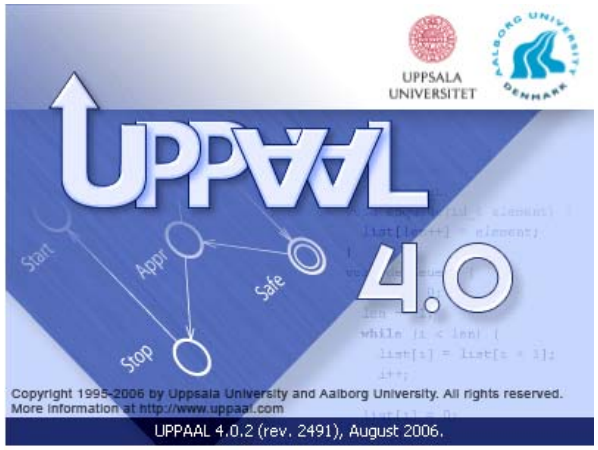


Validation, Synthesis and Performance Evaluation of Embedded Systems

using UPPAAL

Kim Guldstrand Larsen

CISS: *Center for Embedded Software Systems*

Kim Guldstrand Larsen
kgl@cs.auc.dk
96358893

or

CISS
www.ciss.dk
info@ciss.dk
96357220

Aalborg Universitet
Fr. Bajersvej 7B
9220 Aalborg Ø

Aalborg

Aalborg University leading Danish ICT University in terms of public investments (33%)

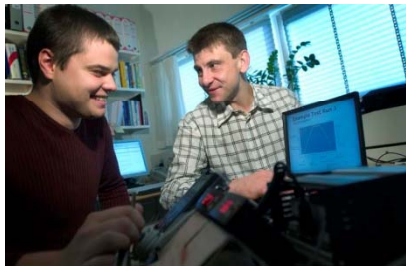
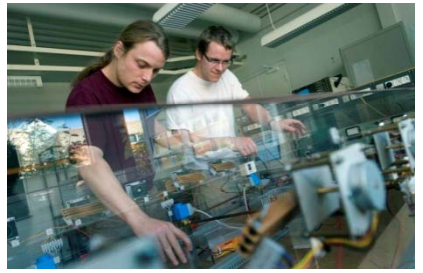
Why CISS

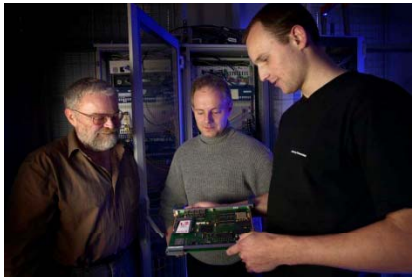
- 80% of all software is embedded
- Demands for
  increased functionality
  with
  minimal resources
- Requires multitude of skills
  - Software construction
  - Hardware platforms
  - Communication
  - Automation
- **Goal**:
  Give a qualitative lift to current industrial practice
  !!!!!

CISS in Numbers

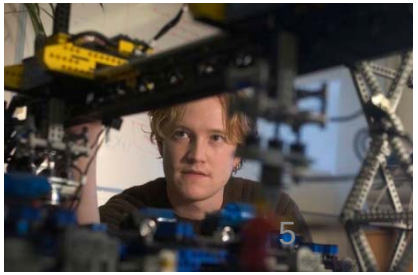
- National Competence Center (2003-..)
  - Ministry of Tech. & Res.
  - North Jutland
  - Aalborg City
  - Aalborg University

- **50** Industrial Projects
- **20** CISS employees
- **25** CISS ass. Res.
- **20** Industrial PhDs
- **10** Elite Students
- **10** MEUR





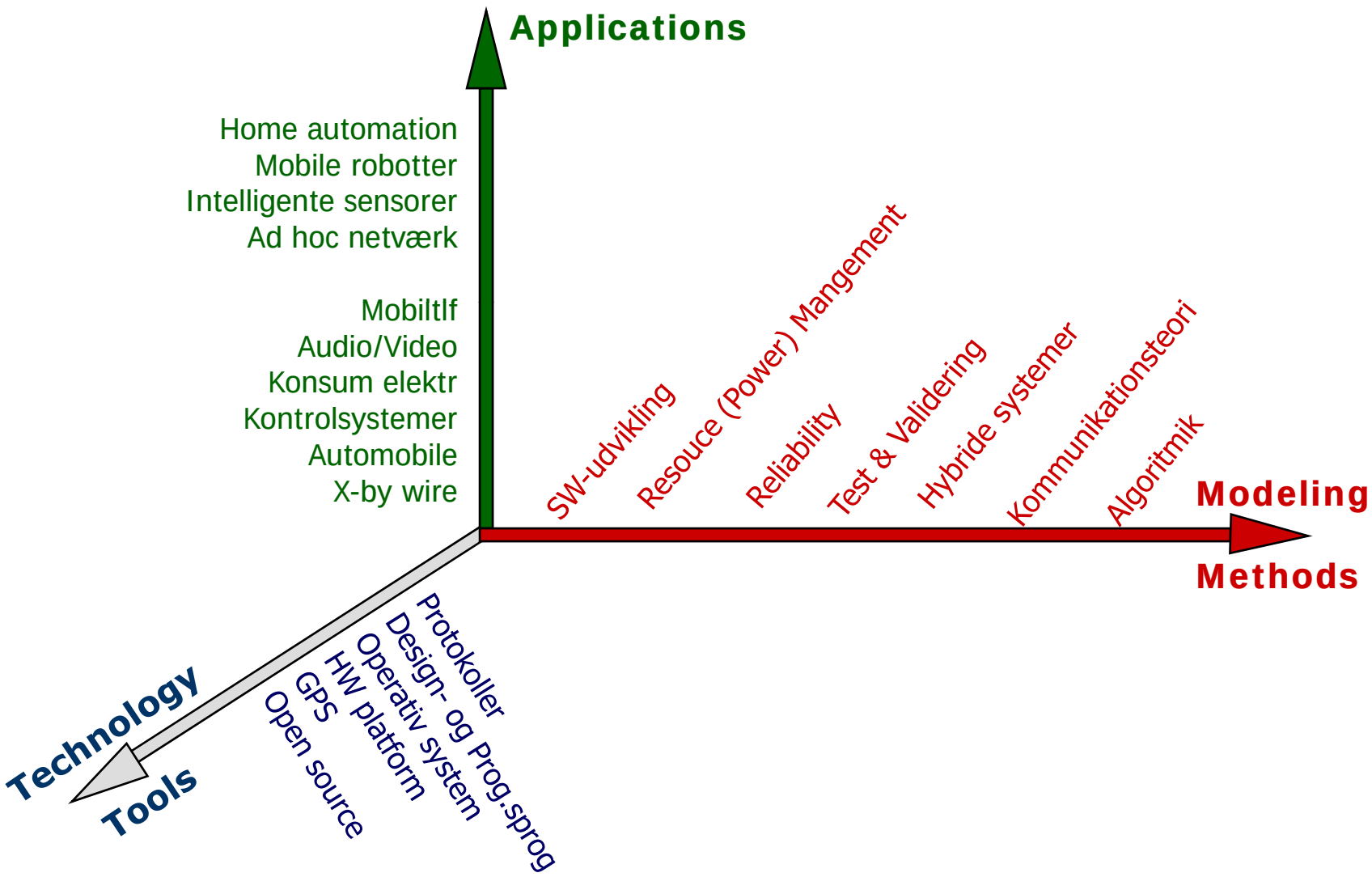


Partners

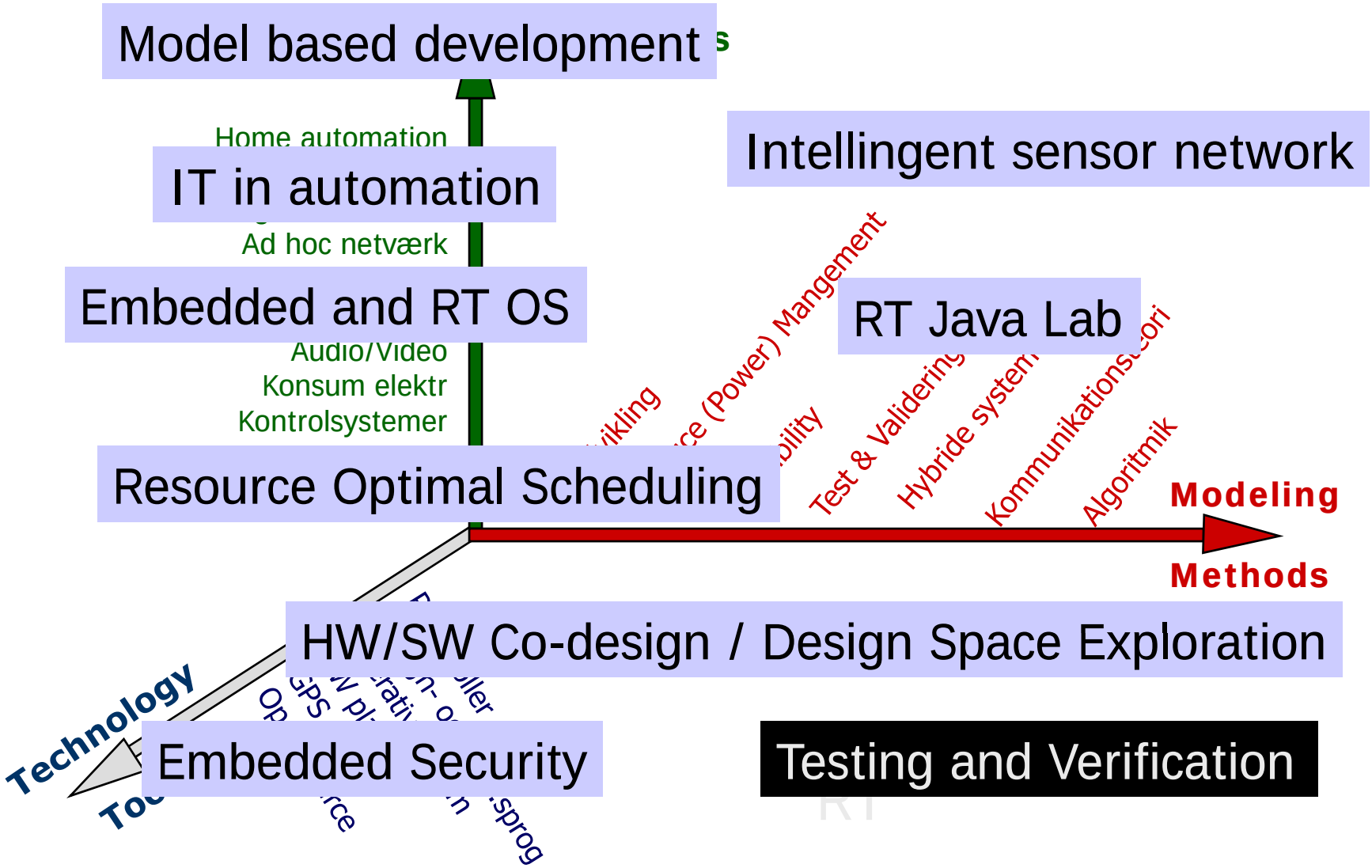
- Aeromark
- Analog Devices
- Blip Systems
- Danfoss
- Ericsson Telebit
- ETI
- Exhausto
- FOSS
- GateHouse
- Grundfos
- IAR Systems
- MAN B&W
- Novo Nordisk
- Motorola
- Panasonic
- RTX Telecom
- S-Card
- Simrad
- Skov
- SpaceCom
- TK Systemtest
- TDC Totalløsninger
- Aalborg Industries
- LandsCentret



Focus Areas

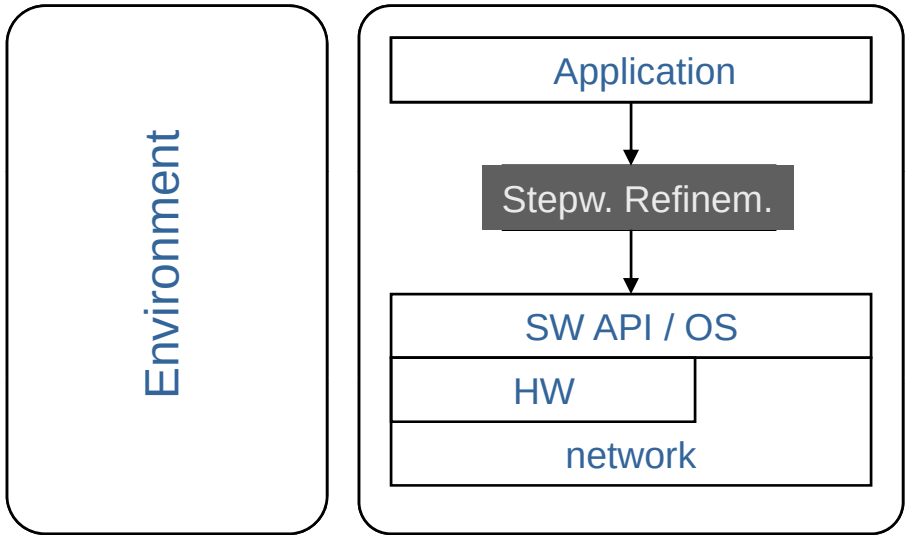

Applications
Home automation
Mobile robotter
Intelligente sensorer
Ad hoc netværk
Mobiltlf
Audio/Video
Konsum elektr
Kontrolsystemer
Automobile
X-by wire
SW-udvikling
Resouce (Power) Mangement
Reliability
Test & Validering
Hybride systemer
Kommunikationsteori
Algoritmik
Modeling
Methods
Technology
Tools
Protokoller
Design- og Prog.sprog
Operativ system
HW platform
GPS
Open source

Focus Areas


Model based development
Home automation
IT in automation
Ad hoc netværk
Embedded and RT OS
Audio/Video
Konsum elektr
Kontrolsystemer
Resource Optimal Scheduling
Intellingent sensor network
RT Java Lab
Test & Validering
Hybride system
Kommunikationsteori
Algoritmik
Modeling
Methods
HW/SW Co-design / Design Space Exploration
Technology
Embedded Security
Testing and Verification

DaNES

Danish Network for Intelligent Embedded Systems



Funded by
Danish Advanced Technology Foundation
Budget 9 MEuro / 4 years

DaNES Challenges

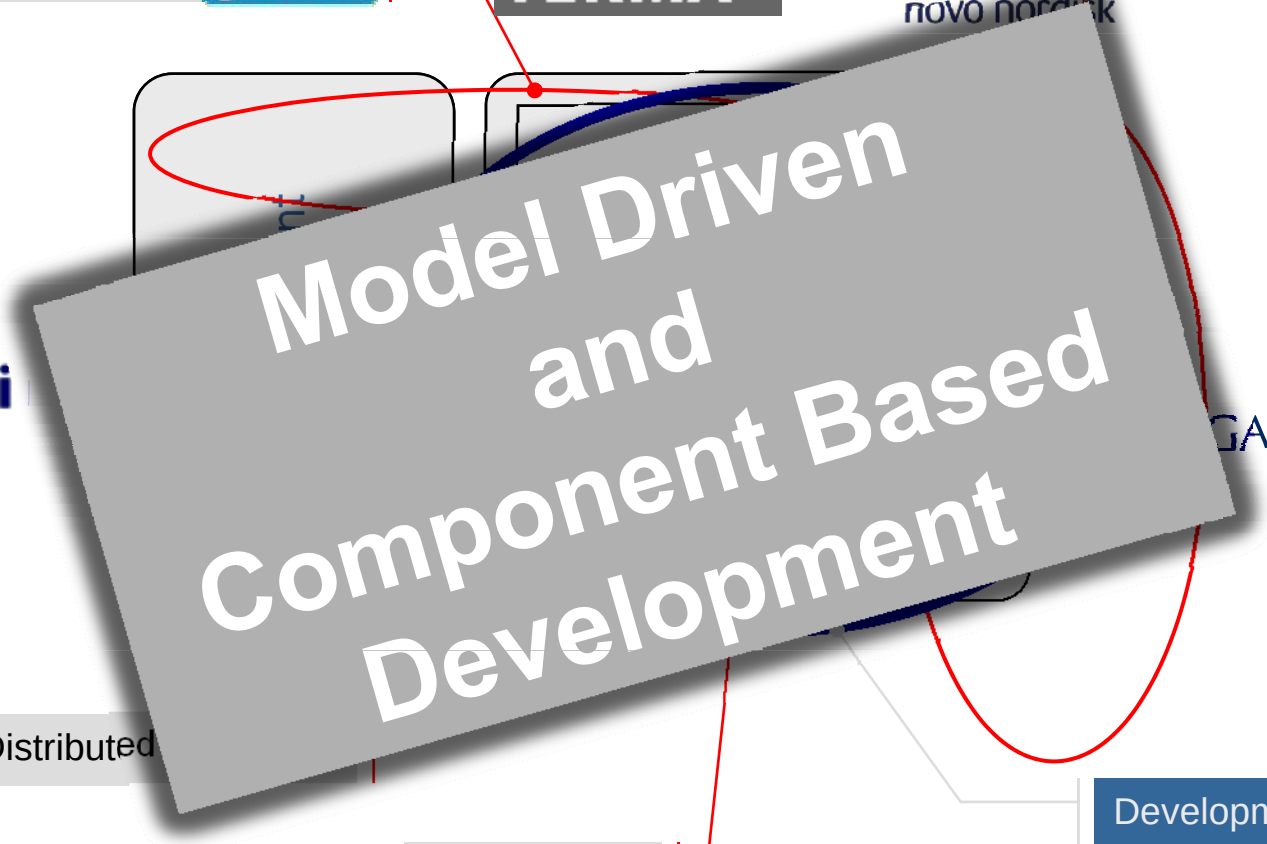
Danish Network for Intelligent Embedded Systems

Selfdiagnosic & -repair

SKOV

TERMA

Test & Verificaiton



Model Driven and Component Based Development

mci

CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

GATEHOUSE

ice power

Embedded & Distributed

Informatik og Matematisk Modellering

Platform

Development

PAJ SYSTEMTEKNIK

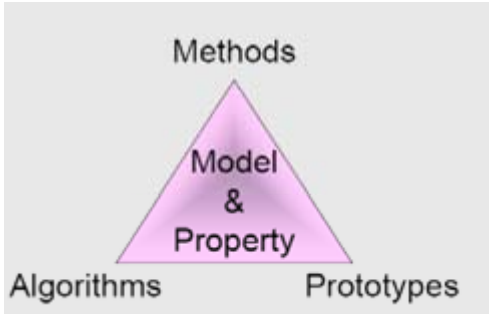
IO TECHNOLOGIES

MT LAB Modelling of Information Technology

Villum-Kahn Rasmussen Center of Excellence

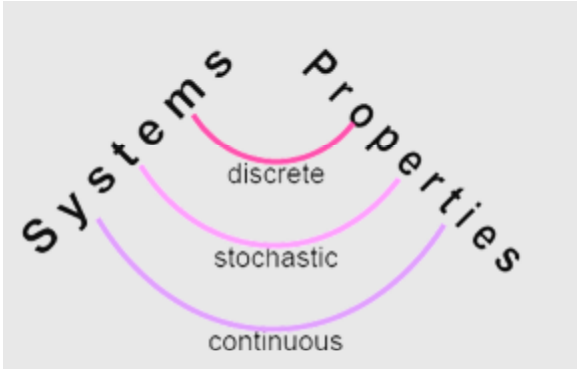
Opening November 19, 2008

6.5 MEUR



Static Analysis **Model Checking**

Embedded Systems



Service Oriented Architectures

IMM/DTU, CISS/AAU, ITU

Director Flemming Nielson

Co-Director Kim G Larsen

IDEA4CPS

Danmarks Grundforskningsfond
Danish National Research Foundation

Foundations for Cyber-Physical Systems

From Computer Science to Cyber Physical Systems

MT-LAB Meeting 11.1.2011 Kim Guldstrand Larsen [6] IDEA4 CPS

Topics & Task Overview


Jan Madsen


ZHU Hiabioa

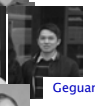

Flemming Nielson

Kim G Larsen


Hanne R. Nielson


ZHANG Jian


Arne Skou


Geguang Pu


Anders P Ravn

- Specification and Modeling
- Validation and Analysis
- Compositionality versus Global Features
- Cross-Level Preservation
- Mini Cases
- Prototype Tools

MT-LAB Meeting 11.1.2011 Kim Guldstrand Larsen [17] IDEA4 CPS

IDEA4CPS

Danmarks Grundforskningsfond
Danish National Research Foundation

Foundations for Cyber-Physical Systems

From Computer Science to Cyber Physical Systems

MT-LAB Meeting 11.1.2011

N

l Modeling

nalysis

versus Global

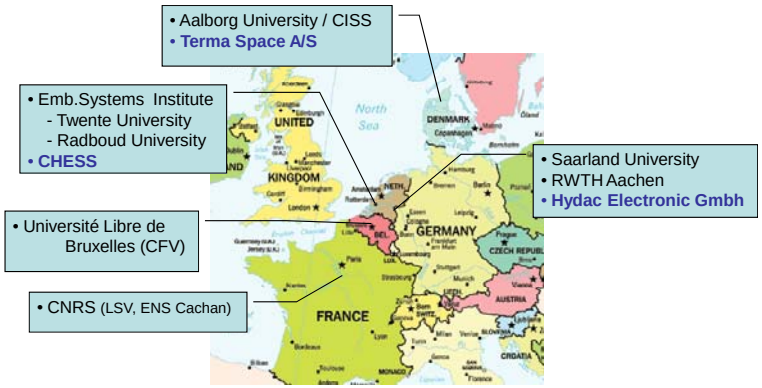
ervation

nd Larsen [17]

IDEA4 CPS

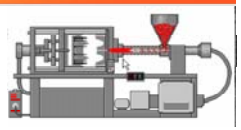
Quasimodo

Partners



WP5 Case Studies

- **Accumulator Charge Controller** (HYDAC)
  - Design of robust and optimal control for hydralic pump
  - (UPPAAL Tiga, Phaver, Simulink)



- **Wireless Sensor Network** (CHESS)
  - Analysis of gMAC protocol (UPPAAL) Potential of time synchronization failing Identified, demonstrated and partially corrected (UPPAAL)
  - Testing (jTorX, TorXakis, TRON)
  - Trade-off between energy comsumption and collision rates (MODEST)

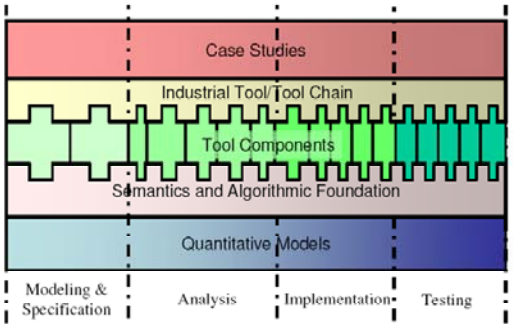
- **Control Software for satellites Hershel and Planck** (TERMA)
  - Schedulability and WCET analysis (UPPAAL)

Workplan Strategy

"Well documented API's & exchange formats"

xLTS

Timed, hybrid, stochastic, priced, .. automata



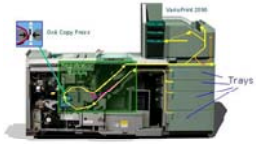
WP5

WP1 WP2 WP3 WP4

WP5 Additional Case Studies

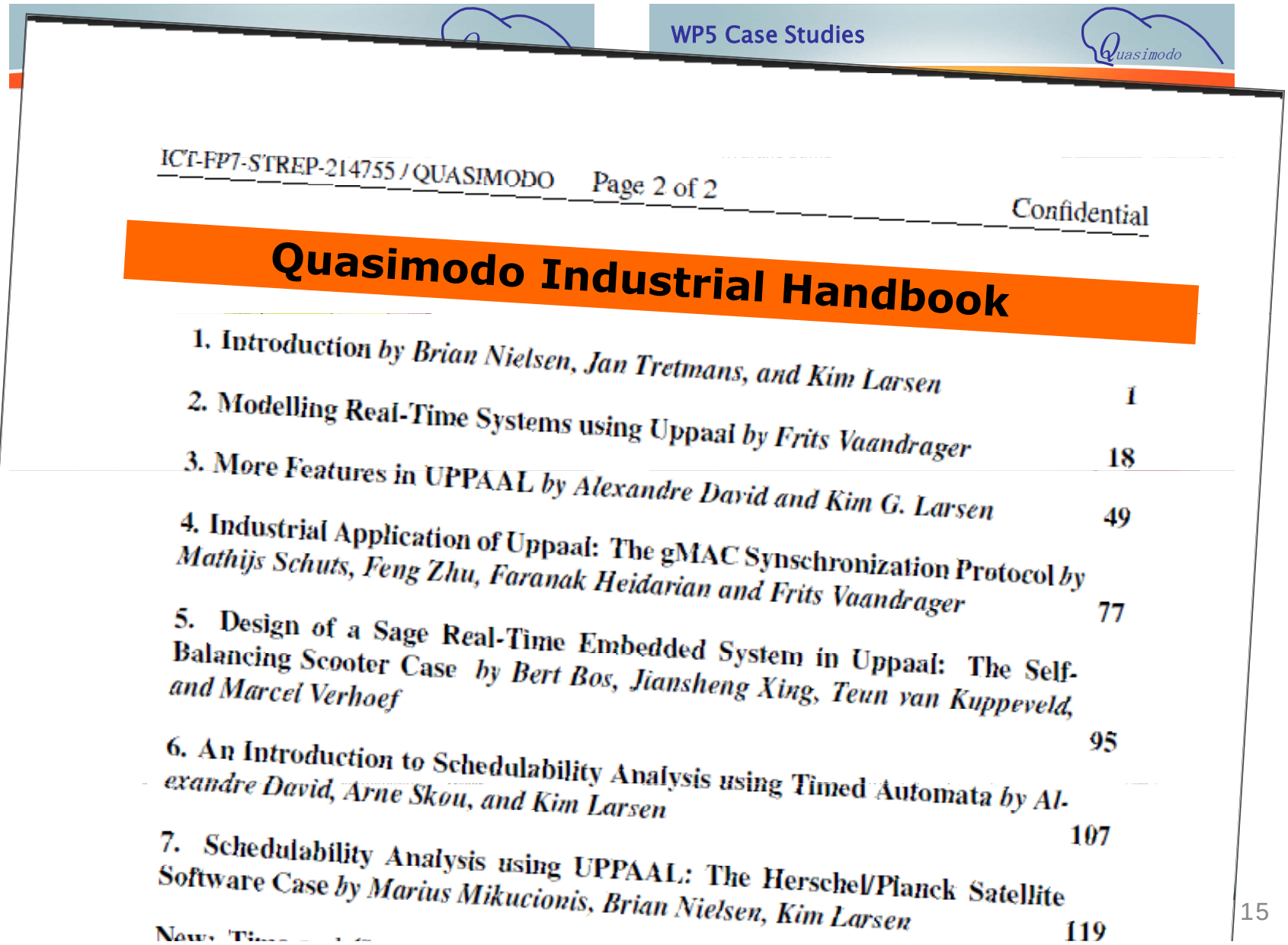
- **Self-Balancing Scooter** (CHESS)
  - Highlevel control-modes modeled by engineers (UPPAAL)
  - Schedulability (UPPAAL)

- **Adaptive scheduling of data paths** (OCE)
  - Synthesis of optimal data path (CORA)



- **Rapid Input-Output Packet Switch** (ASML)
  - Simulation and verificatoin ofworst-case latencies (POOSL, UPPAAL)

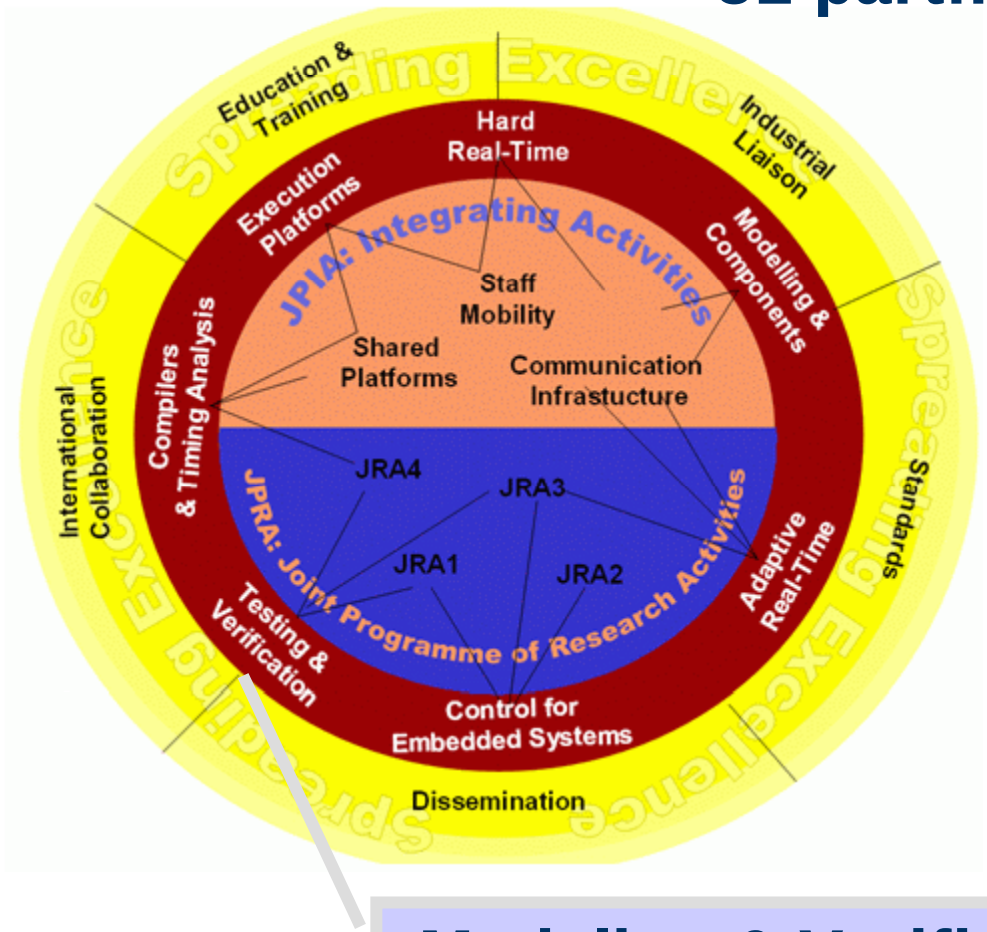
Quasimodo


WP5 Case Studies
Quasimodo
ICT-FP7-STREP-214755 / QUASIMODO Page 2 of 2 Confidential
Quasimodo Industrial Handbook
1. Introduction by Brian Nielsen, Jan Tretmans, and Kim Larsen 1
2. Modelling Real-Time Systems using Uppaal by Frits Vaandrager 18
3. More Features in UPPAAL by Alexandre David and Kim G. Larsen 49
4. Industrial Application of Uppaal: The gMAC Synschronization Protocol by Mathijs Schuts, Feng Zhu, Faranak Heidarian and Frits Vaandrager 77
5. Design of a Sage Real-Time Embedded System in Uppaal: The Self-Balancing Scooter Case by Bert Bos, Jiansheng Xing, Teun van Kuppeveld, and Marcel Verhoef 95
6. An Introduction to Schedulability Analysis using Timed Automata by Alexandre David, Arne Skou, and Kim Larsen 107
7. Schedulability Analysis using UPPAAL: The Herschel/Planck Satellite Software Case by Marius Mikucionis, Brian Nielsen, Kim Larsen 119

European Network of Excellence

32 partners

Information Society Technologies



ARTEMIS

Joseph Sifakis

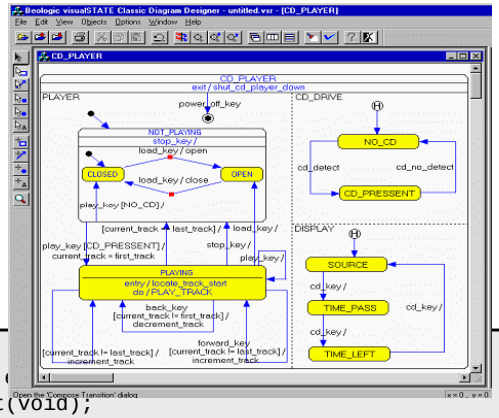
Co-winner of Turing Award 2007

ARTIST Director

Modeling & Verification
CISS coordinator

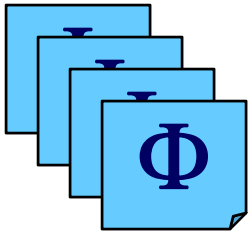
Verification and Testing

Model




```
/* Wait for
void OS_Wait(void);

/* Operating system visualSTATE process. Mimics a OS process for a
 * visualSTATE system. In this implementation this is the mainloop
 * interfacing to the visualSTATE basic API. */
void OS_VS_Process(void);

/* Define completion code variable. */
unsigned char cc;

void HandleError(unsigned char ccArg)
{
  printf("Error code %c detected, exiting application.\n", ccArg);
  exit(ccArg);
}


/* In d-241 we only use the OS_Wait call. It is used to simulate a
 * system. It purpose is to generate events. How this is done is up to
 * you.
 */
void OS_Wait(void)
{
  /*  Ignore the parameters; just retrieve events from the keyboard and
   *  put them into the queue. When EVENT UNDEFINED is read from the
   *  keyboard, return to the calling process. */
  SEM_EVENT_TYPE event;
  int num;
```

Code

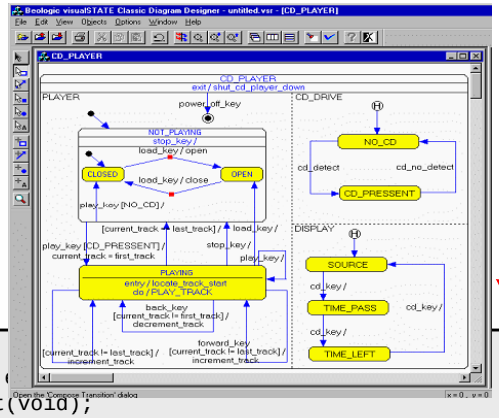


Req

Running System

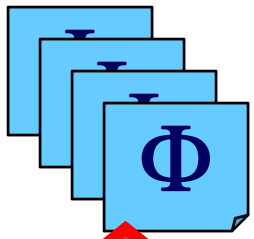
Verification and Testing

Model




```
/* Wait for
void OS_Wait(void);

/* Operating system visualSTATE process. Mimics a OS process for
 * visualSTATE system. In this implementation this is the mainloop
 * interfacing to the visualSTATE basic API. */
void OS_VS_Process(void);

/* Define completion code variable. */
unsigned char cc;

void HandleError(unsigned char ccArg)
{
  printf("Error code %c detected, exiting application.\n", ccArg);
  exit(ccArg);
}


/* In d-241 we only use the OS_Wait call. It is used to simulate a
 * system. It purpose is to generate events. How this is done is up to
 * you.
 */
void OS_Wait(void)
{
  /*  Ignore the parameters; just retrieve events from the keyboard and
   *  put them into the queue. When EVENT UNDEFINED is read from the
   *  keyboard, return to the calling process. */
  SEM_EVENT_TYPE event;
  int num;
```

Code

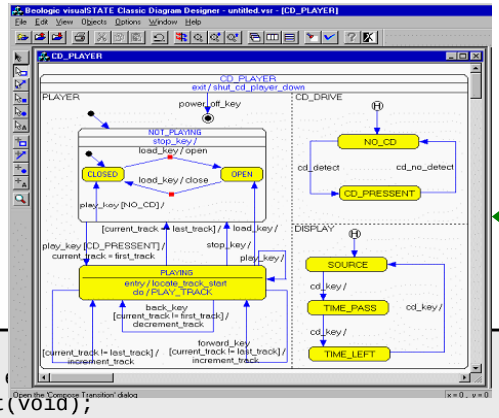


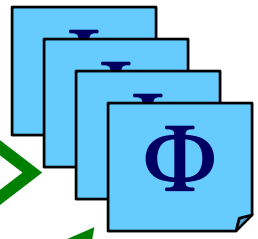
Req

Running System

Verification and Testing

Model





Req

```
/* Wait for
void OS_Wait(void);

/* Operating system visualSTATE process. Mimics a OS process for a
 * visualSTATE system. In this implementation this is the mainloop
 * interfacing to the visualSTATE basic API. */
void OS_VS_Process(void);

/* Define completion code variable. */
unsigned char cc;

void HandleError(unsigned char ccArg)
{
  printf("Error code %c detected, exiting application.\n", ccArg);
  exit(ccArg);
}


/* In d-241 we only use the OS_Wait call. It is used to simul
 * system. It purpose is to generate events. How this is don      to
 * you.
 */
void OS_Wait(void)
{
  /*  Ignore the parameters; just retrieve events from the keyboard and
   *  put them into the queue. When EVENT UNDEFINED is read from the
   *  keyboard, return to the calling process. */
  SEM_EVENT_TYPE event;
  int num;
```

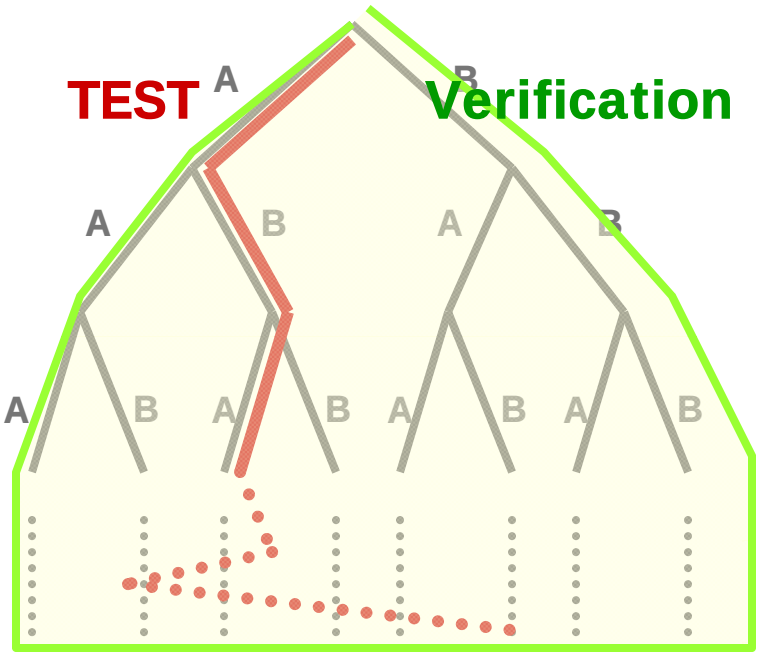
Code

Running System

Test versus Verification

E F E E G H … H A

T1 T3 T5 T1 … T4 T3



2^n sequences of length n

Deadlock identified using **Verification**
After sequence of 2000 telegrams / < 1min

UPPAAL

Why Verification and Testing

- 30-40% of production time is currently spend on elaborate, ad-hoc testing:
  - Errors expensive and difficult to fix!
  - The potential of existing/improved testing methods and tools is enormous!
  - Time-to-market may be shortened considerable by verification and performance analyses of early designs!

```
*** STOP: 0x0000000A (0x802aa502,0x00000002,0x00000000,0xFA84001C)
IRQL_NOT_LESS_OR_EQUAL*** Address fa84001c has base at fa840000 - i8042prt.SYS

CPUID: GenuineIntel 5.2.c irql:1f    SYSVER 0xF0000565

Dll Base  Date Stamp - Name                 Dll Base  Date Stamp - Name
80100000  2be154c9   - ntoskrnl.exe         80400000  2bc153b0   - hal.dll
80200000  2bd49628   - ncrc710.sys          8025c000  2bd49688   - SCSIPORT.SYS
80267000  2bd49683   - scsidisk.sys         802a6000  2bd496b9   - Fastfat.sys
fa800000  2bd49666   - Floppy.SYS           fa810000  2bd496db   - Npfs_Rec.SYS
fa820000  2bd49676   - Null.SYS             fa830000  2bd4965a   - Beep.SYS
fa840000  2bdaab00   - i8042prt.SYS         fa850000  2bd5a020   - SERMOUSE.SYS
fa860000  2bd4966f   - kbdclass.SYS         fa870000  2bd49671   - MOUCLASS.SYS
fa880000  2bd9c0be   - Videoprt.SYS         fa890000  2bd49638   - NCR77C22.SYS
fa8a0000  2bd4a4ce   - Vga.SYS              fa8b0000  2bd496d0   - Msfs.SYS
fa8c0000  2bd496c3   - Npfs.SYS             fa8e0000  2bd496c9   - Ntfs.SYS
fa940000  2bd496df   - NDIS.SYS             fa930000  2bd49707   - wdlan.sys
fa970000  2bd49712   - TDI.SYS              fa950000  2bd5a7fb   - nbf.sys
fa980000  2bd72106   - atrcamc.sys          fa9b0000  2bd4975f   - ubnb.sys
fa9c0000  2bd5bfd7   - mcsxns.sys           fa9d0000  2bd4971d   - netbios.sys
fa9e0000  2bd49678   - Parallel.sys         fa9f0000  2bd4969f   - serial.SYS
faa00000  2bd49739   - mup.sys              faa40000  2bd4971f   - SMBTRSUP.SYS
faa10000  2bd6f2a2   - srv.sys              faa50000  2bd4971a   - afd.sys
faa60000  2bd6fd80   - rdr.sys              faaa0000  2bd49735   - bowser.sys

Address  dword dump   Build [1381]                              - Name
fe9cdaec fa84003c fa84003c 00000000 00000000 80149905            - i8042prt.SYS
fe9cdaf8 8025dfe0 8025dfe0 ff8e6b8c 80129c2c ff8e6b94            - SCSIPORT.SYS
fe9cdb10 8013e53a 8013e53a ff8e6b94 00000000 ff8e6b94            - ntoskrnl.exe
fe9cdb18 8010a373 8010a373 ff8e6df4 ff8e6f60 ff8e6c58            - ntoskrnl.exe
fe9cdb38 80105683 80105683 ff8e6f60 ff8e6c3c 8015ac7e            - ntoskrnl.exe
fe9cdb44 80104722 80104722 ff8e6df4 ff8e6f60 ff8e6c58            - ntoskrnl.exe
fe9cdb4c 8012034c 8012034c 00000000 80088000 80106fc0            - ntoskrnl.exe
```

Why Verification and Testing

- **IMPORTANCE for EMBEDDED SYSTEMS**
  - Often safety critical
  - Often economical critical
  - Hard to patch

- **CHALLENGES for EMBEDDED SYSTEMS**
  - Correctness of embedded systems depend crucially on use of

    resources

    e.g. real-time, memory, bandwidth, energy.
  - Need for

    quantitative models

Spectacular software bugs
Ariane 5

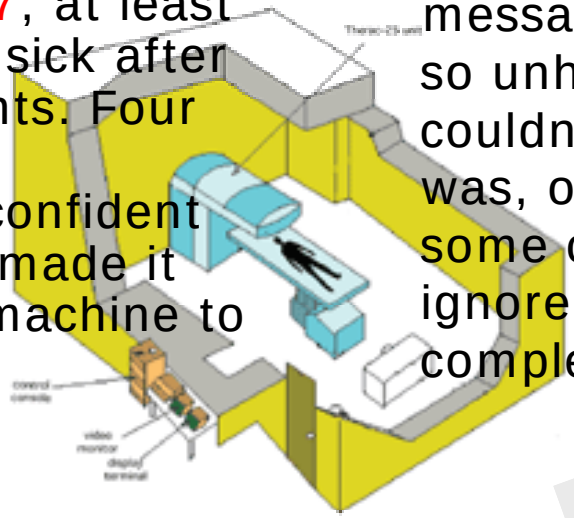
- The first Ariane 5 rocket was launched in June, 1996. It used software developed for the successful Ariane 4. The rocket carried two computers, providing a backup in case one computer failed during launch. Forty seconds into its maiden flight, the rocket veered off course and exploded. The rocket, along with $500 million worth of satellites, was destroyed.

- Ariane 5 was a much more powerful rocket and generated forces that were larger than the computer could handle. Shortly after launch, it received an input value that was too large. The main and backup computers shut down, causing the rocket to veer off course.

Spectacular software bugs
Therac 25

Safety Critical

- The Therac-25 radiation therapy machine was a medical device that used beams of electrons or photons to kill cancer cells. Between 1985-1987, at least six people got very sick after Therac-25 treatments. Four of them died. The manufacturer was confident that their software made it impossible for the machine to harm patients.



- The Therac-25 was withdrawn from use after it was determined that it could deliver fatal overdoses under certain conditions. The software would shut down the machine before delivering an overdose, but the error messages it displayed were so unhelpful that operators couldn't tell what the error was, or how serious it was. In some cases, operators ignored the message completely.

"H-tilt"

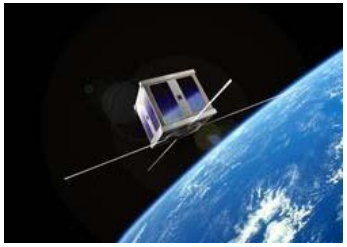
"Malfunction 54"

IEEE Computer, Vol. 26, No. 7, July 1993, pp. 18-41

Spectacular Software Bugs
…. continued

- INTEL Pentium II floating-point division
  470 Mill US $
- Baggage handling system, Denver
  1.1 Mill US $/day for 9 months
- Mars Pathfinder
- …….

ES are Pervasive



- **Characteristica**:
  - **Dedicated function**
  - **Complex environment**
  - **SW / HW / Mechanics**
  - **Autonomous**
  - **Ressource constrained**
    - **: Energy**
    - **: Bandwidth**
    - **: Memory**
    - **: …**
  - **Timing constraints**

ES are often Safety Critical

- 300 horse power
- 100 processors

- How to achieve ES that are:
  - correct
  - predicable
  - dependable
  - fault tolerant
  - ressource minial
  - cheap

- Model-Based Development

A simple program

```
int x=100;

Process INC
        do
        :: x<200 --> x:=x+1
        od

Process DEC
        do
        :: x>0 --> x:=x-1
        od

Process RESET
        do
        :: x=200 --> x:=0
        od

( INC || DEC || RESET )
```

Which values may x take ?

Questions/Properties:

```
E<>(x>100)
E<>(x>200)
A[](x<=200)
E<>(x<0)
A[](x>=0)
```

Possibly

Always

Another simple program

```
int x=0;

Process P
   do
      x:=x+1
   10 times

( P || P )
```

What are the possible final values of x ?

```
int x=0;

Process P
int r
   do
      r:=x; r++; x:=r
   10 times

( P || P )
```

Atomic stm

Yet another simple program

```
int x=1;

Process P
   do
      x:=x+x
   forever

( P || P )
```

What are the possible values that x may posses during execution?

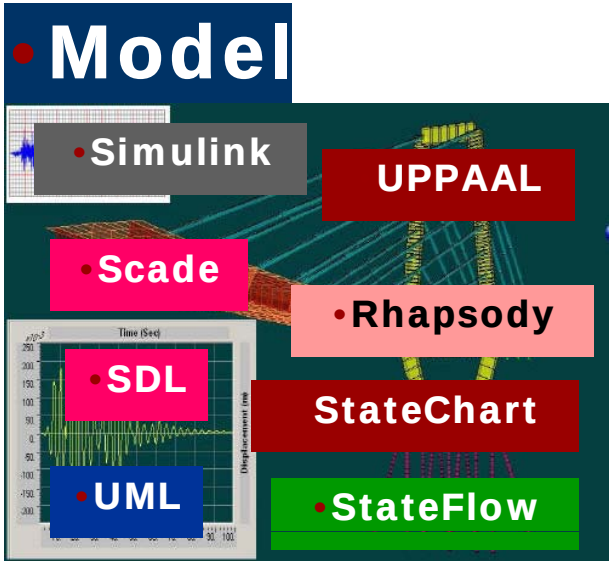
```
int x=1;

Process P
int r
   do
      r:=x; r:=x+r; x:=r
   forever

( P || P )
```

Atomic stm

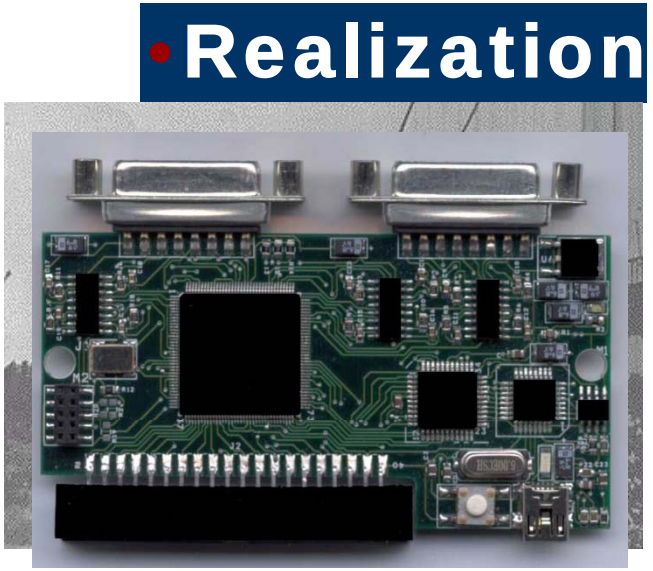
Model-based Approach

Models

- A model is a simplified representation of the real world.
- User gains confidence in the adequacy and validity of a proposed system.
- Models selected aspects. Removes irrelevant details.
- Early design exploration.

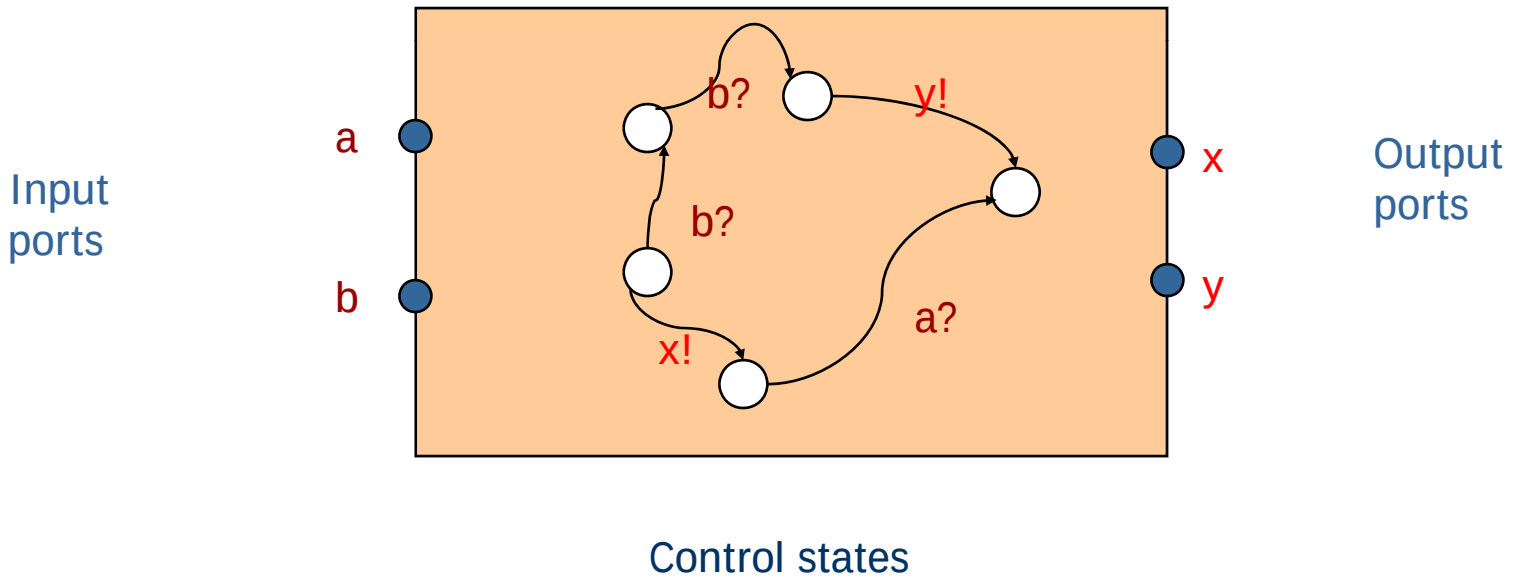


•"implements?"

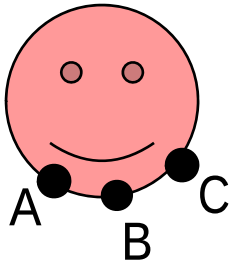


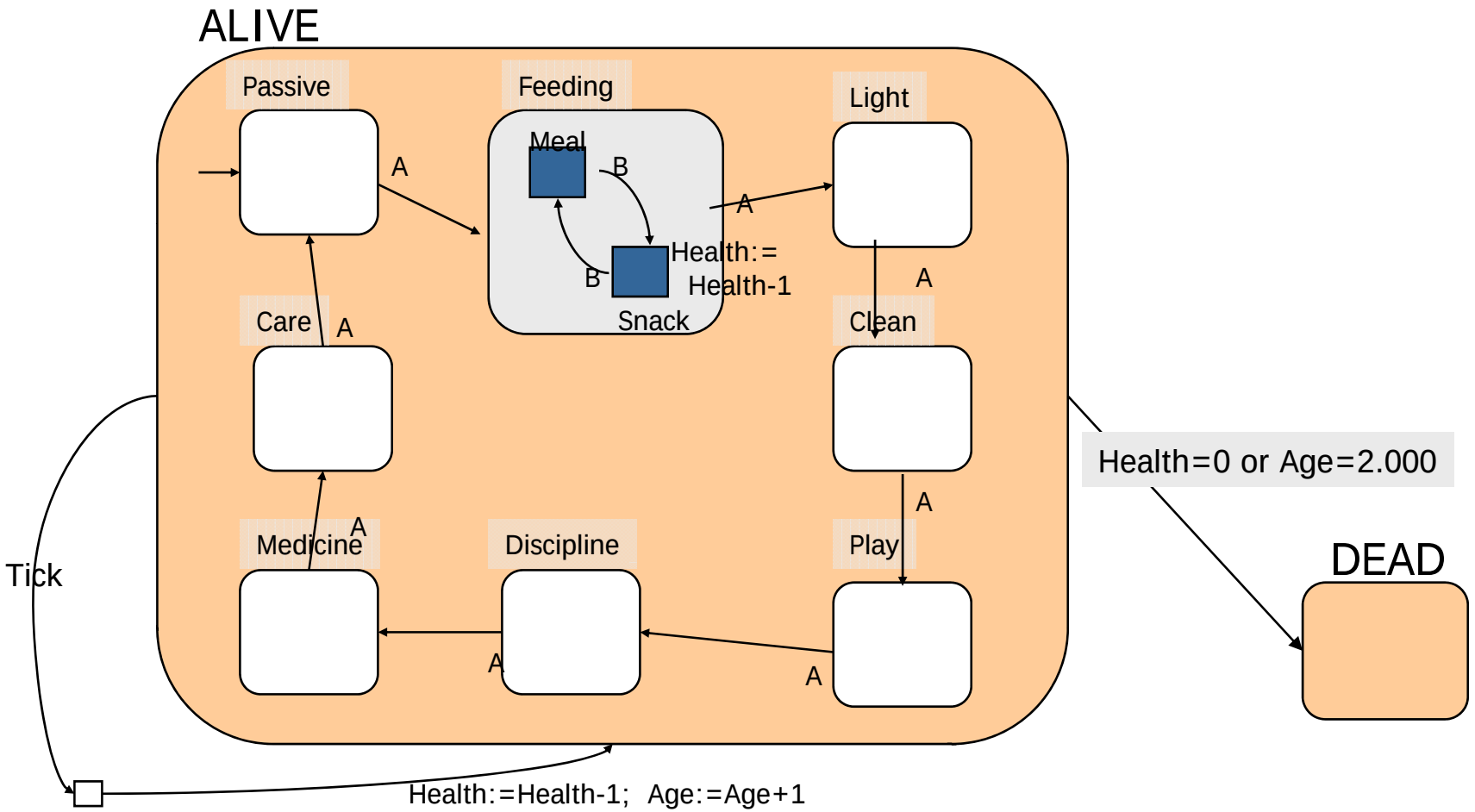
How?

Unified Model = State Machine!

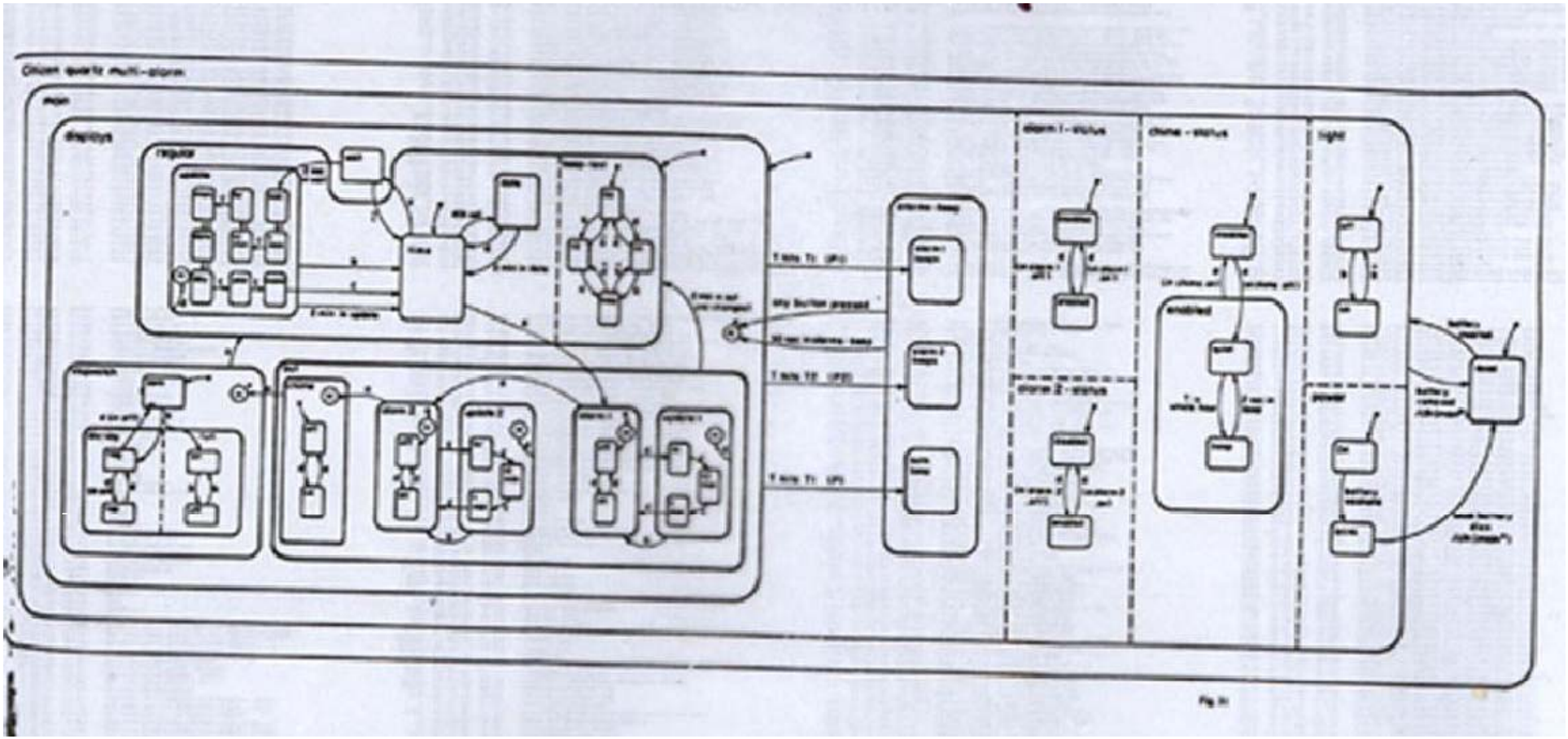


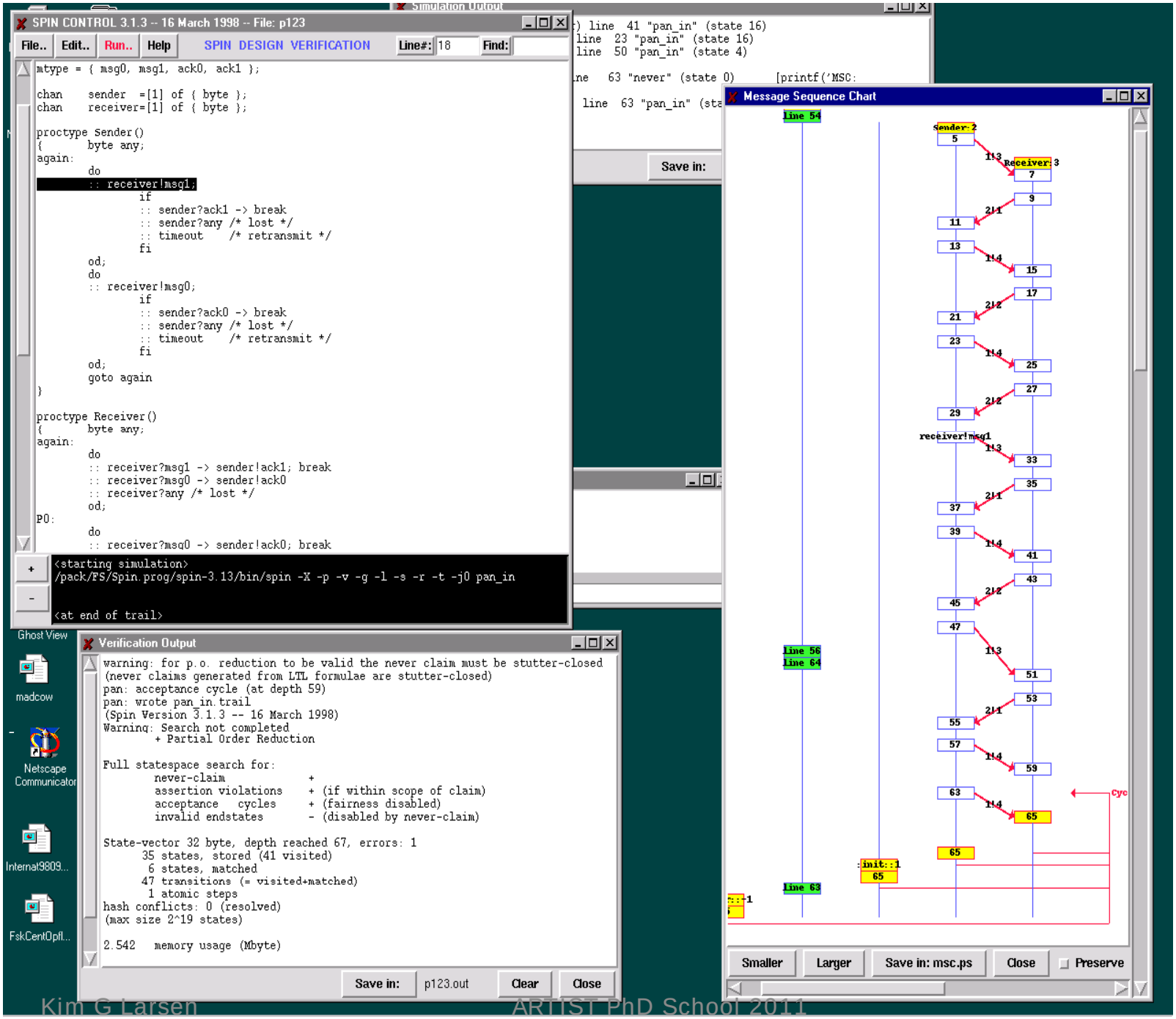
Tamagotchi


A
B
C


ALIVE
Passive
Feeding
Light
Meal
A
B
A
B
Health:=
Health-1
Snack
A
Care
A
Clean
A
A
Medicine
A
Discipline
Play
Tick
A
A
Health=0 or Age=2.000
DEAD
Health:=Health-1; Age:=Age+1

Digital Watch – UML Statechart





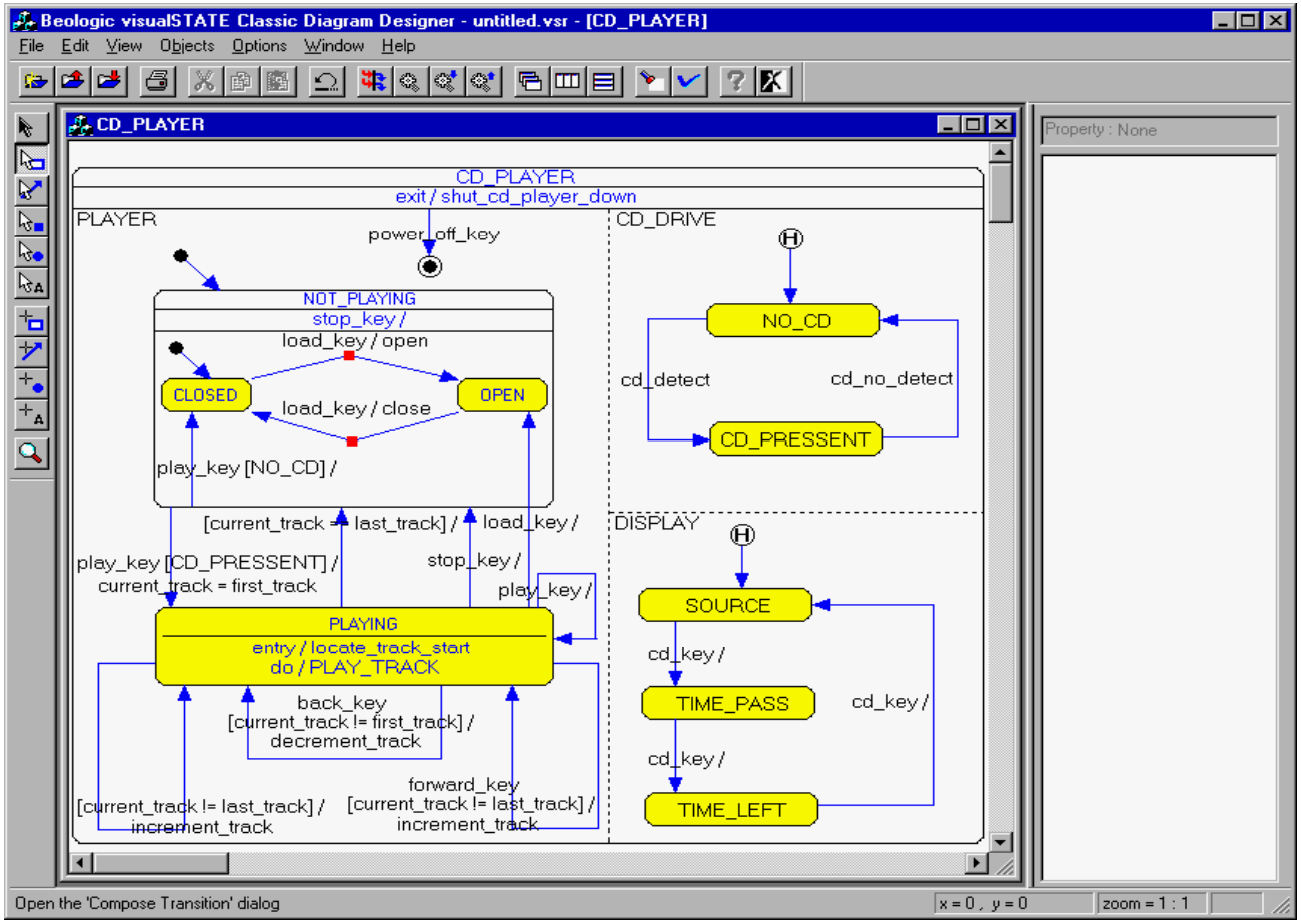
SPIN, Gerald Holzmann AT&T

visualSTATE

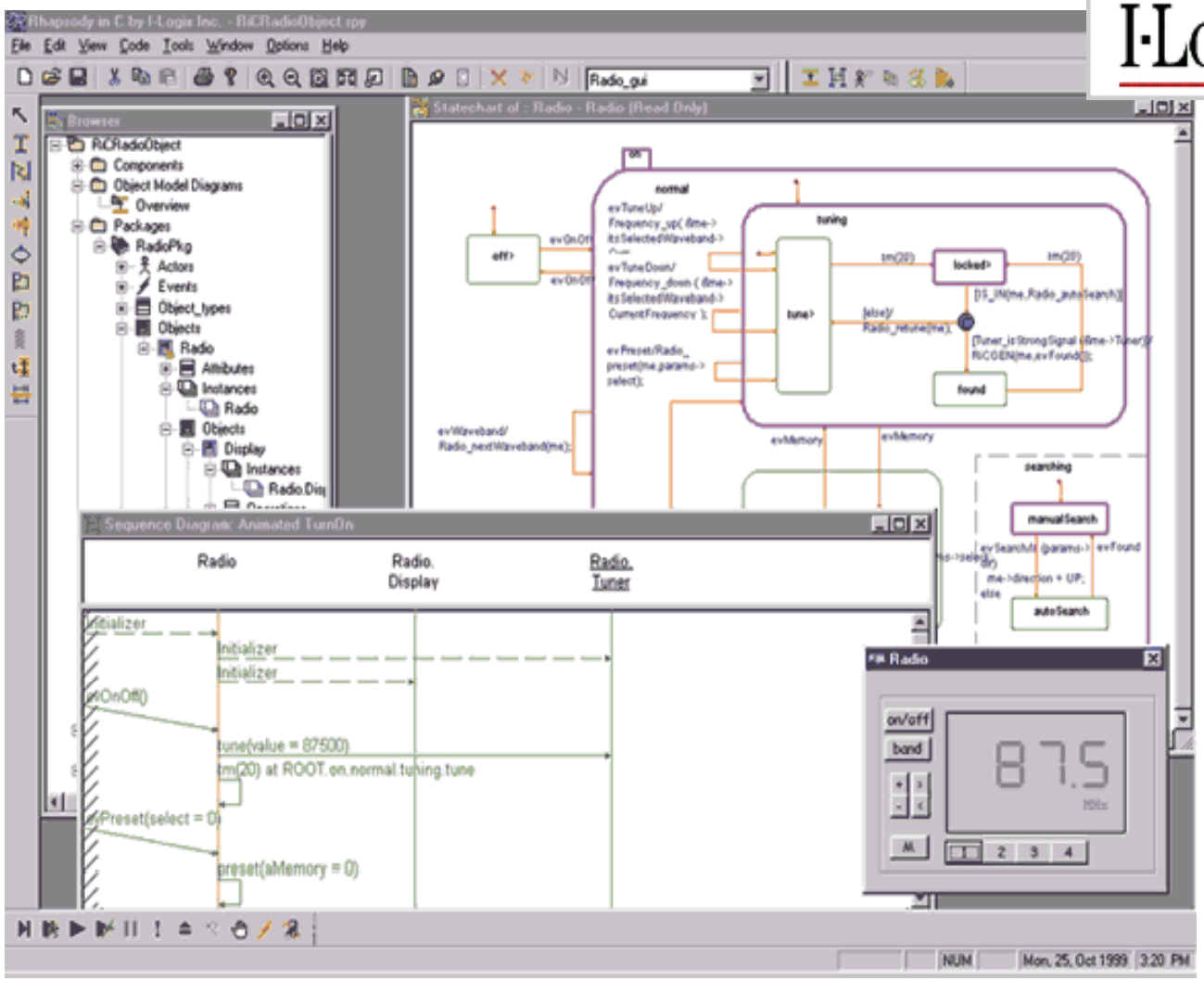
VVS

w Baan Visualstate, DTU (CIT project)

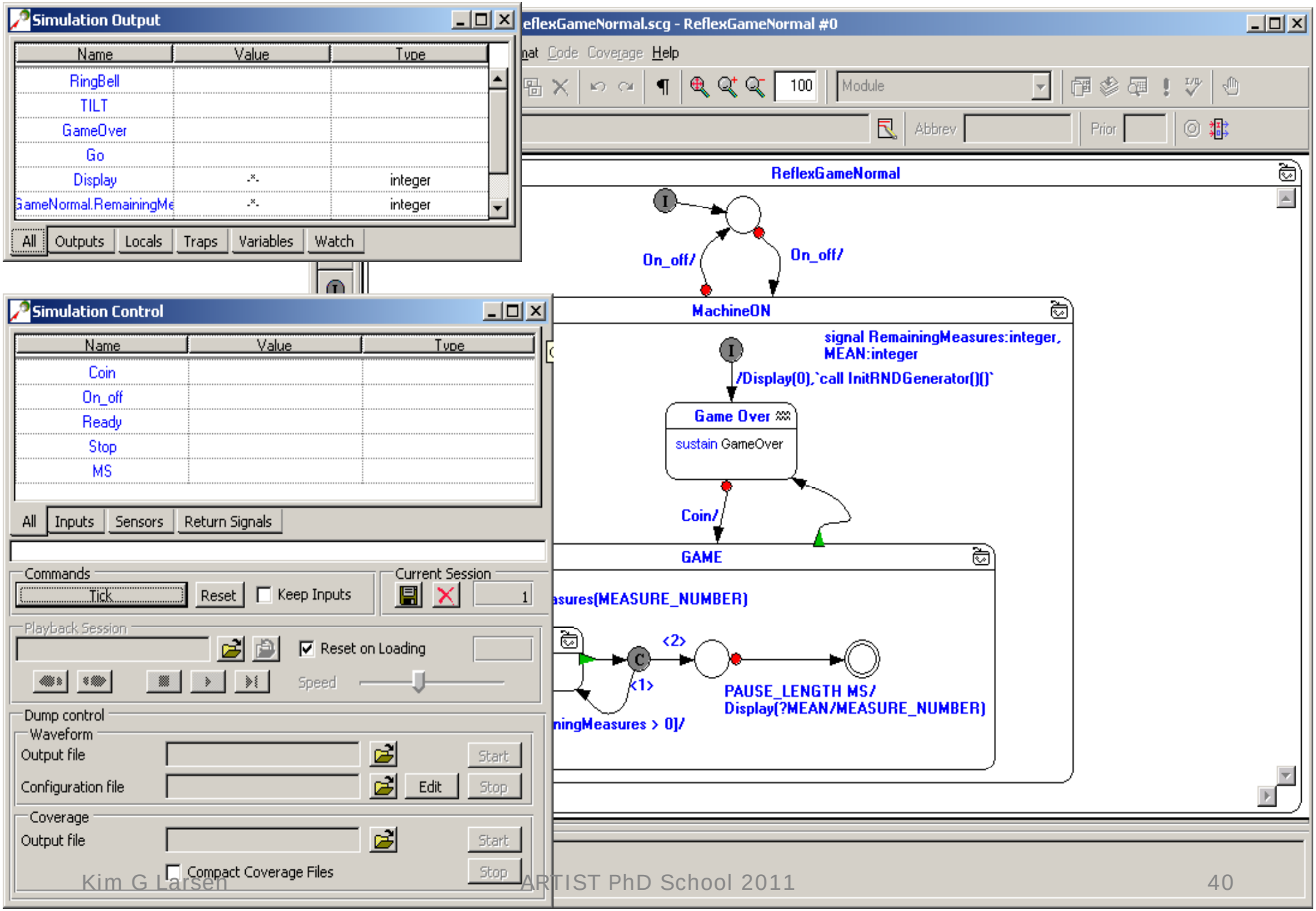
- Hierarchical state systems
- Flat state systems
- Multiple and inter-related state machines
- Supports UML notation
- Device driver access

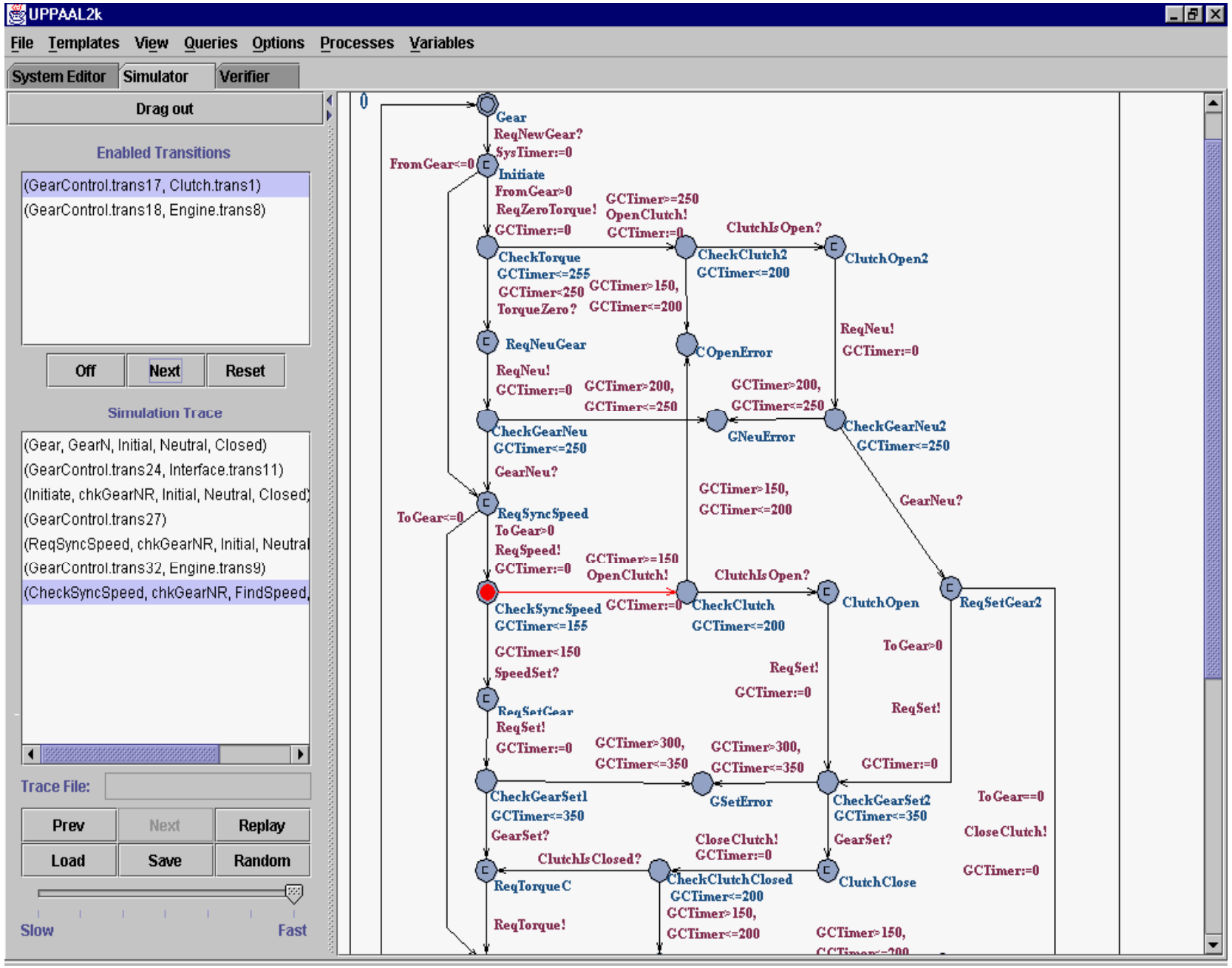


Rhapsody


Rhapsody in C by I-Logix Inc. - RiCRadioObject.rpy
File Edit View Code Tools Window Options Help
Radio_gui
Browser
RiCRadioObject
Components
Object Model Diagrams
Overview
Packages
RadioPkg
Actors
Events
Object_types
Objects
Radio
Attributes
Instances
Radio
Objects
Display
Instances
Statechart of : Radio - Radio (Read Only)
off
normal
tuning
tune
locked
found
searching
manualSearch
autoSearch
Sequence Diagram: Animated TurnOn
Radio
Radio.Display
Radio.Tuner
Initializer
Initializer
Initializer
evOnOff()
tune(value = 87500)
tm(20) at ROOT.on.normal.tuning.tune
evPreset(select = 0)
preset(aMemory = 0)
Radio
on/off
band
87.5
1
2
3
4
NUM
Mon, 25, Oct 1999
3:20 PM

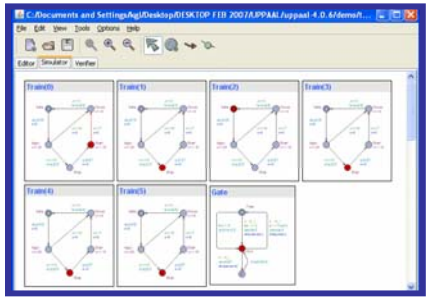
ESTEREL


Simulation Output
Name
Value
Type
RingBell
TILT
GameOver
Go
Display
integer
integer
All
Outputs
Locals
Traps
Variables
Watch
Simulation Control
Coin
On_off
Ready
Stop
MS
Inputs
Sensors
Return Signals
Commands
Tick
Reset
Keep Inputs
Current Session
Playback Session
Reset on Loading
Speed
Dump control
Waveform
Output file
Configuration file
Start
Edit
Stop
Coverage
Compact Coverage Files
eflexGameNormal.scg - ReflexGameNormal #0
Code
Coverage
Help
Module
Abbrev
Prior
ReflexGameNormal
On_off/
On_off/
MachineON
signal RemainingMeasures:integer,
MEAN:integer
/Display(0),`call InitRNDGenerator()()`
Game Over
sustain GameOver
Coin/
GAME
asures(MEASURE_NUMBER)
<2>
<1>
PAUSE_LENGTH MS/
Display(?MEAN/MEASURE_NUMBER)
ningMeasures > 0]/


UPPAAL2k
File Templates View Queries Options Processes Variables
System Editor Simulator Verifier
Drag out
Enabled Transitions
(GearControl.trans17, Clutch.trans1)
(GearControl.trans18, Engine.trans8)
Off Next Reset
Simulation Trace
(Gear, GearN, Initial, Neutral, Closed)
(GearControl.trans24, Interface.trans11)
(Initiate, chkGearNR, Initial, Neutral, Closed)
(GearControl.trans27)
(ReqSyncSpeed, chkGearNR, Initial, Neutral
(GearControl.trans32, Engine.trans9)
(CheckSyncSpeed, chkGearNR, FindSpeed,
Trace File:
Prev Next Replay
Load Save Random
Slow Fast
Gear
ReqNewGear?
SysTimer:=0
FromGear<=0
Initiate
FromGear>0
ReqZeroTorque!
GCTimer:=0
GCTimer>=250
OpenClutch!
GCTimer:=0
ClutchIsOpen?
CheckTorque
GCTimer<=255
GCTimer<250
TorqueZero?
CheckClutch2
GCTimer<=200
ClutchOpen2
GCTimer>150,
GCTimer<=200
ReqNeuGear
COpenError
ReqNeu!
GCTimer:=0
ReqNeu!
GCTimer:=0
GCTimer>200,
GCTimer<=250
GCTimer>200,
GCTimer<=250
CheckGearNeu
GCTimer<=250
GNeuError
CheckGearNeu2
GCTimer<=250
GearNeu?
GearNeu?
GCTimer>150,
GCTimer<=200
ToGear<=0
ReqSyncSpeed
ToGear>0
ReqSpeed!
GCTimer:=0
GCTimer>=150
OpenClutch!
ClutchIsOpen?
CheckSyncSpeed
GCTimer<=155
GCTimer:=0
CheckClutch
GCTimer<=200
ClutchOpen
ReqSetGear2
ToGear>0
GCTimer<150
SpeedSet?
ReqSet!
GCTimer:=0
ReqSet!
ReqSetGear
ReqSet!
GCTimer:=0
GCTimer>300,
GCTimer<=350
GCTimer>300,
GCTimer<=350
GCTimer:=0
CheckGearSet1
GCTimer<=350
GSetError
CheckGearSet2
GCTimer<=350
ToGear==0
GearSet?
CloseClutch!
GCTimer:=0
GearSet?
CloseClutch!
ClutchIsClosed?
ReqTorqueC
CheckClutchClosed
GCTimer<=200
GCTimer>150,
GCTimer<=200
ClutchClose
GCTimer:=0
ReqTorque!
GCTimer>150,

UPPAAL

QUANTITATIVE Model Checking



Time

Cost

Probability

System Description

Requirement

No!

Debugging Information

Yes

Prototypes
Executable Code
Test sequences

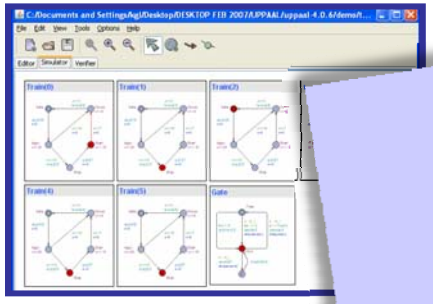
$A\Box(\ req \Rightarrow A\Diamond\ grant)$

$A\Box(\ req \Rightarrow A\Diamond_{t<30s}\ grant)$

$A\Box(\ req \Rightarrow A\Diamond_{t<30s,c<5\$}\ grant)$

$A\Box(\ req \Rightarrow A\Diamond_{t<30s\ ,\ p>0.90}\ grant)$

QUANTITATIVE Model Checking



State/Event Systems
Timed Automata
Stochastic Processes

Probability

System Description

No!
Debugging Information

Requirement

Yes
Prototypes
Code

Datastructures & Algorithms
BDD, DBM, CDD, MTBDD, ..

CTL
TCTL
PCTL

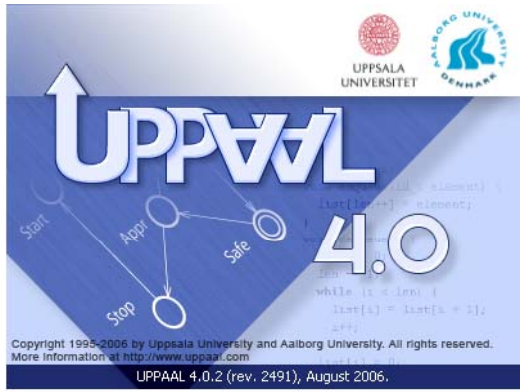
A□(re… <30s grant)

A□(r… t<30s,c<5$ grant)

A□(req ⇒ A◇t<30s , p>0.90 grant)

UPPAAL Branches

CLASSIC



- **Real Time**
  - **Modelling & Verification**
  - **Decidability**
  - **Engine**

- **Real Time**
  - **Scheduling &**
  - **Schedulability Analysis**

CORA

- **Real Time**
  - **Controller Synthesis**
  - **Compositionality**

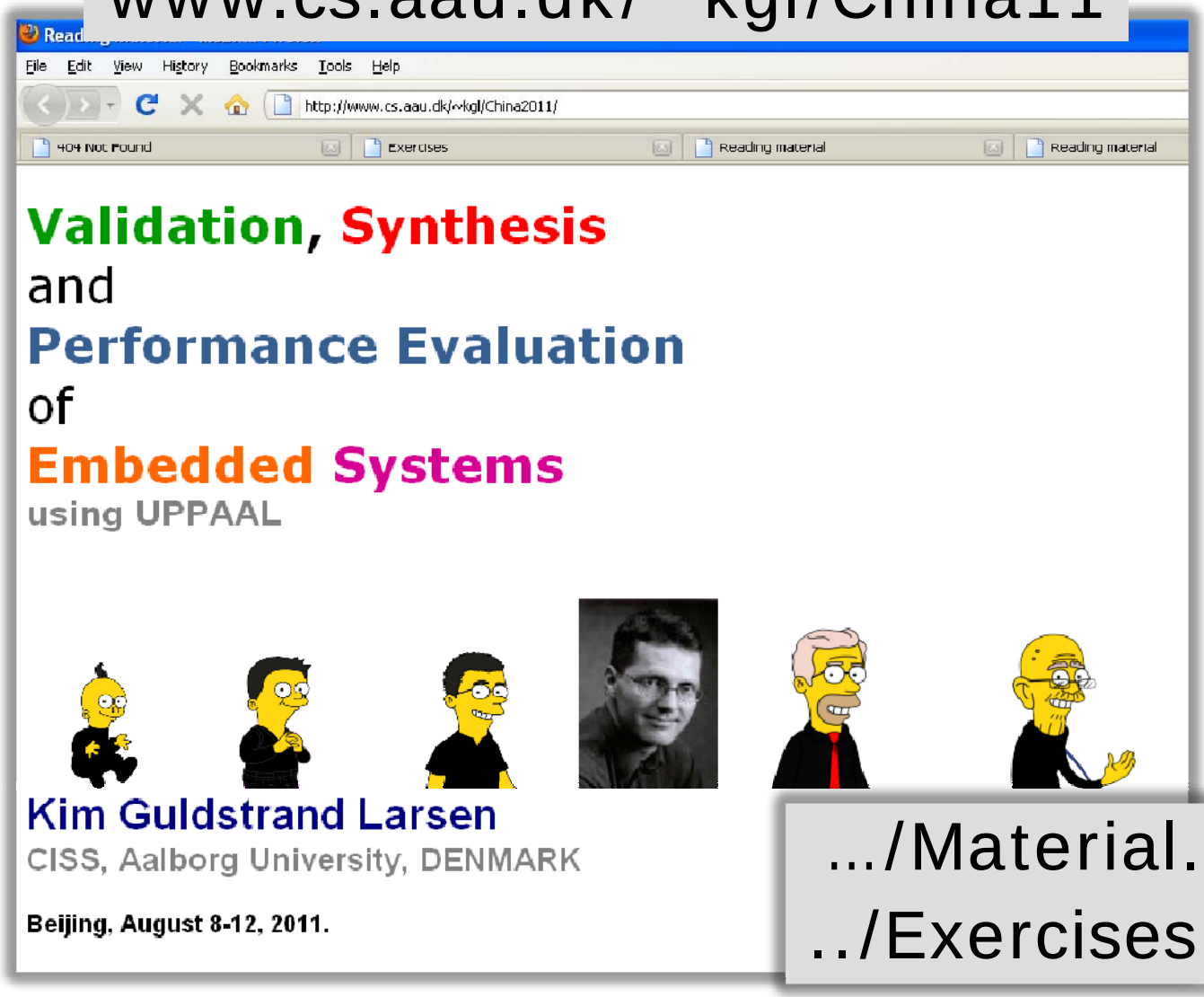
TIGA ECDAR

- **Real Time**
  - **Testing**
  - **Performance Analysis**

TRON SMC

Slides, Reading Material, Exer …

www.cs.aau.dk/~kgl/China11



.../Material.html

../Exercises.html