

CISS-Projekter related to Modeldriven Software Development

Arne Skou, CISS



01



01



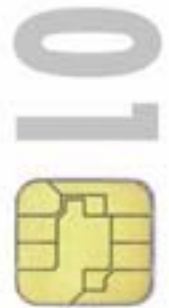
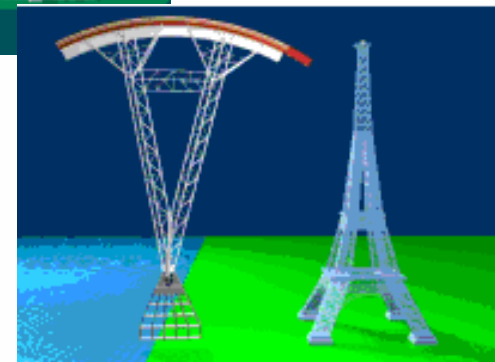
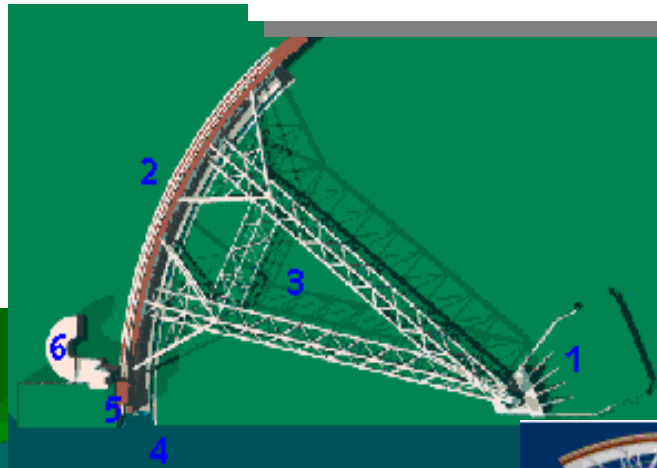
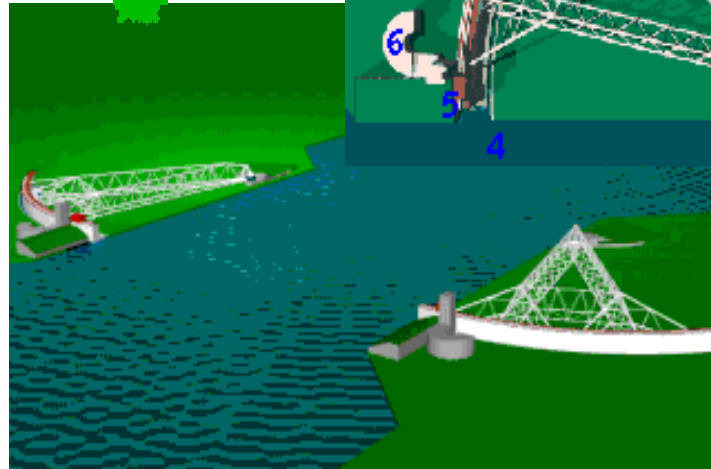
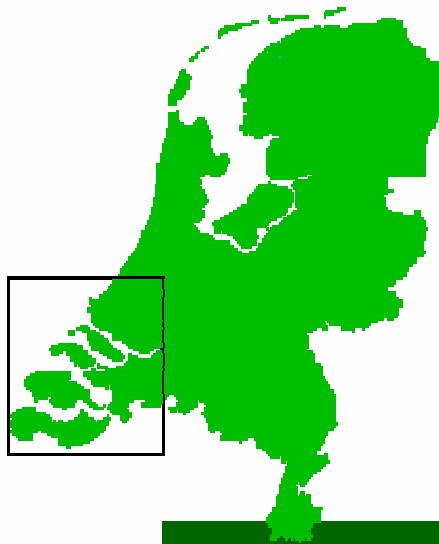
0101

1



Rotterdam Stormflodssikring

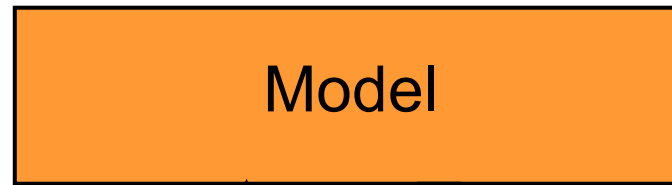
- ARIANE-5
- INTEL Pentium II floating-point division
470 Mill US \$
- Mars Pathfinder
- Radiation therapy, Therac-25



Del af løsning: Brug modeller

Udregninger

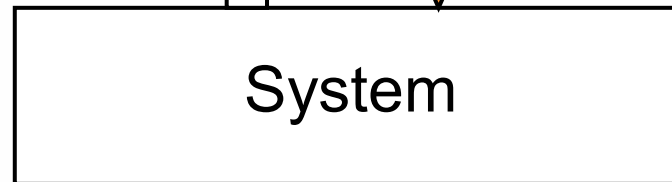
Matematik



Abstraktion

Forudsigelser

Test



Broer 😊

Fly 😊

Biler 😊

Software ☹️



0101010101



01010101

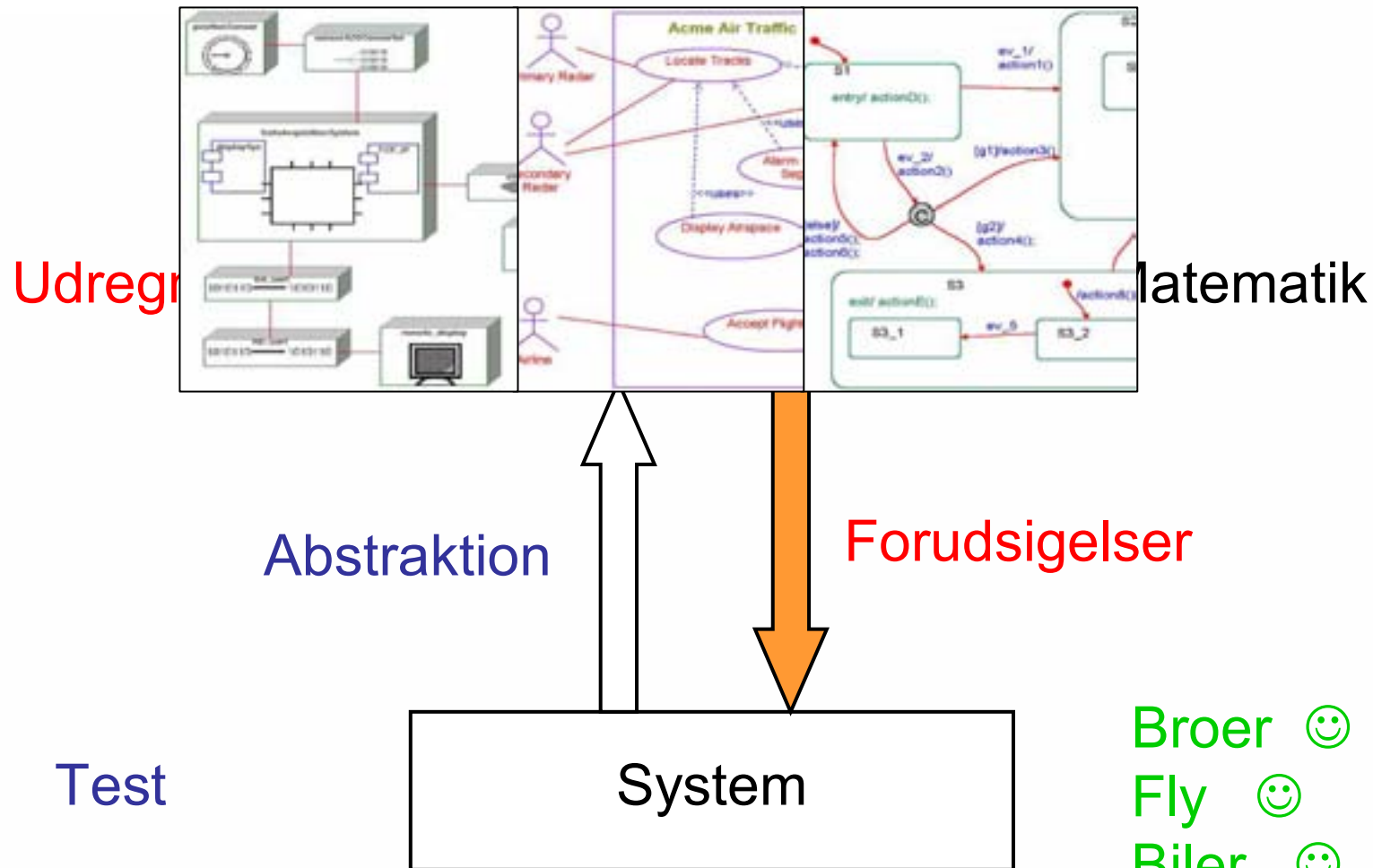


010101

0101



Del af løsning: Brug modeller



- Broer 😊
- Fly 😊
- Biler 😊
- Software ☹️



Ultimativt mål: Kun een systemdokumentation



01010101



01010101



01010101

01010101

01010101



Analysis

Validation

Design Model ↔ **Specification**

Verification & Refusal

Automatic
Code generation

Automatic
Test generation

Implementation

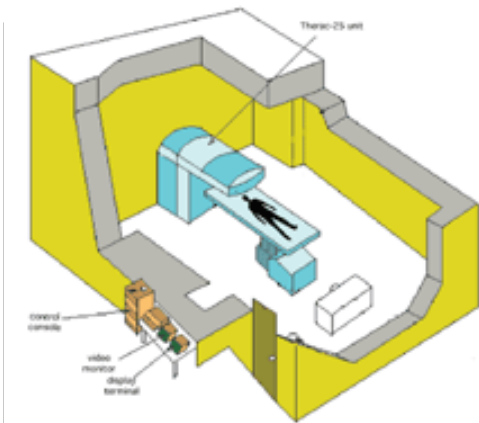
Testing

FORMAL METHODS

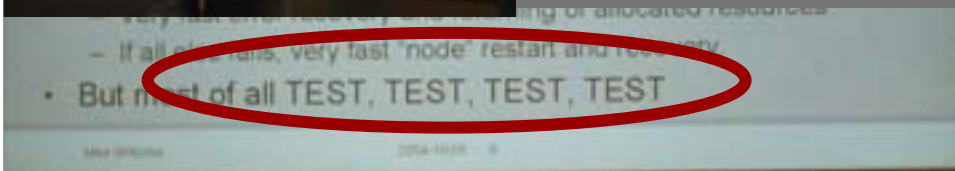
UML

- Værktøjer:
- Rhapsody
 - Visualstate
 - Telelogic
 - Rational
 - Uppaal

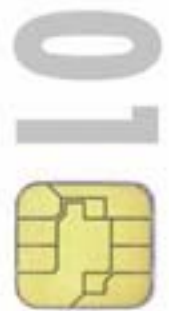
Hvorfor T&V ?



Michael Williams
Research Director, Ericsson,
SE

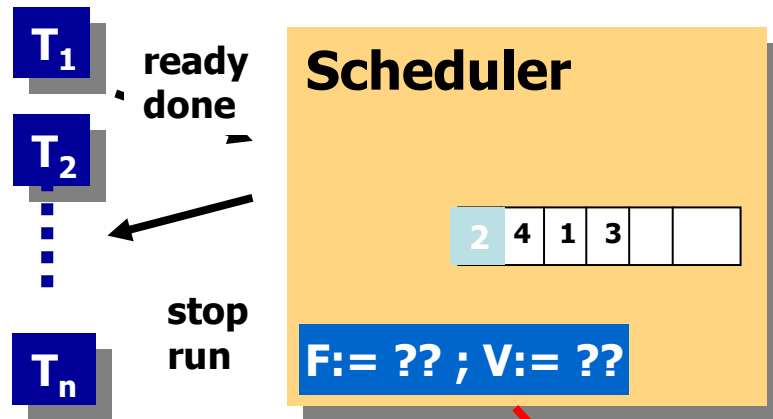


- Fejl i indlejret software forbundet med voldsomme udgifter.
- 30-40% af udviklingstid bruges på tidskrævende, ad-hoc aftestning.
- Potentialet for forbedrede metoder og værktøjer enormt.
- "Time-to-market" kan reducere betydeligt ved brug af tidlig verifikation og performanceanalyse

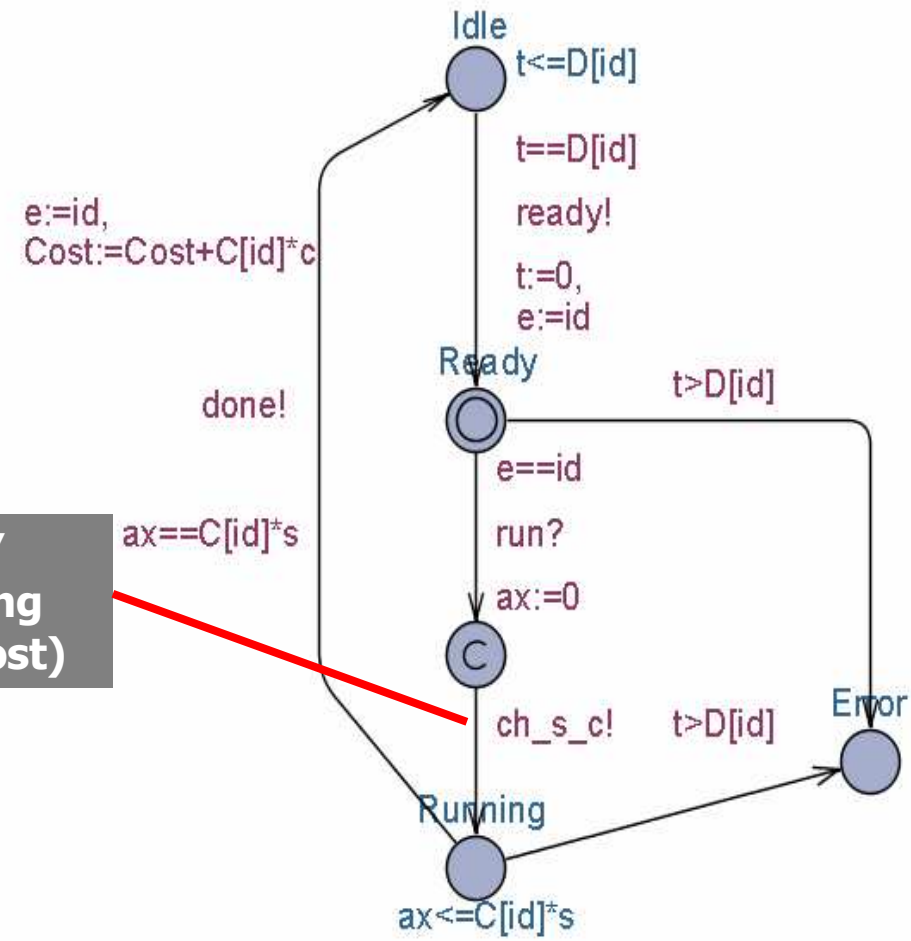
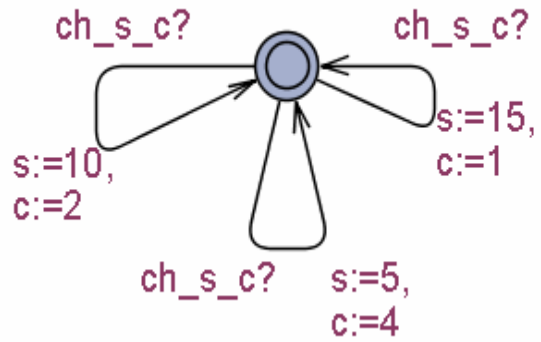


Energy Optimal Scheduling- Triggered by Analog Devices

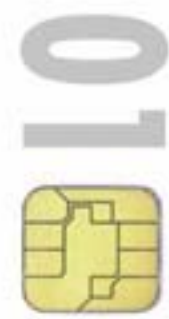
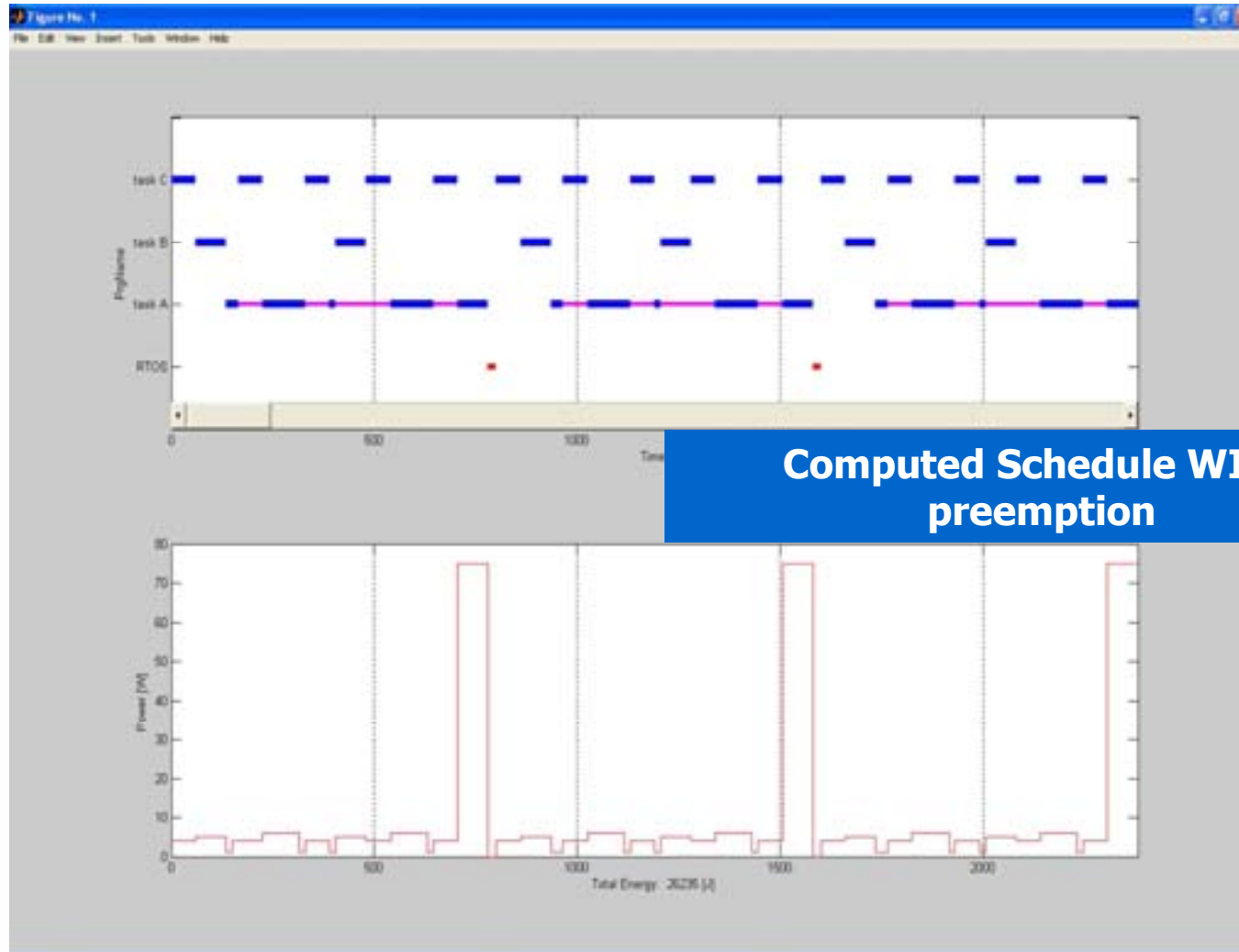
Using PTA 



"Choose" Freq/Scaling (Voltage/Cost)



Preliminary Results



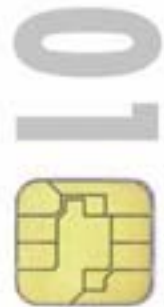
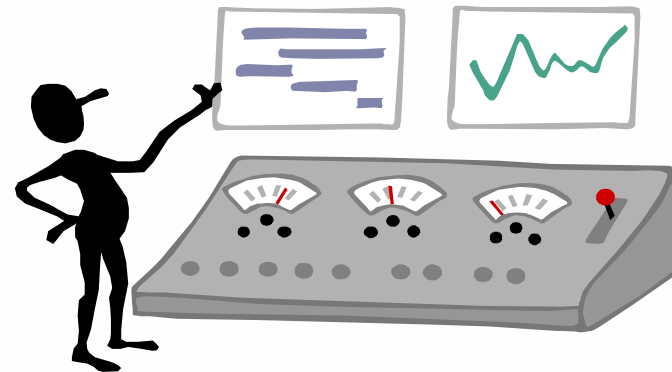
Testing Embedded Software

Testing:

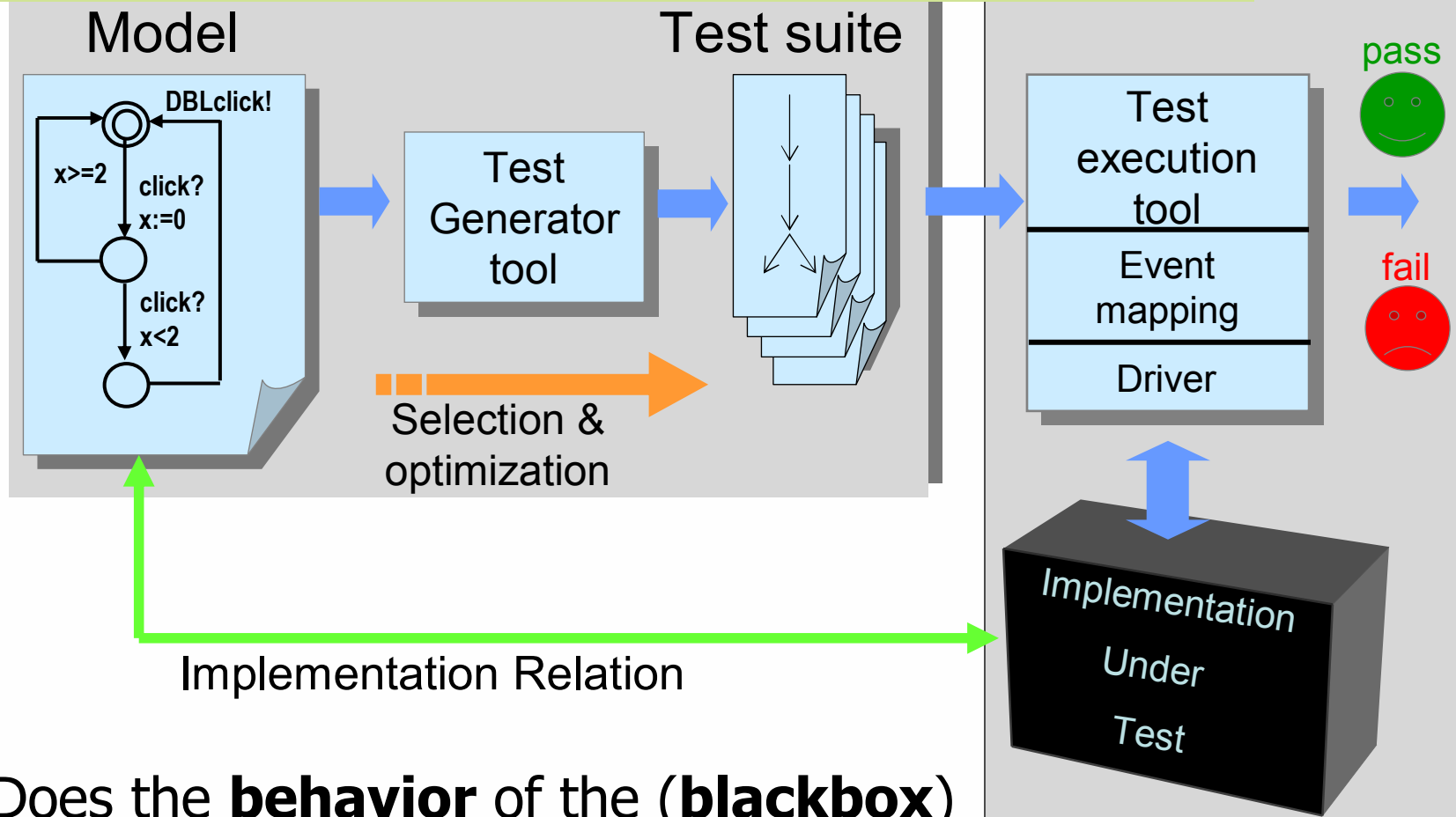
- to check the **quality** (functionality, reliability, performance, ...) of an (software) object
 - by performing experiments
 - in a controlled way

- 10-20 errors per 1000 LOC
- 30-50 % of development time and cost in embedded software

- To find errors
- To determine risk of release



Automated Model Based Conformance Testing

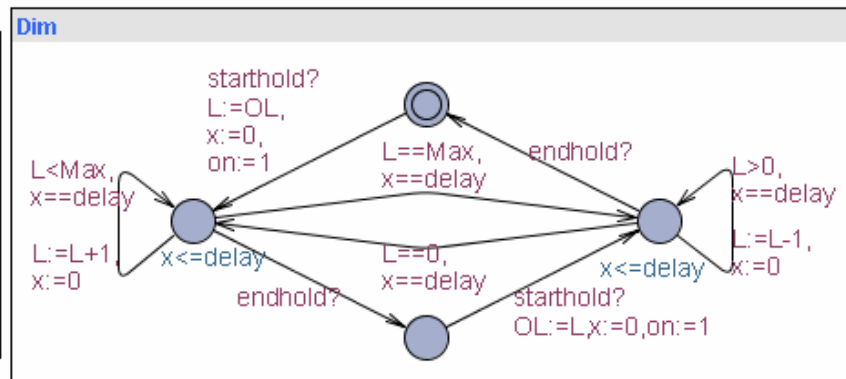
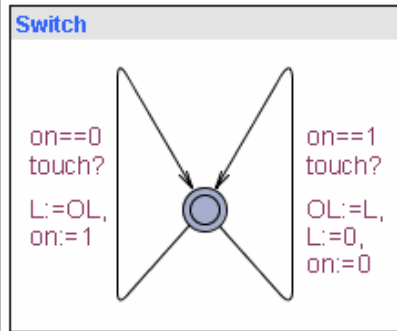
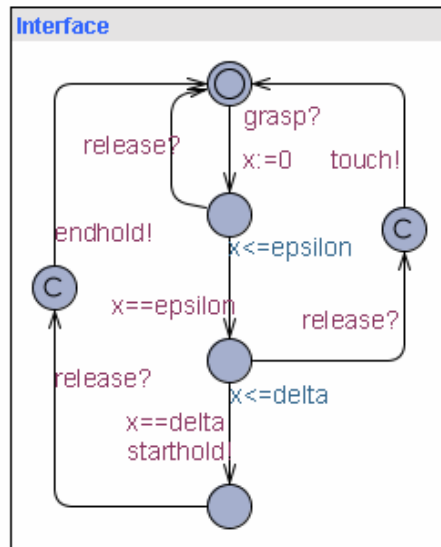


Does the **behavior** of the (**blackbox**) implementation **comply** to that of the specification?



Offline Generated Tests

Cost=12600 ms



Fastest Edge Covering Sequence

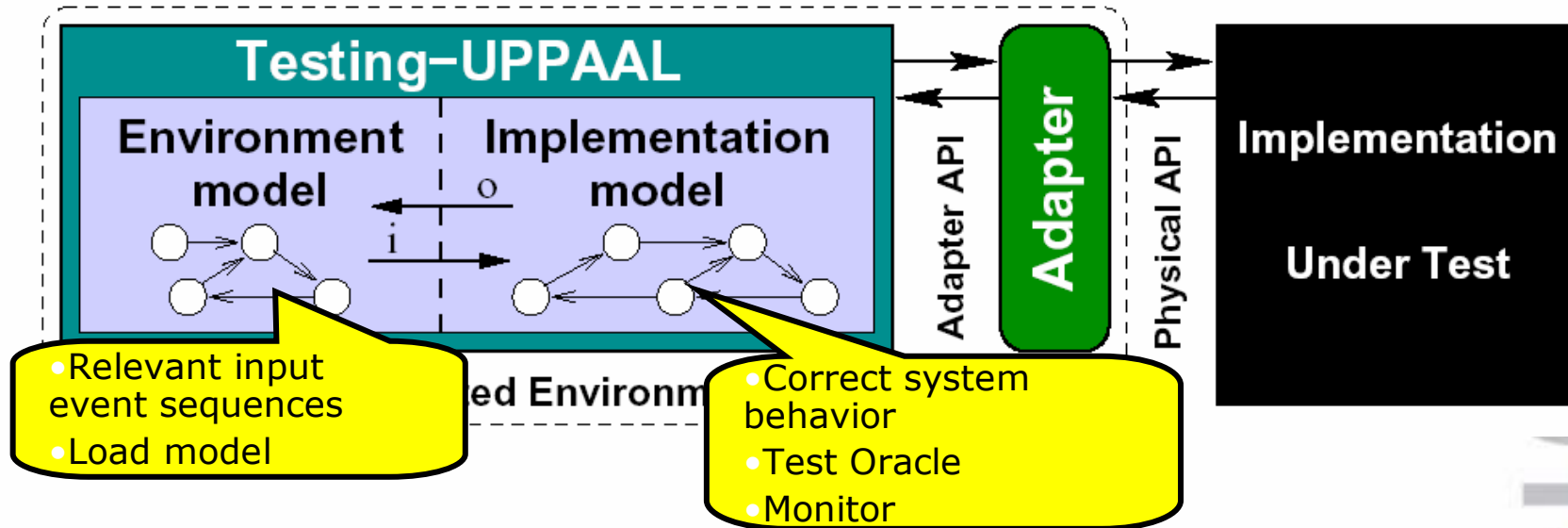
```
grasp!·200·release!·L(0)?·grasp·
200·release!·L(0)?·grasp·500·L(0)·
release!·grasp·500·1000·L(1)?·
1000·L(2)?·1000·L(3)?·1000·
L(4)?·1000·L(5)?·1000·L(6)?·1000·
L(7)?·1000·L(8)?·1000·L(9)?·1000·
L(10)?·1000·L(9)?·release!·grasp!·
release!·pass
```

Java Test Case Implementation



```
out(IGrasp); //touch:switch light on
silence(200);
out(IRRelease);
in(OSetLevel,0);
out(IGrasp); //@200 // touch: switch light off
silence(200);
out(IRRelease); //touch
in(OSetLevel,0);
//9
out(IGrasp); //@400 //Bring dimmer from ActiveUp
silence(500); //hold //To Passive DN (level=0)
in(OSetLevel,0);
out(IRRelease);
//13
out(IGrasp); //@900 // Bring dimmer PassiveDn->ActiveDn->
silence(500); //hold // ActiveUP+increase to level 10
silence(1000); in(OSetLevel,1); silence(1000); in(OSetLevel,2);
silence(1000); in(OSetLevel,3); silence(1000); in(OSetLevel,4);
silence(1000); in(OSetLevel,5); silence(1000); in(OSetLevel,6);
silence(1000); in(OSetLevel,7); silence(1000); in(OSetLevel,8);
silence(1000); in(OSetLevel,9);silence(1000); in(OSetLevel,10);
silence(1000); in(OSetLevel,9); //bring dimm State to ActiveDN
out(IRRelease); //check release->grasp is ignored
out(IGrasp); //@12400
out(IRRelease);
silence(dfTolerance);
```

On-line Testing



User Supplied Test Specification

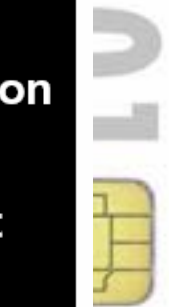
Closed TA Network partitioned into Env and IUT.

IUT model weakly input input enabled

Model of Environment

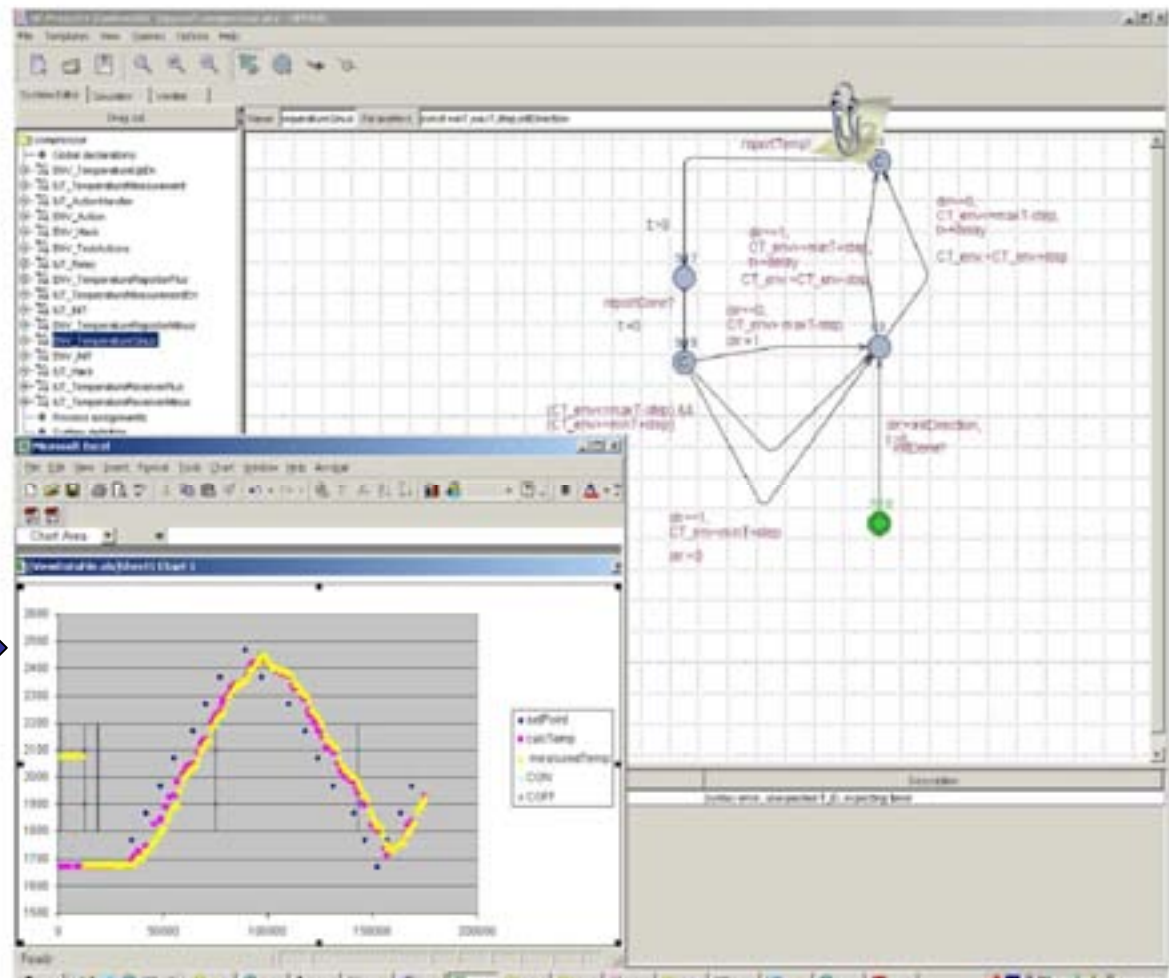
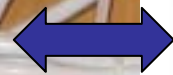
Designate observable input and output actions.

Specify amount of real time per one time-unit in model.



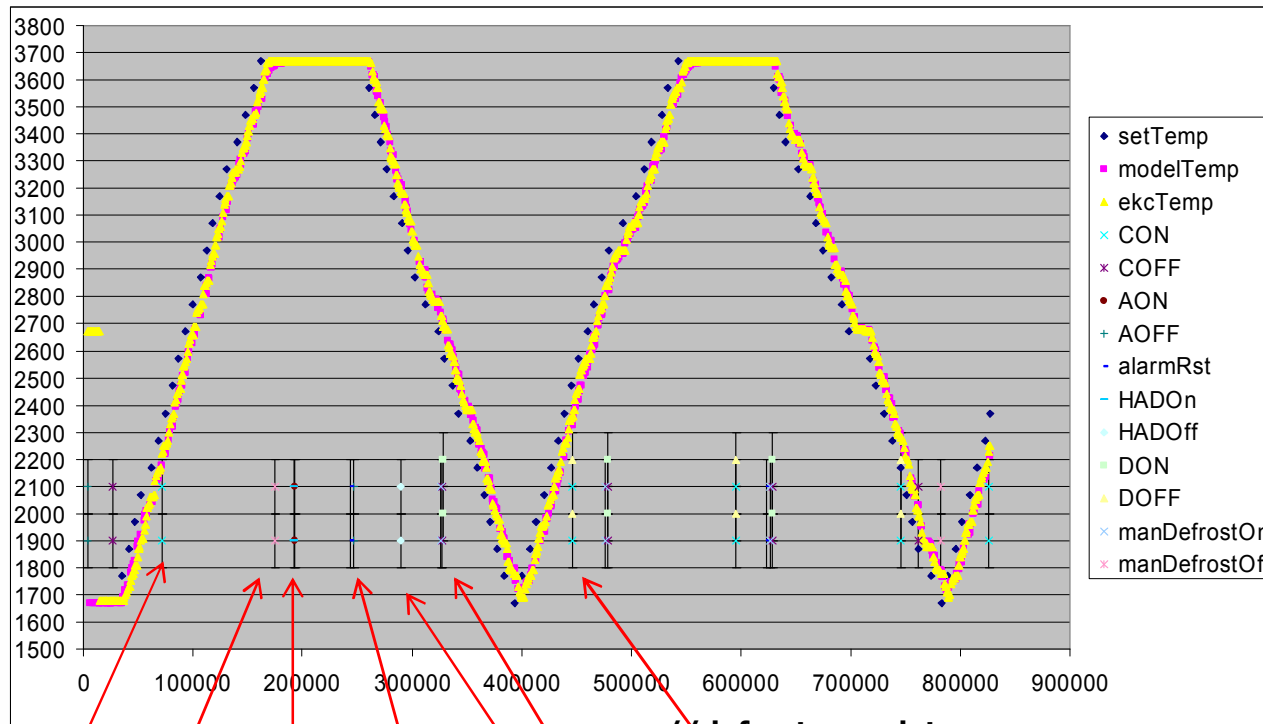
First Industrial Application

Danfoss Electronic Cooling Controller



Example Test Run

(log visualization)



compressorOn!

defrostOff?

alarmOn!
alarmDisplayOn!

resetAlarm?
AOFF!

HighAlarmDisplayOff!

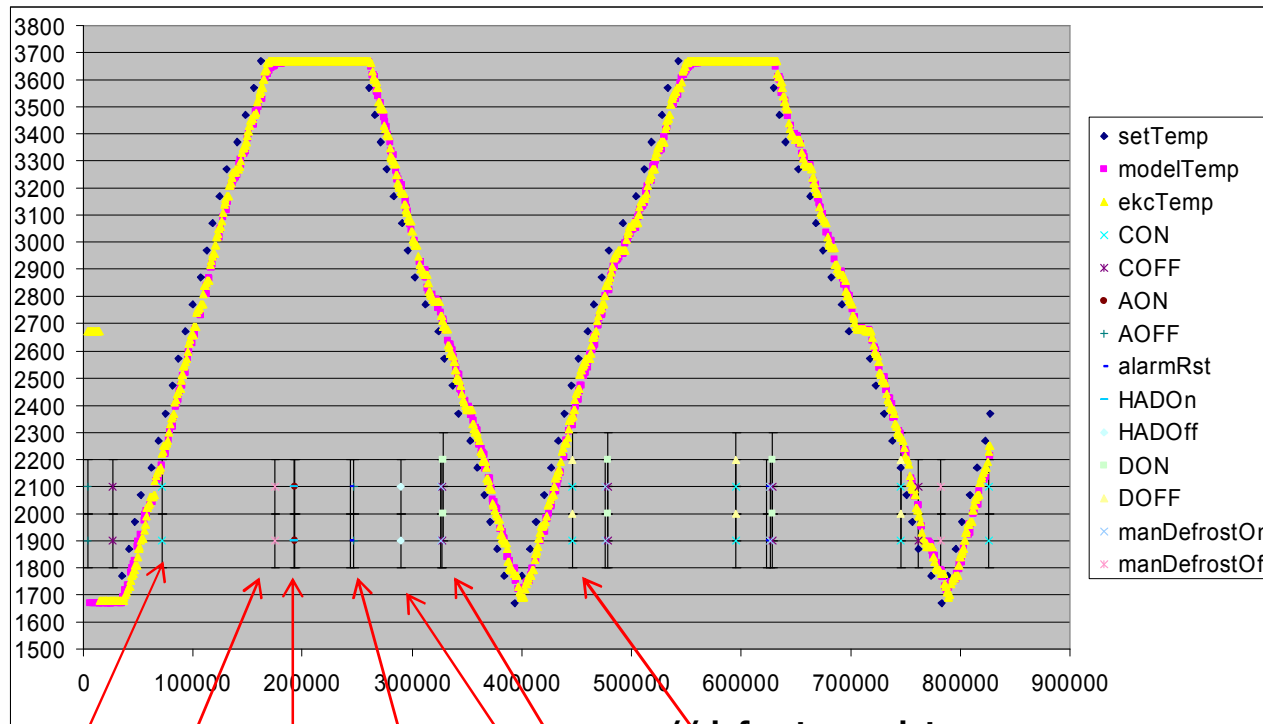
manualDefrostOn?
COFF!
DON!

//defrost complete
DOFF!
CON!



Example Test Run 1

(log visualization)



compressorOn!
 defrostOff?
 alarmOn!
 alarmDisplayOn!
 resetAlarm?
 AOFF!
 HighAlarmDisplayOff!
 manualDefrostOn?
 COFF!
 DON!
 //defrost complete
 DOFF!
 CON!



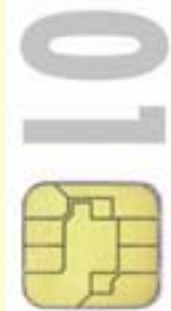
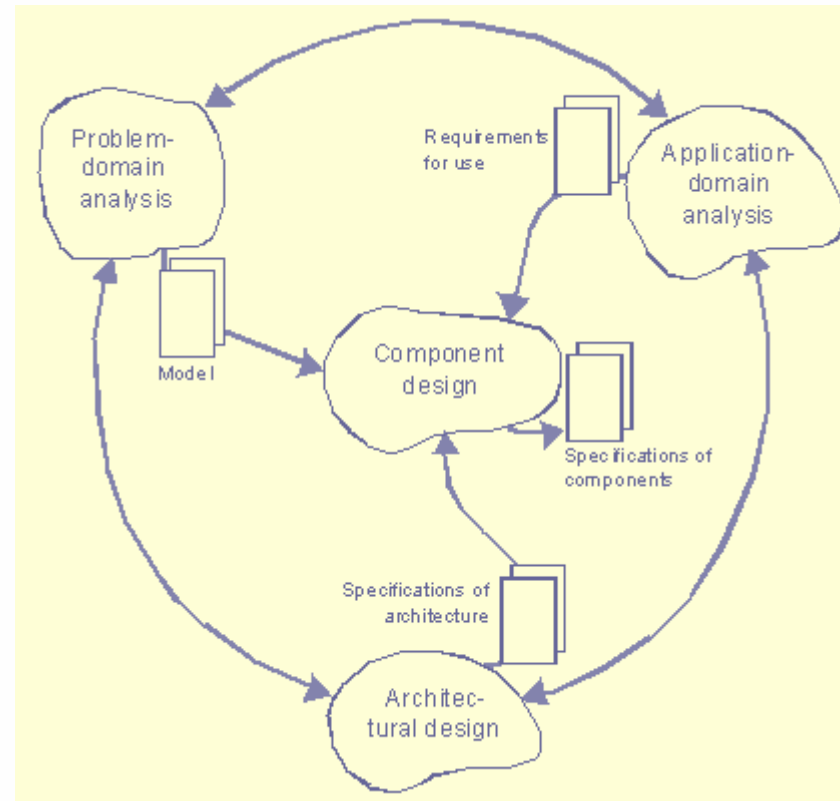
Feasibility studies – a collaboration project with Siemens Mobile

- A feasibility study should ideally reveal all possible risks.
- It should not take very long time – compared to the overall project time.
- The results should be usable in the final project.

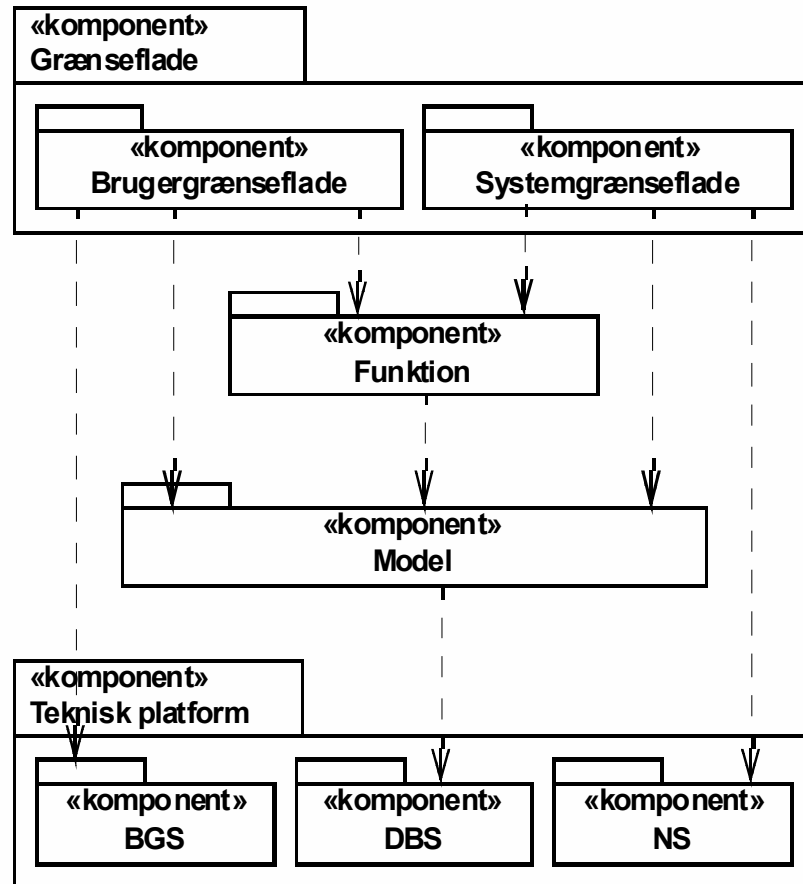


The Aalborg OOA&D method (Mathiassen et.al.)

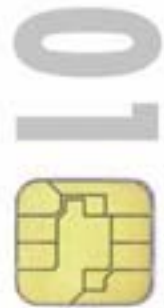
- May be interpreted as a linear or as an iterative method.
- A first iteration may be seen as a feasibility study.
- Based on the UML notation.



Pattern: Generic basic architecture

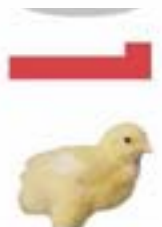
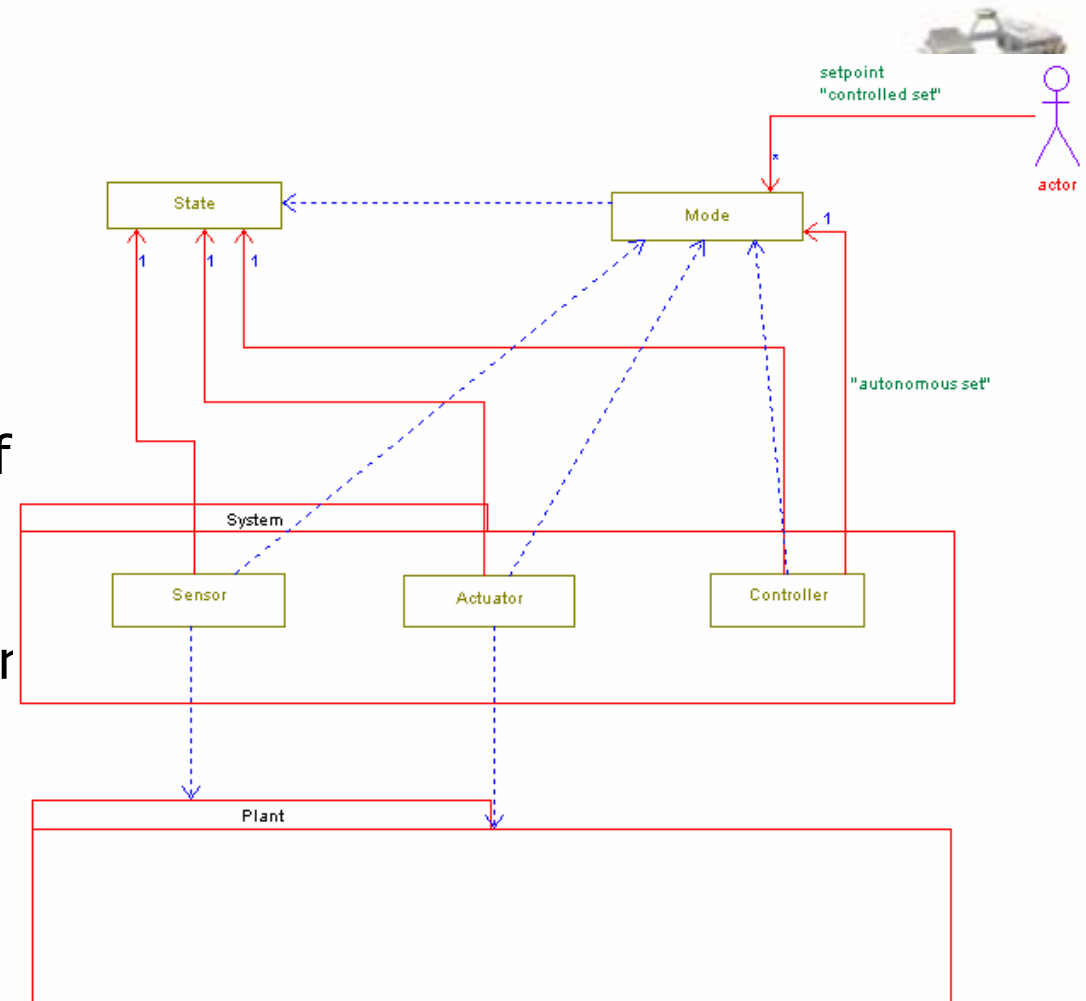


- The architecture reflects the division of the environment into problem and application domain.
- “Technical platform” is an extension and encapsulation of the underlying technical platform



Model based development at Man B&W - Aalborg

- Application of a UML based tool on the development of electronic engine control
- Reverse engineering of existing system
- Development of a generic architecture for embedded control systems



The Prosoft project: IT corridor project 2004-2005

- Joint collaboration between CISS, HiH and 10 companies.
- 1 main project
- 5 sub projects
- Year 1: Feasibility studies/requirement specifications
- Year 2: Prototypes and experiments

