, rpm, pilot, data)
  output (control, data) {
    schedule NavControlImplementation(gps, laser, compass, filter,
    rpm, pilot, control, data)}
```

Click to edit Master text styles

Second level

• Third level

• Fourth level

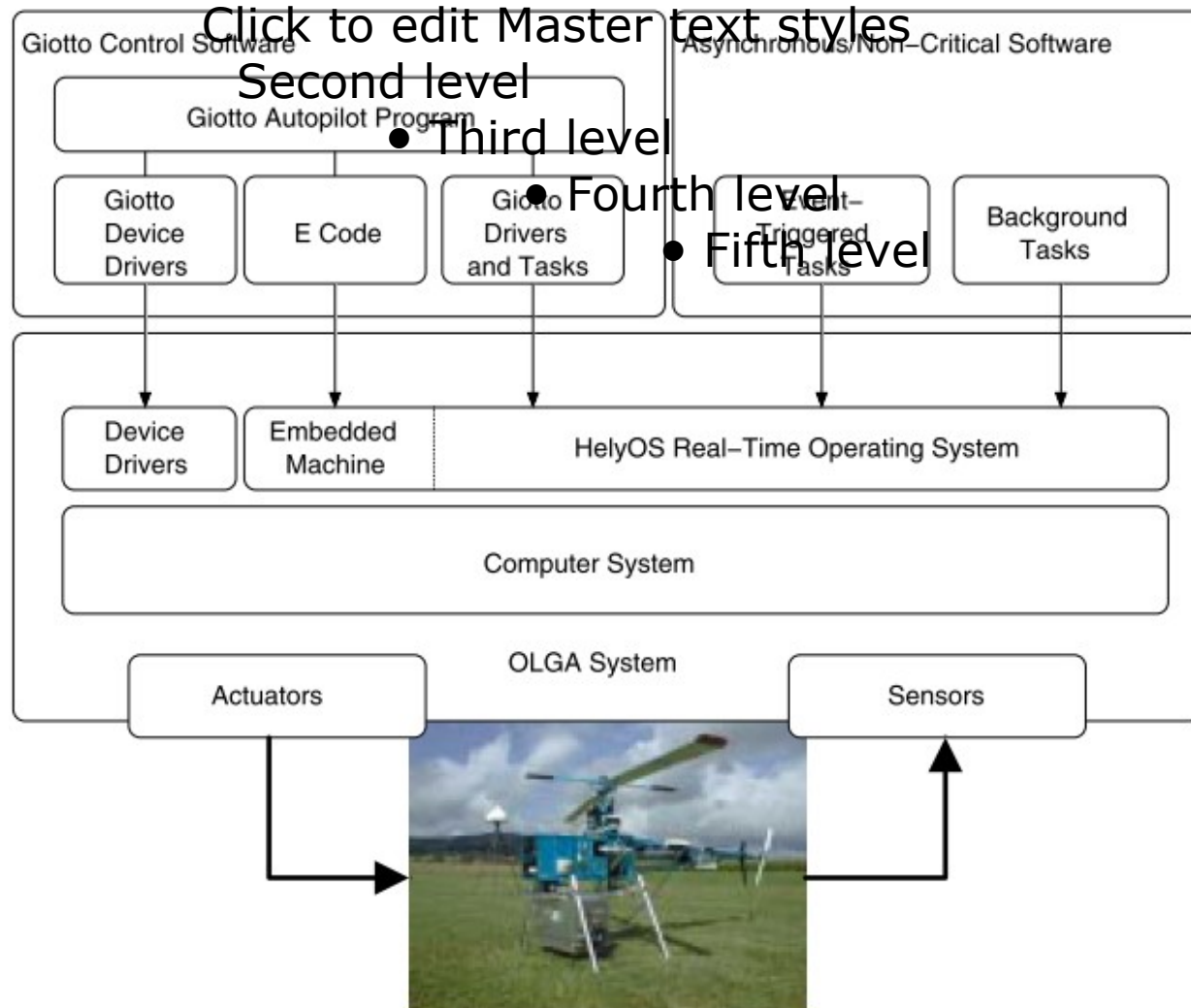
• Fifth level

actuator

```
ServoPort servos uses ServoPut;
DataPoolPort datapool uses DataPoolPut;
```

Das Giotto Programm: ready for compile ?

- Giotto Compiler:
 - Aufgabe: erzeugt *timing code* für ein Giotto Programm für eine bestimmte Plattform
 - Möglich durch Angabe von oberen Schranken für Laufzeit von *tasks*
- Giotto Annotation
 - Giotto-H
 - Giotto-HM
 - Giotto-HMS



Vielen Dank!

- Henzinger, T.; Horowitz, B. & Kirsch, C.
Giotto: a time-triggered language for embedded programming
Proceedings of the IEEE, **2003**, 91, 84 - 99
- Kirsch, C. M.
Principles of Real-Time Programming
in Proc. Second International Workshop on Embedded Software (EMSOFT), LNCS 2491, Springer Verlag, {2002}, {61-75}
- Henzinger, T. A. & Kirsch, C. M.
The embedded machine: Predictable, portable real-time code
ACM Trans. Program. Lang. Syst., ACM, 2007, 29
- Kirsch, C. M.; Sanvido, M. A. A.; Henzinger, T. A. & Pree, W.
A Giotto-Based Helicopter Control System
Proceedings of the Second International Conference on Embedded Software, Springer-Verlag, 2002, 46-60