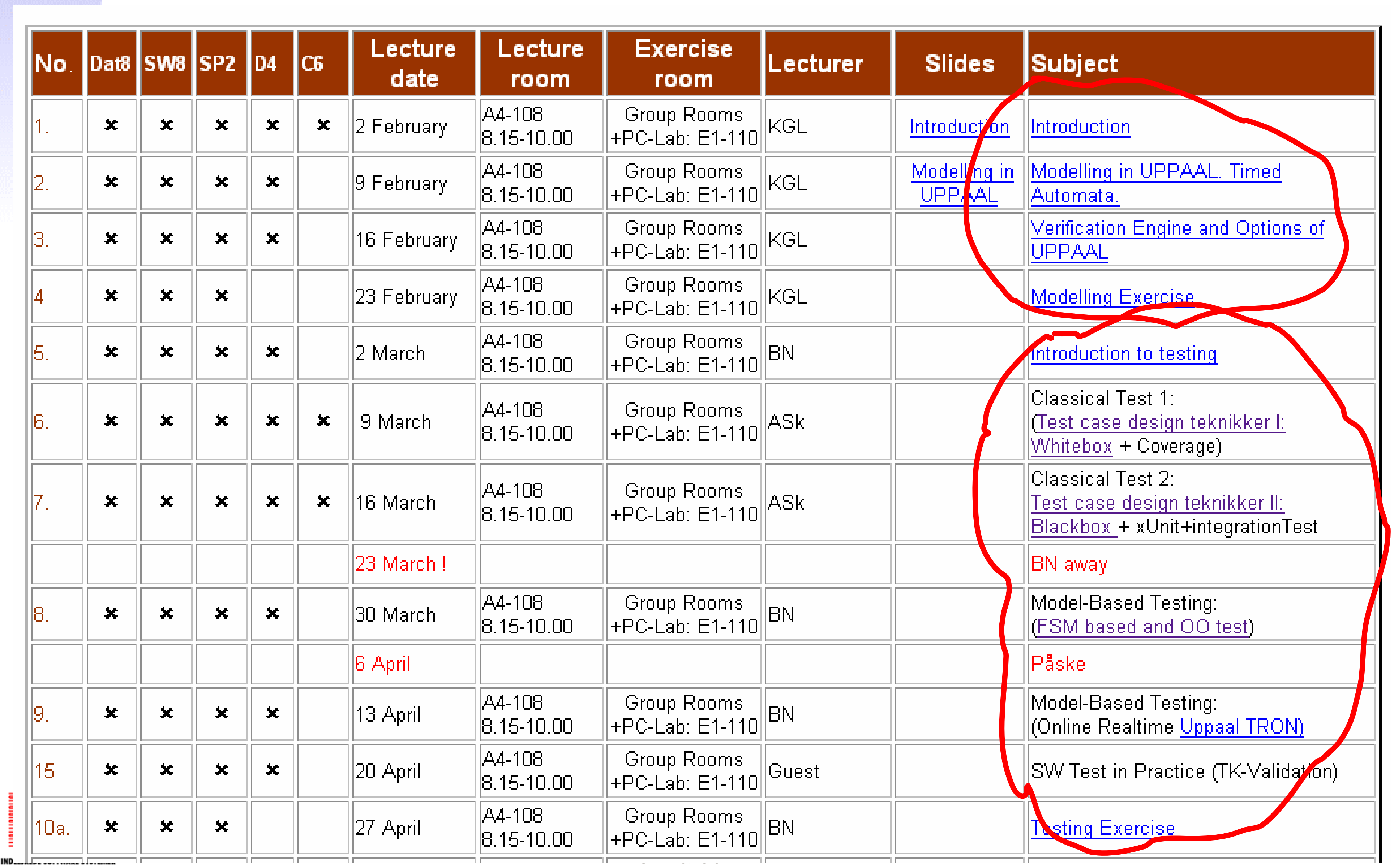
Test & Verifikation

Kim Guldstrand Larsen
Brian Nielsen
Arne Skou

CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

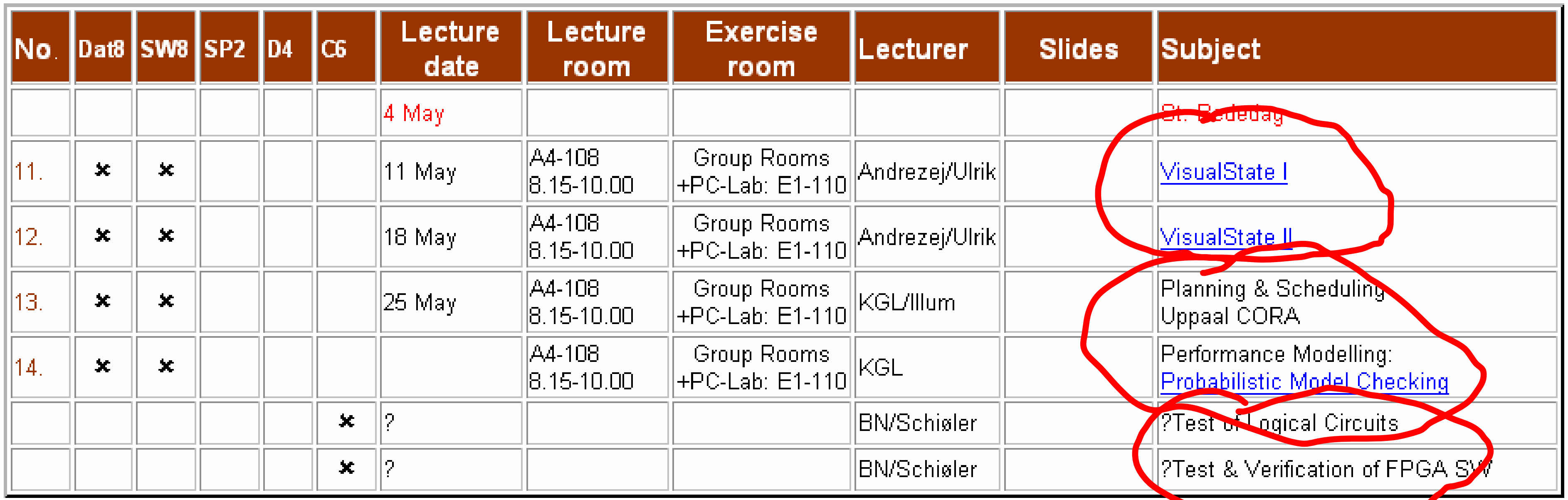
BRICS
Basic Research
in Computer Science

Plan for kursus



No.	Dat8	SW8	SP2	D4	C6	Lecture date	Lecture room	Exercise room	Lecturer	Slides	Subject
1.	×	×	×	×	×	2 February	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	KGL	Introduction	Introduction
2.	×	×	×	×		9 February	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	KGL	Modelling in UPPAAL	Modelling in UPPAAL. Timed Automata.
3.	×	×	×	×		16 February	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	KGL		Verification Engine and Options of UPPAAL
4	×	×	×			23 February	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	KGL		Modelling Exercise
5.	×	×	×	×		2 March	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	BN		Introduction to testing
6.	×	×	×	×	×	9 March	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	ASk		Classical Test 1: (Test case design teknikker I: Whitebox + Coverage)
7.	×	×	×	×	×	16 March	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	ASk		Classical Test 2: Test case design teknikker II: Blackbox + xUnit+integrationTest
						23 March !					BN away
8.	×	×	×	×		30 March	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	BN		Model-Based Testing: (FSM based and OO test)
						6 April					Påske
9.	×	×	×	×		13 April	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	BN		Model-Based Testing: (Online Realtime Uppaal TRON)
15	×	×	×	×		20 April	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	Guest		SW Test in Practice (TK-Validation)
10a.	×	×	×			27 April	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	BN		Testing Exercise

Plan for kursus



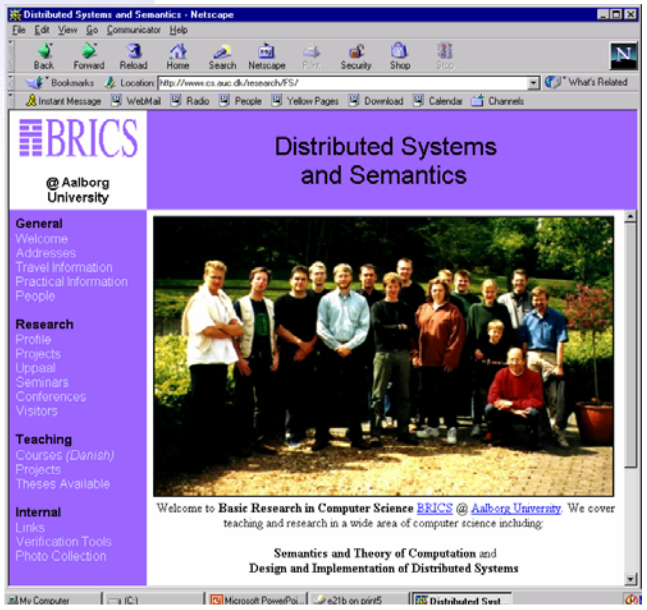
No.	Dat8	SW8	SP2	D4	C6	Lecture date	Lecture room	Exercise room	Lecturer	Slides	Subject
						4 May					St. Bededag
11.	×	×				11 May	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	Andrezej/Ulrik		VisualState I
12.	×	×				18 May	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	Andrezej/Ulrik		VisualState II
13.	×	×				25 May	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	KGL/Illum		Planning & Scheduling Uppaal CORA
14.	×	×					A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	KGL		Performance Modelling: Probabilistic Model Checking
					×	?			BN/Schiøler		?Test of Logical Circuits
					×	?			BN/Schiøler		?Test & Verification of FPGA SW

Plan

- Background
  - Research Group and Projects
- Why (and what) test and verification
- Model-based approach
  - Finite State Machines (review)
  - Interacting State Machines
- Verification=Model Checking (1st glance)
- Model-based Testing (1st glance)

Research Profile

Distributed Systems & Semantics Unit



Distributed Systems and Semantics - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location: http://www.cs.auc.dk/research/FS/ What's Related

Instant Message WebMail Radio People Yellow Pages Download Calendar Channels

BRICS @ Aalborg University

Distributed Systems and Semantics

General
Welcome
Addresses
Travel Information
Practical Information
People

Research
Profile
Projects
Uppaal
Seminars
Conferences
Visitors

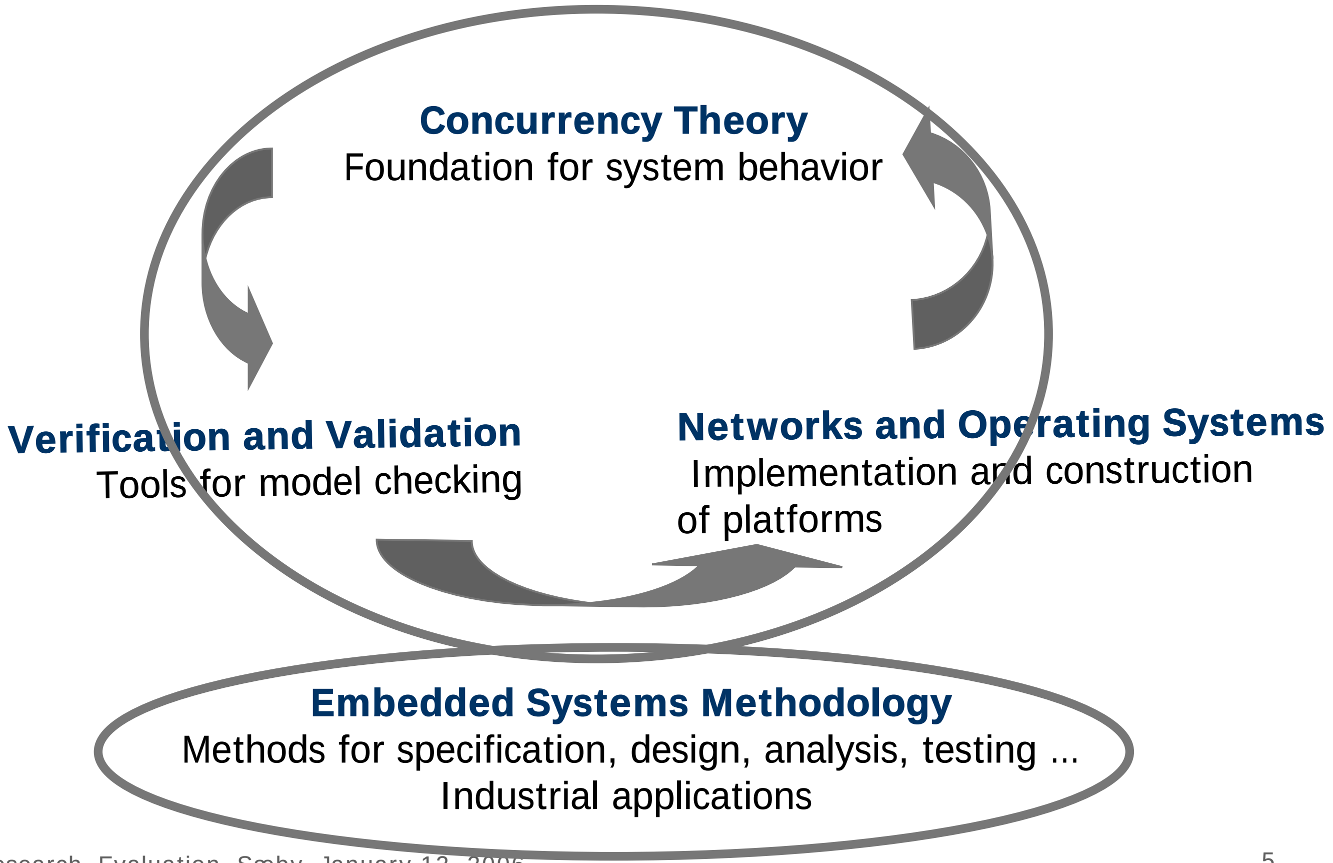
Teaching
Courses *(Danish)*
Projects
Theses Available

Internal
Links
Verification Tools
Photo Collection

Welcome to **Basic Research in Computer Science** BRICS @ Aalborg University. We cover teaching and research in a wide area of computer science including:

Semantics and Theory of Computation and
Design and Implementation of Distributed Systems

My Computer (C:) Microsoft PowerPoi... e21b on print5 Distributed Syst...


Concurrency Theory
Foundation for system behavior
Verification and Validation
Tools for model checking
Networks and Operating Systems
Implementation and construction of platforms
Embedded Systems Methodology
Methods for specification, design, analysis, testing ...
Industrial applications

BRICS Machine

Basic Research in Computer Science, 1993-2006

30+40+40 Millkr

100

100

Tools

Aalborg

Aarhus

Other revelvant projects
ARTIST, AMETIST

BRICS
Basic Research
in Computer Science

Tools and BRICS

Semantics & Verification
(DAT4)
Every Thursday afternoon

Applications

visualSTATE

SPIN

UPPAAL

PVS

HOL

TLP

ALF

Logic

- Temporal Logic
- Modal Logic
- MSOL
-
-

Algorithmic

- (Timed) Automata Theory
- Graph Theory
- BDDs
- Polyhedra Manipulation
-
-

Semantics

- Concurrency Theory
- Abstract Interpretation
- Compositionality
- Models for real-time & hybrid systems
-
-

CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

CISS

Center for Indlejrede Software Systemer

Why CISS ?

- 80% of all software is embedded
- Demands for increased functionality with minimal resources

- Requires multitude of skills
  - Software construction
  - Hardware platforms
  - Control theory
  - Comm. technology

- **Goal**:
  Give a qualitative lift to current industrial practice
  !!!!!

CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

CISS Structure

MVTU
25.5 MDKK

stern

Nordjyllands Amt
Aalborg Kommune
12 MDKK

ES Oldenborg
ES Holland
ARTIS

AAU
12.75 MDKK

IKT Virksomheder

Virksomheder
12.75 MDKK

Institut for Datalogi

Institut for Elektroniske Systemer

BRICS@Aalborg
Modelling and Validation;
Programming Languages;
Software Engineering

Distributed Real Time Systems
Control Theory;
Real Time Systems;
Networking.

Embedded Systems
Communication;
HW/SW
Power Management

CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

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

CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

CISS on the way

- Kick-start, 2001:
  700.000 DKK
  Northern Jutland Region & City of Aalborg

- **35** projects
- **20** CISS employees
- **25** CISS associated researcher at 3 different research groups at AAU.

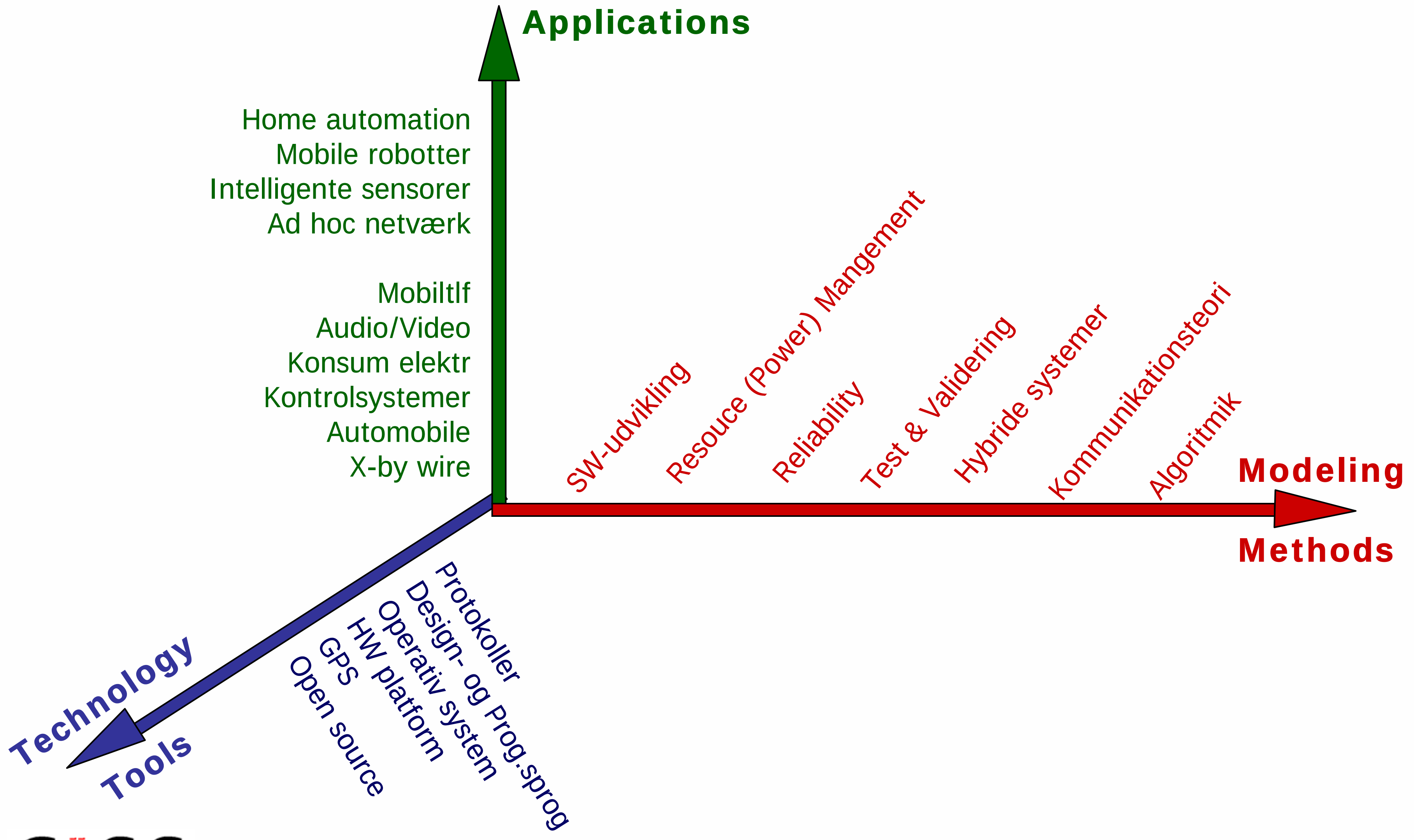
- Jutland-Fun IT-initiative, 2002:

25,5	mil. kr Ministry
6	mil. kr North Jutland
6	mil. kr Aalborg City
12,75	mil. kr Companies
12,75	mil. kr AAU

- **50%** over budgetteret industrial financing
- **19** industrial Ph.D.'s initiated

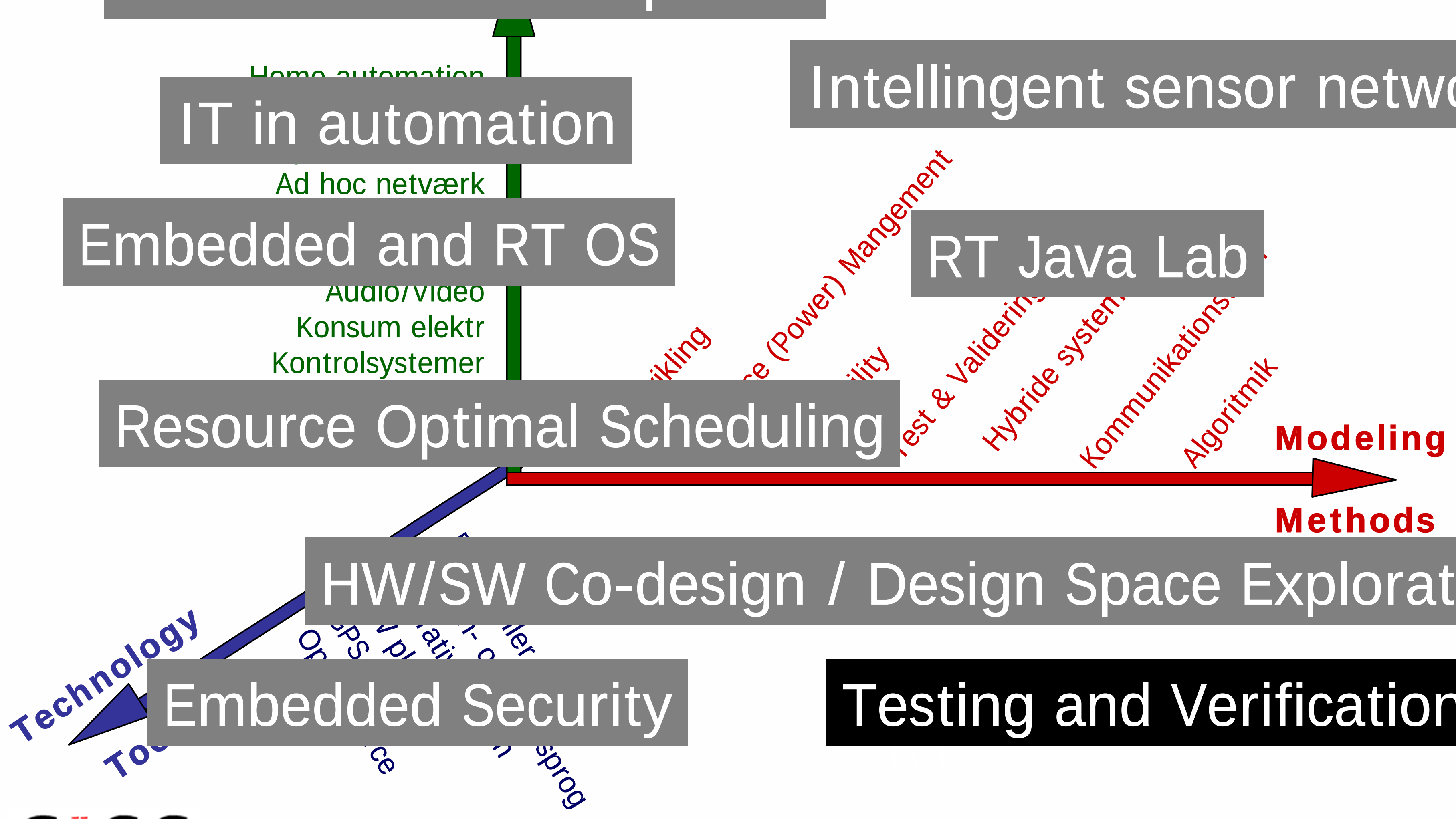
CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

Focus Areas



CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

Focus Areas



Model based development

IT in automation

Intellingent sensor network

Embedded and RT OS

RT Java Lab

Resource Optimal Scheduling

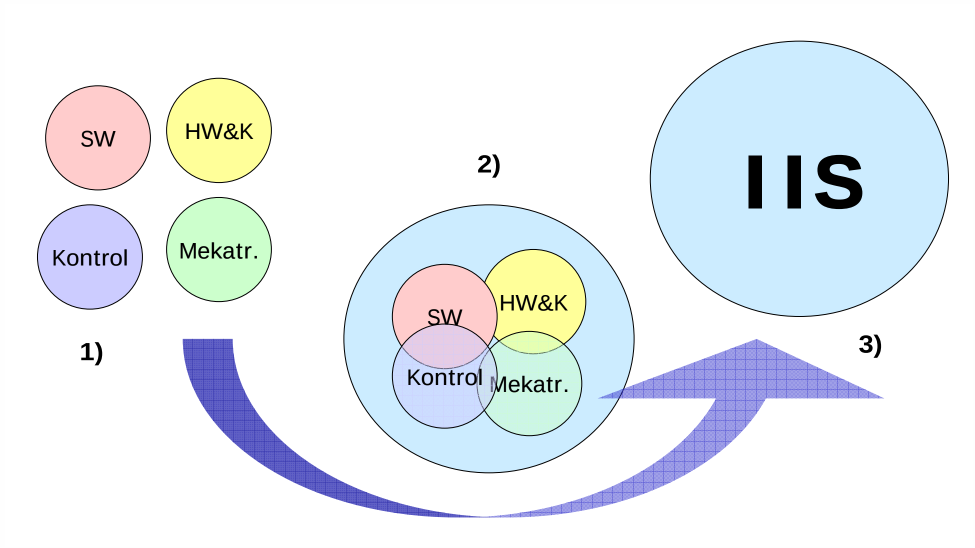
HW/SW Co-design / Design Space Exploration

Embedded Security

Testing and Verification

CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

Local → Regional → National



DaNES

- Danish Network for Intelligent Embedded Systems
- **PARTNERS**

  CISS, IMM, MCI,
  PAJ Systemteknik
  GateHouse A/S
  ICE Power
  Skov A/S
  Terma A/S
  Novo Nordisk A/S
  IO Technologies

- **Funded** by Højteknologifonden

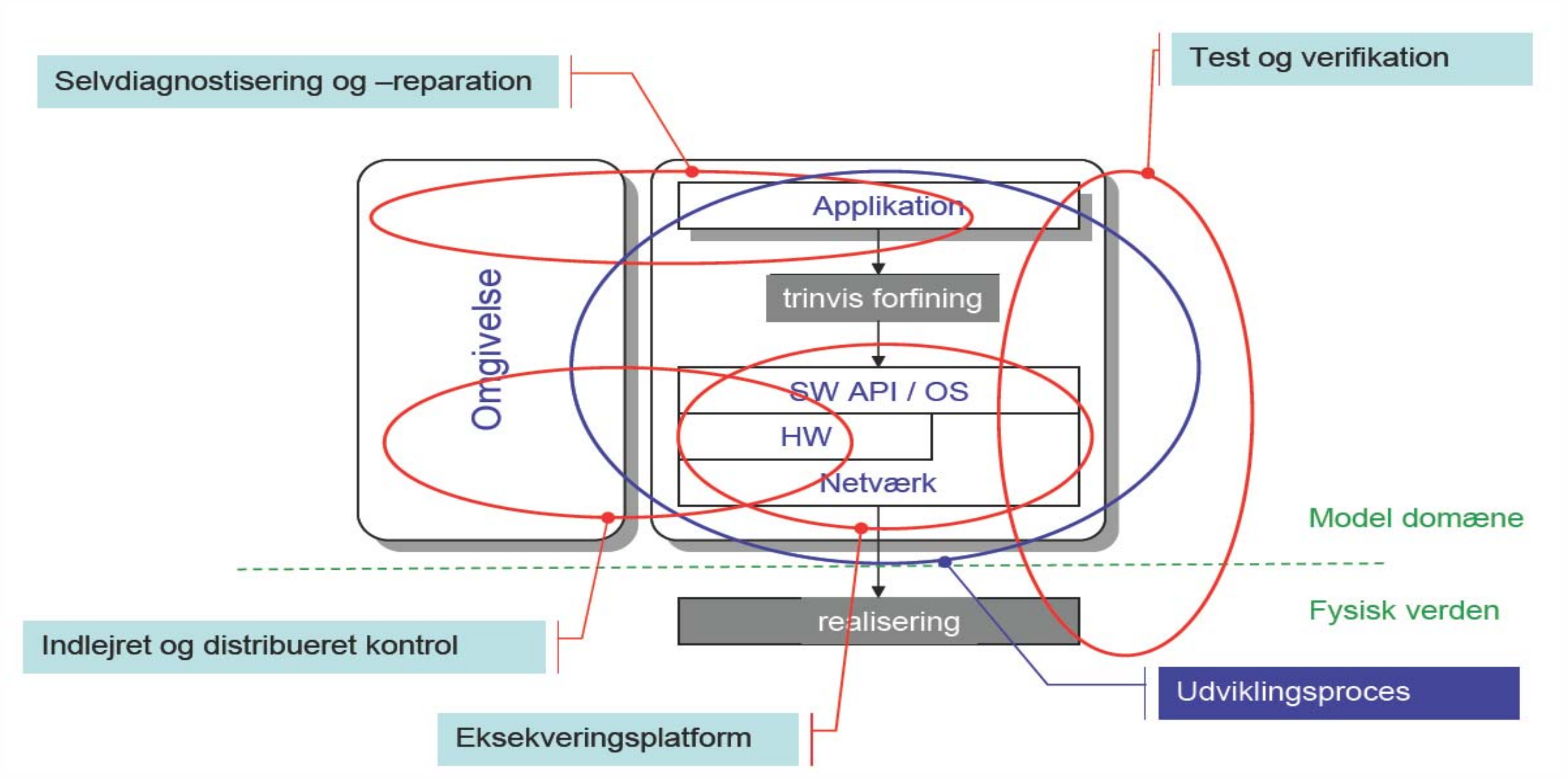
- **Budget**

  63 MDKK / 4 years

CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

Local → Regional → National

DaNES

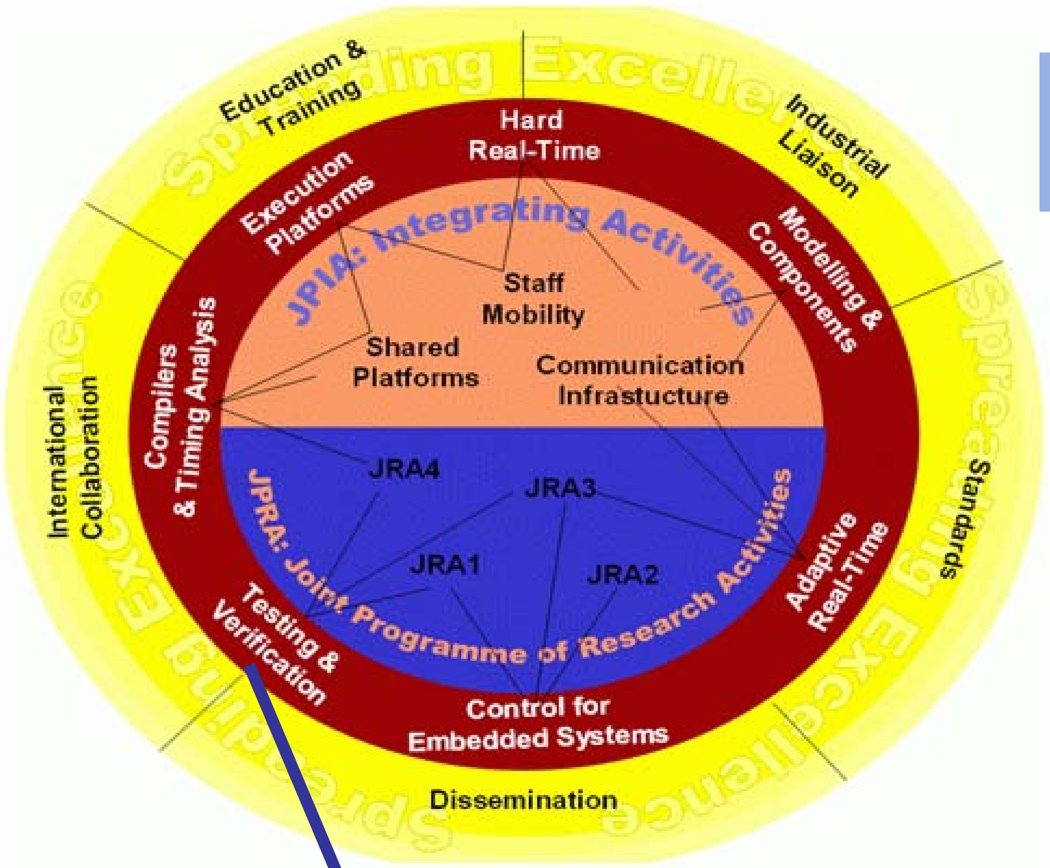


Local → Regional → National → International

ARTIST2 Network of Excellence

6,5MEuro, 32 partners

Information Society Technologies



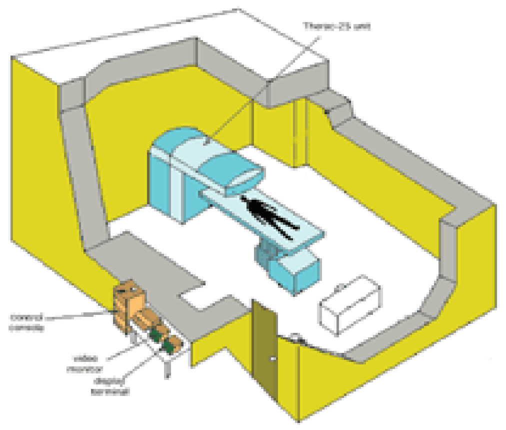
embeddedworld2005 Exhibition&Conference Nürnberg

EU's 7th Framework
→
ARTEMIS Research Platform
→
Centers of Excellence

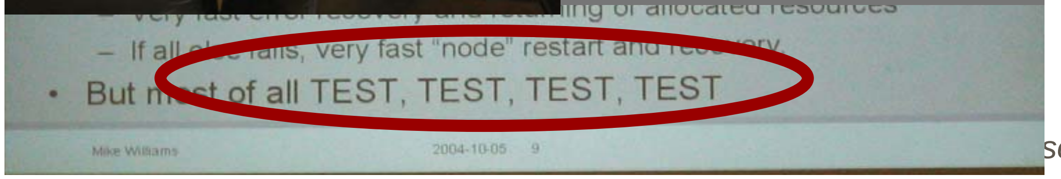
Testing & Verification
CISS koordinator

CISS CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

Hvorfor T&V ?



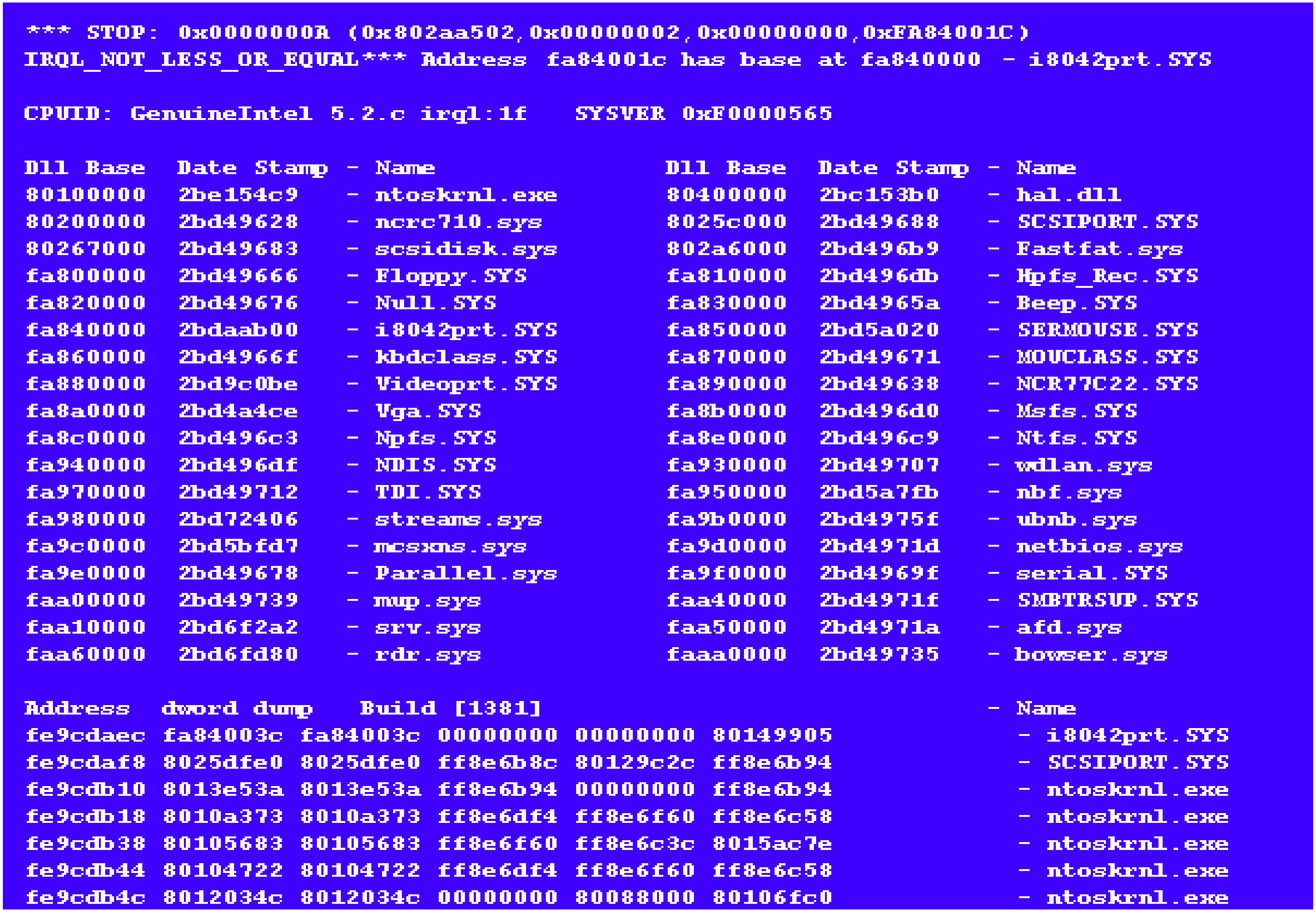
Michael Williams
Research Director, Ericsson, SE



– Very fast error recovery and returning of allocated resources
– If all else fails, very fast "node" restart and recovery
• But most of all TEST, TEST, TEST, TEST

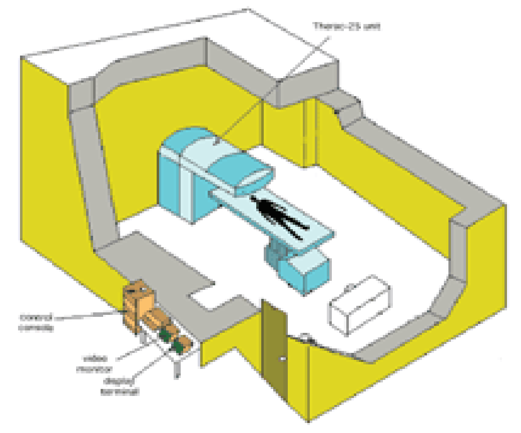
Mike Williams 2004-10-05 9

- Fejl i indlejret software forbundet med voldsomme udgifter.

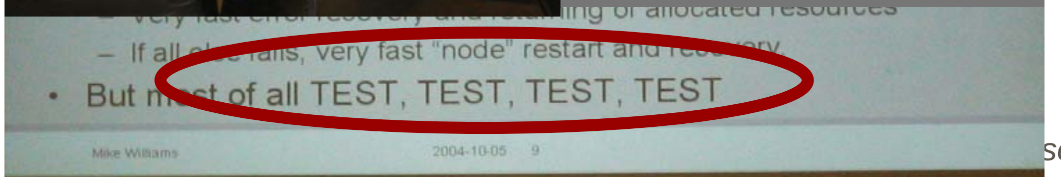


```
*** STOP: 0x0000000A (0x802aa502,0x00000002,0x00000000,0xFA84001C)
IRQL_NOT_LESS_OR_EQUAL*** Address fa84001c has base at fa840000 - i8042prt.SYS

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   - Hpfs_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  2bd72406   - streams.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
```

Hvorfor T&V ?



Michael Williams
Research Director, Ericsson, SE



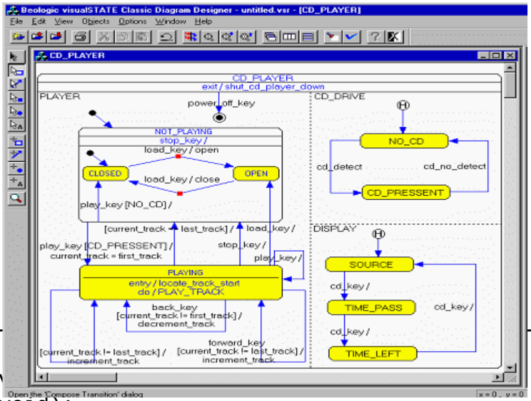
– Very fast error recovery and returning of allocated resources
– If all else fails, very fast "node" restart and recovery.
• But most of all TEST, TEST, TEST, TEST

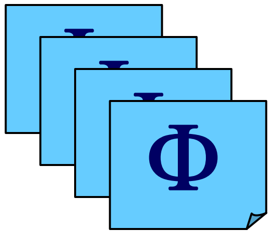
Mike Williams 2004-10-05 9

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

Verifikation og Test

Model





Spec

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

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

/* Define completi
unsigned char cc;

void HandleError(un
{
  printf("Error code
  exit(ccArg);
}


/* In d-241 we only us                              simulate a
 * system. It purpose                     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;
```

- Verifikation
  Kode/Model mht Spec
- Test
  System mht Model/Spec

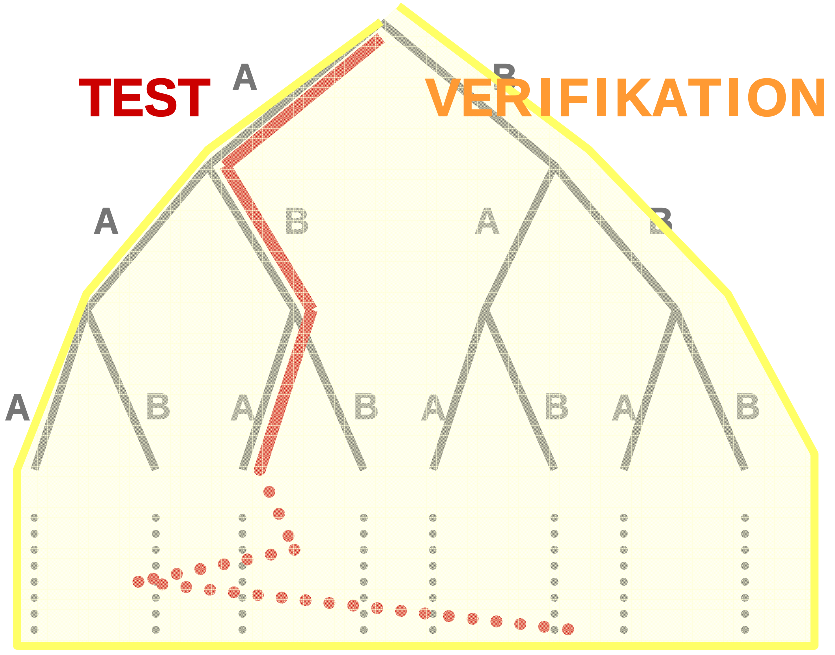
System

Kode

Test versus Verifikation

E F E E G H ... H A

T1 T3 T5 T1 ... T4 T3



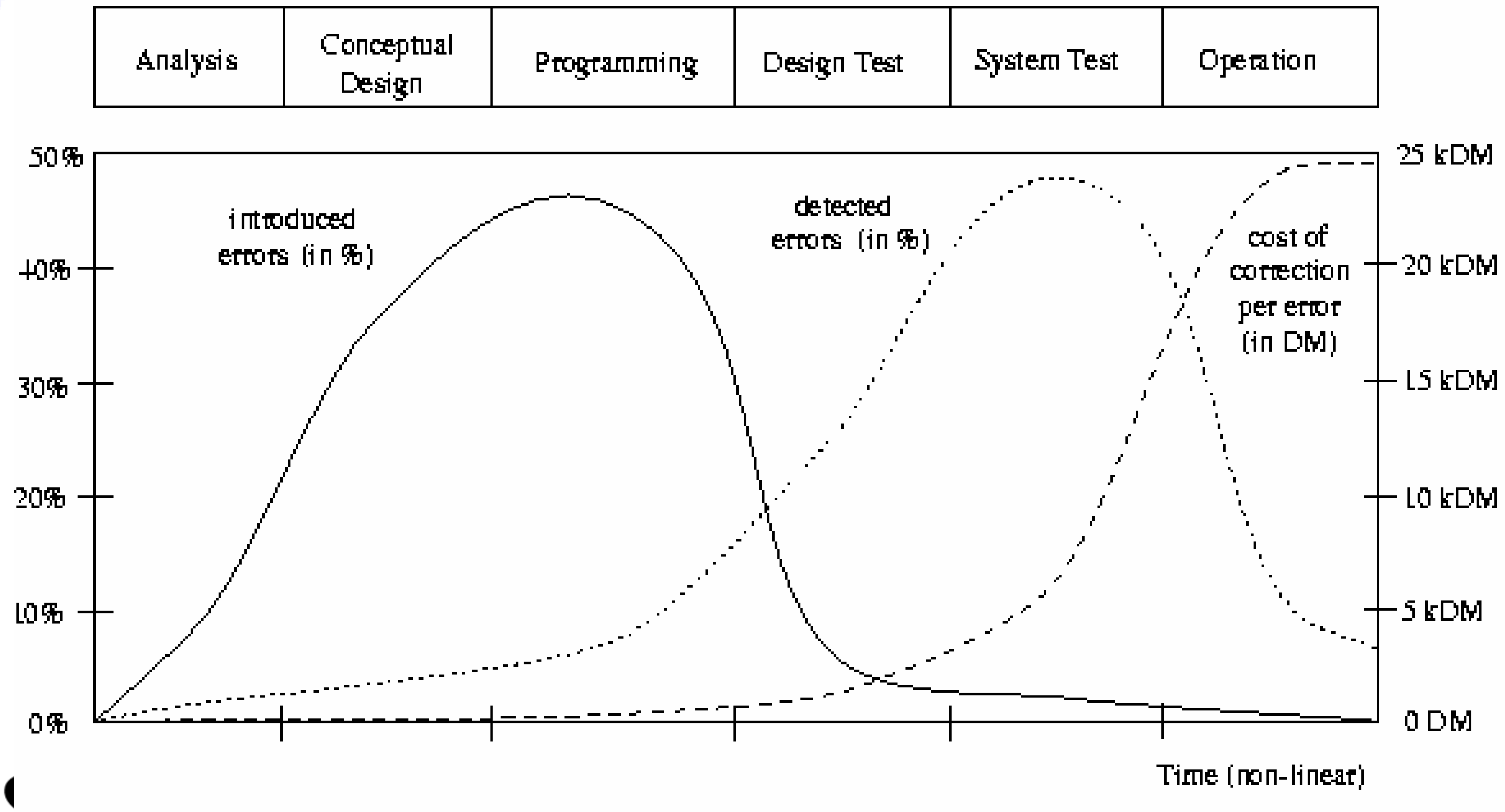
2^n sekvenser af lgd n

Deadlock identificeret ved **VERIFIKATION** efter sekvens på 2000 telegrammer / < 1min

UPPAAL

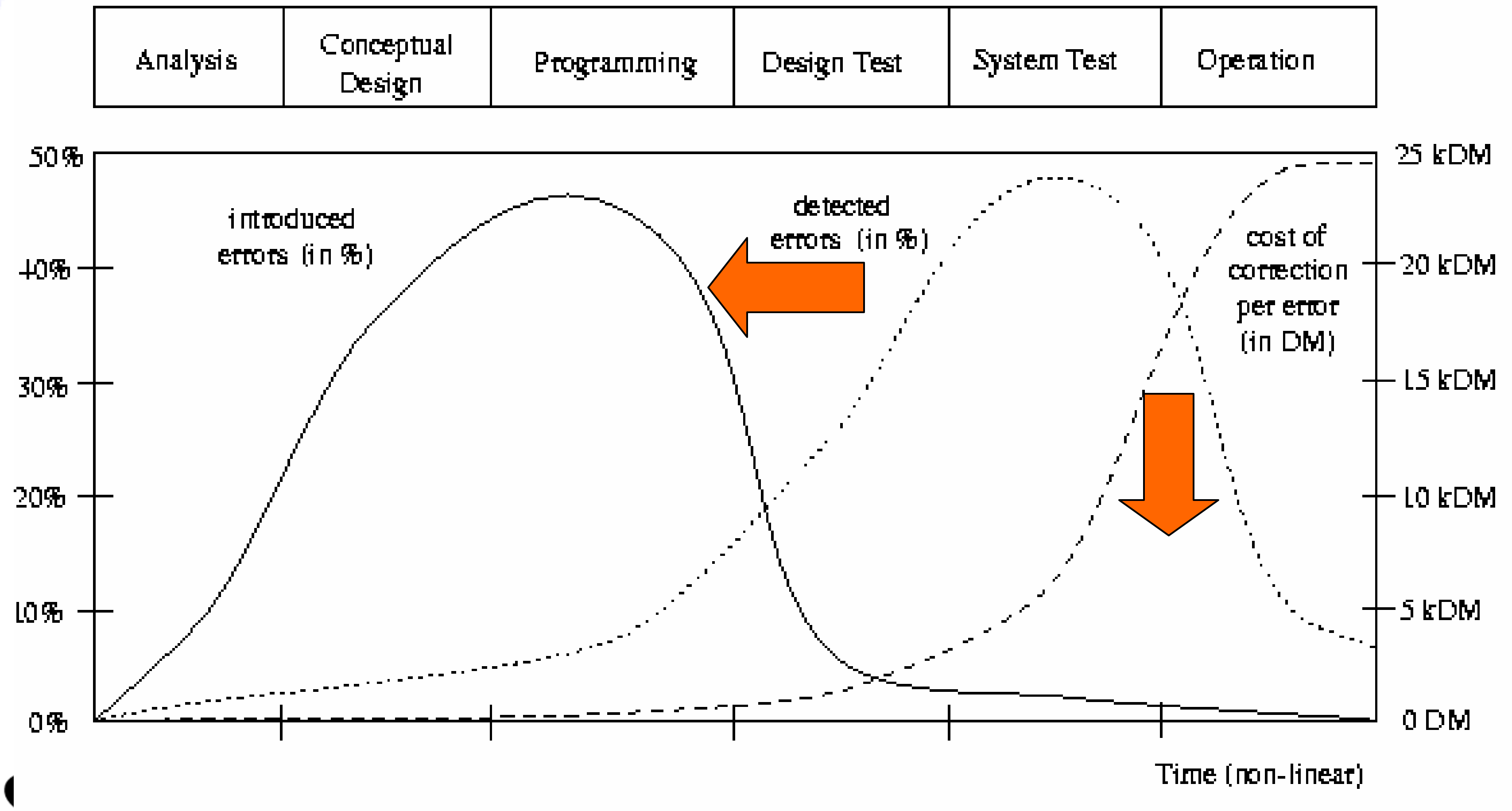
Introducing, Detecting and Repairing Errors

Liggesmeyer 98



Introducing, Detecting and Repairing Errors

Liggesmeyer 98



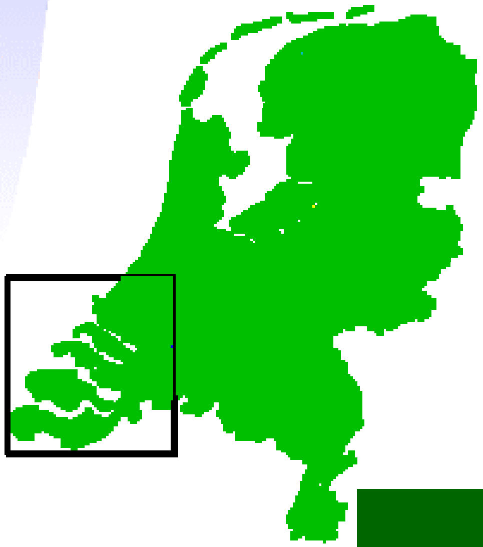
BRICS Basic Research in Computer Science

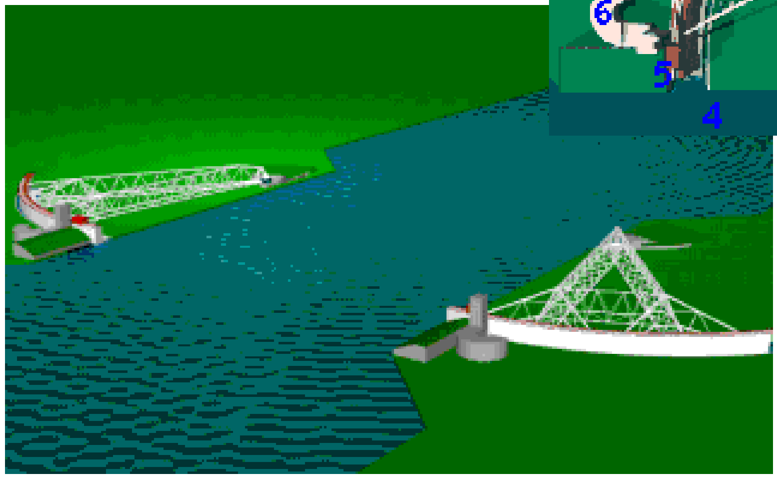
A very complex system

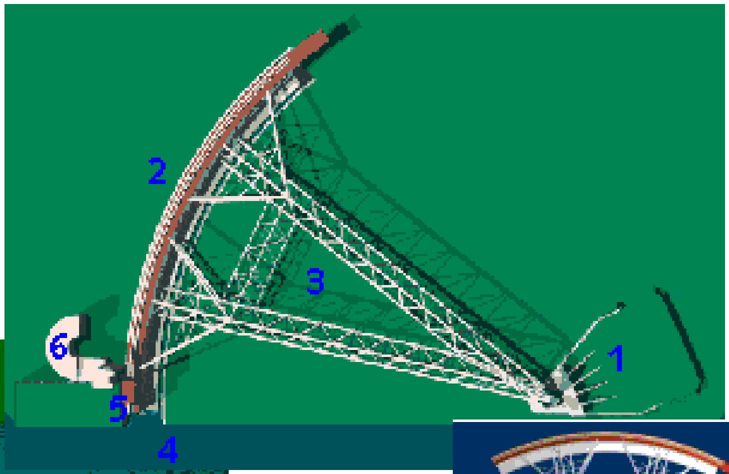
Klaus Havelund, NASA

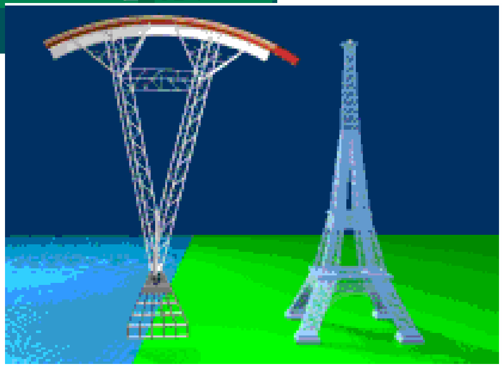
CISS CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

Rotterdam Storm Surge Barrier









BRICS Basic Research in Computer Science

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.

CISS CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

Spectacular software bugs U.S.S. Yorktown, U.S. Navy

- In 1998, the USS Yorktown became the first ship to test the US Navy's Smart Ship program. The Navy planned to use off-the-shelf computers and software instead of expensive U.S.S. Yorktown, courtesy of U.S. Navy custom-made machines. A sailor mistakenly entered a zero for a data value on a computer. Within minutes, Yorktown was dead in the water. It was several hours before the ship could move again.
- When the sailor entered the mistaken number, the computer tried to divide by zero, which isn't possible. The software didn't check to see if the inputs were valid before computing and generated an invalid answer that was used by another computer. The error cascaded several computers and eventually shut down the ship's engines.

Spectacular software bugs Moon or Missiles

- The United States established the Ballistic Missile Early Warning System (BMEWS) during the Cold War to detect a Soviet missile attack. On October 5, 1960 the BMEWS radar at Thule, Greenland detected something. Its computer control system decided the signal was made by hundreds of missiles

- The radar had actually detected the Moon rising over the horizon. Unfortunately, the BMEWS computer had not been programmed to understand what the moon looked like as it rose in the eastern sky, so it interpreted the huge signal as Soviet missiles. Luckily for all of us, the mistake was realized in time.

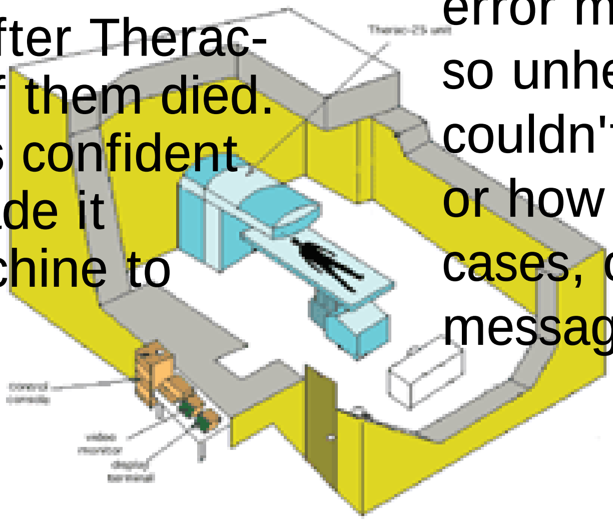
Spectacular Software Bugs .... continued

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

Spectacular software bugs
Therac 25

- 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 completel



“Malfunction 54”

“H-tilt”

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

More complex systems

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

BRICS
Basic Research
in Computer Science

Another simple program

What are the possible final values of x ?

```
int x=0;

Process P
   do
      x:=x+1
   10 times

( P || P )
```

```
int x=0;

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

( P || P )
```

Atomic stm.

CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

BRICS Basic Research in Computer Science

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.

CISS CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

Model-based Approach

CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

BRICS
Basic Research in Computer Science

BRICS
Basic Research
in Computer Science

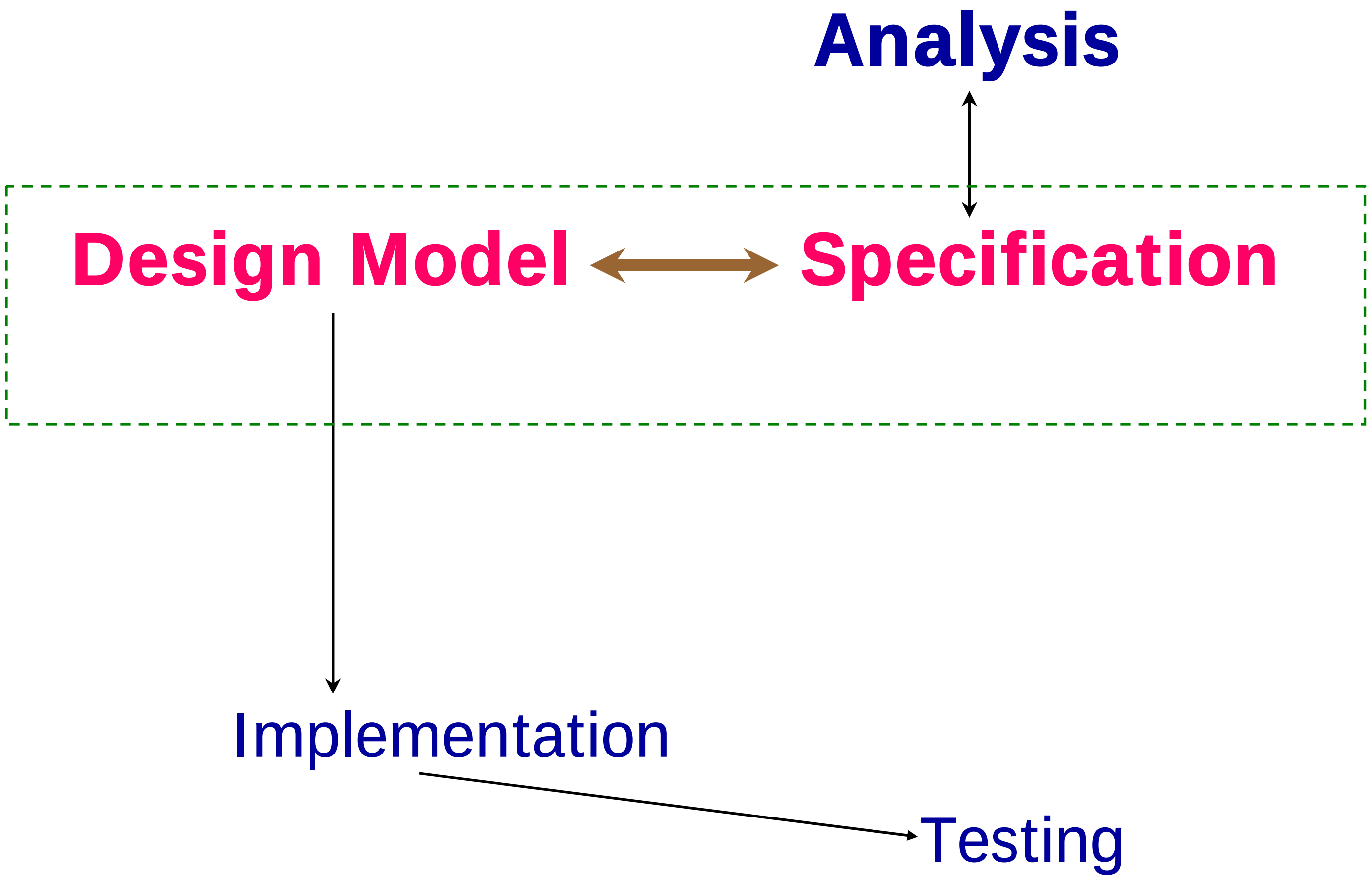
Suggested Solution?

Model based

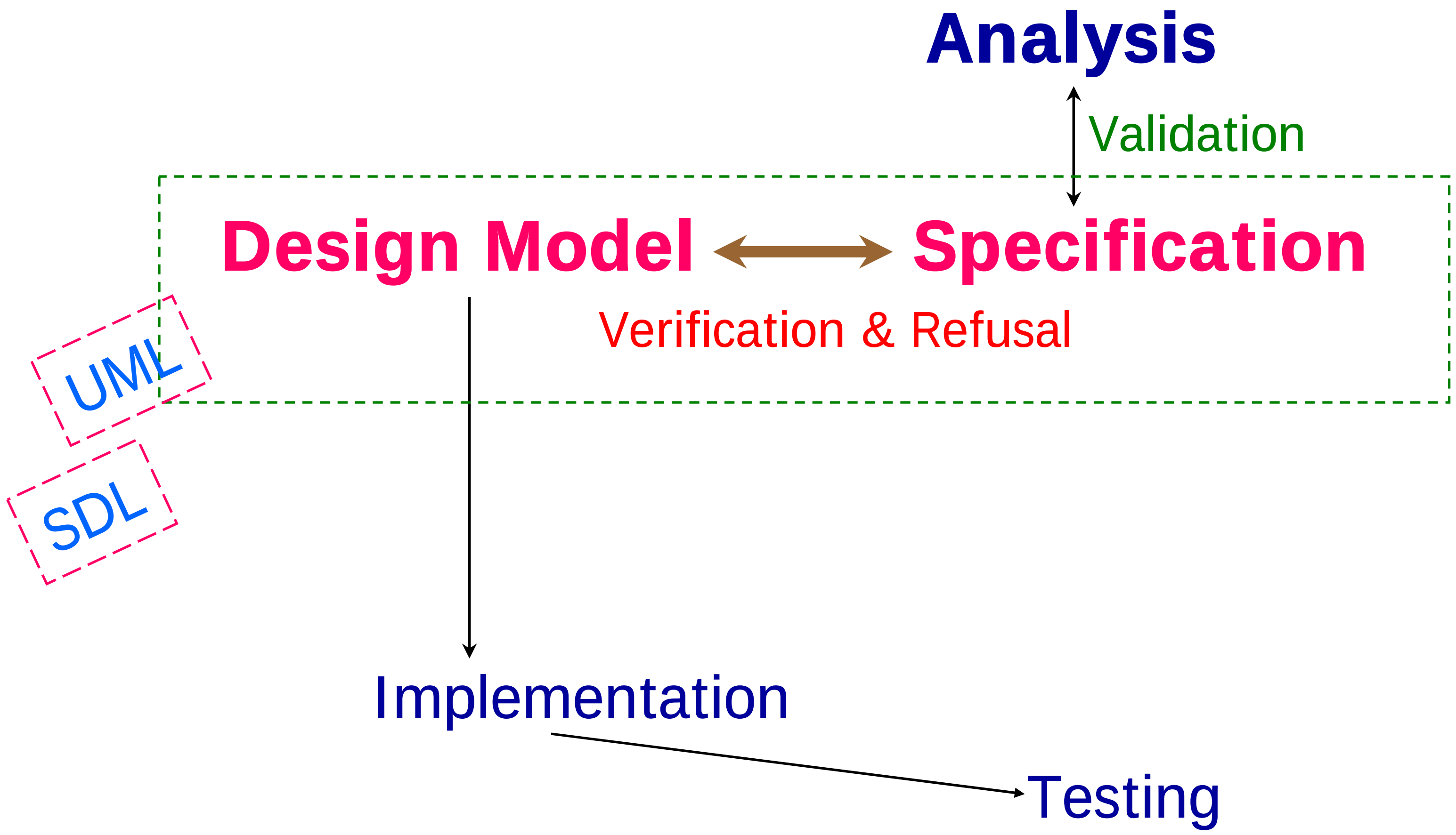
validation, verfication and testing of software and hardware

CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

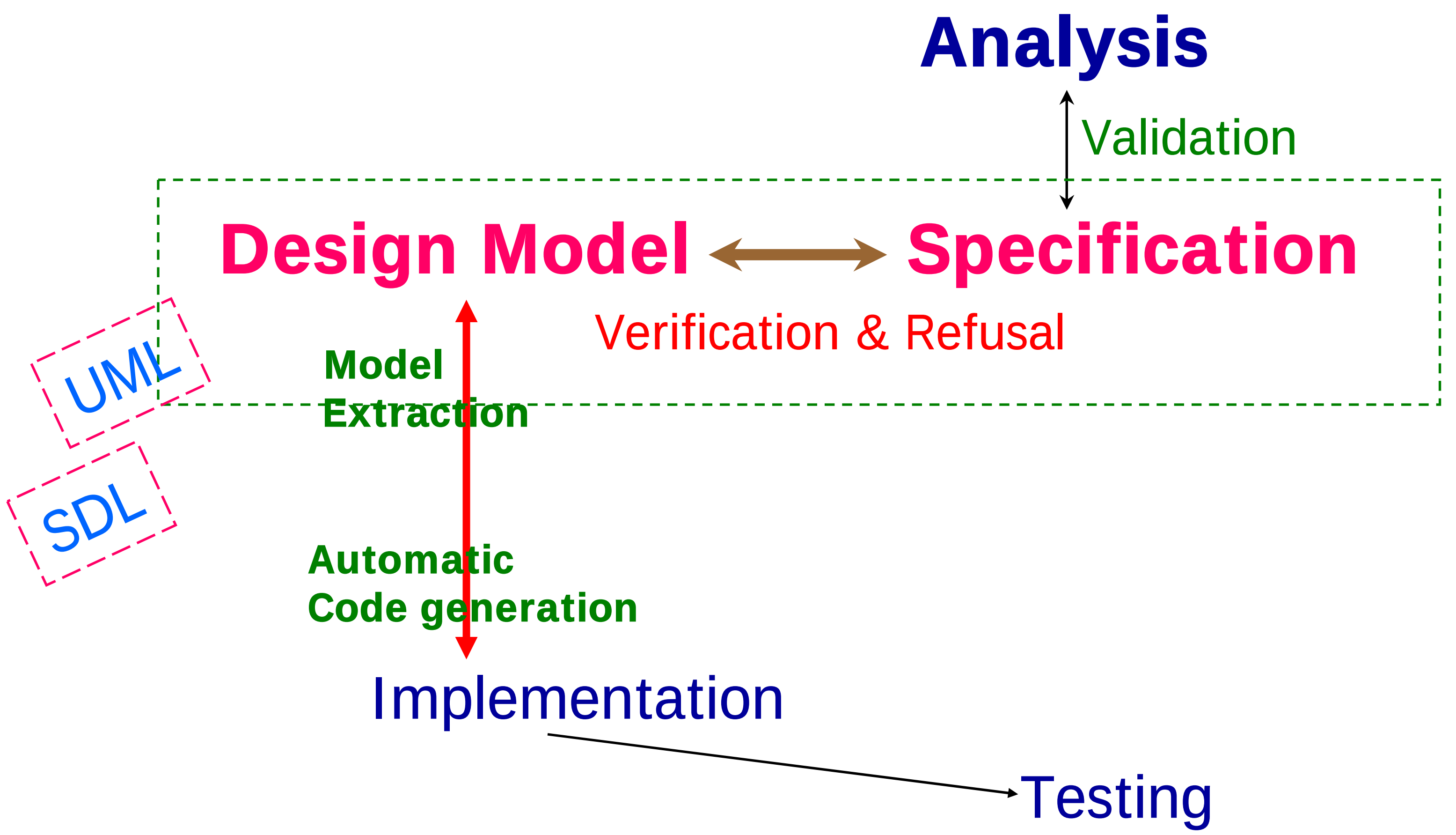
Verification & Validation



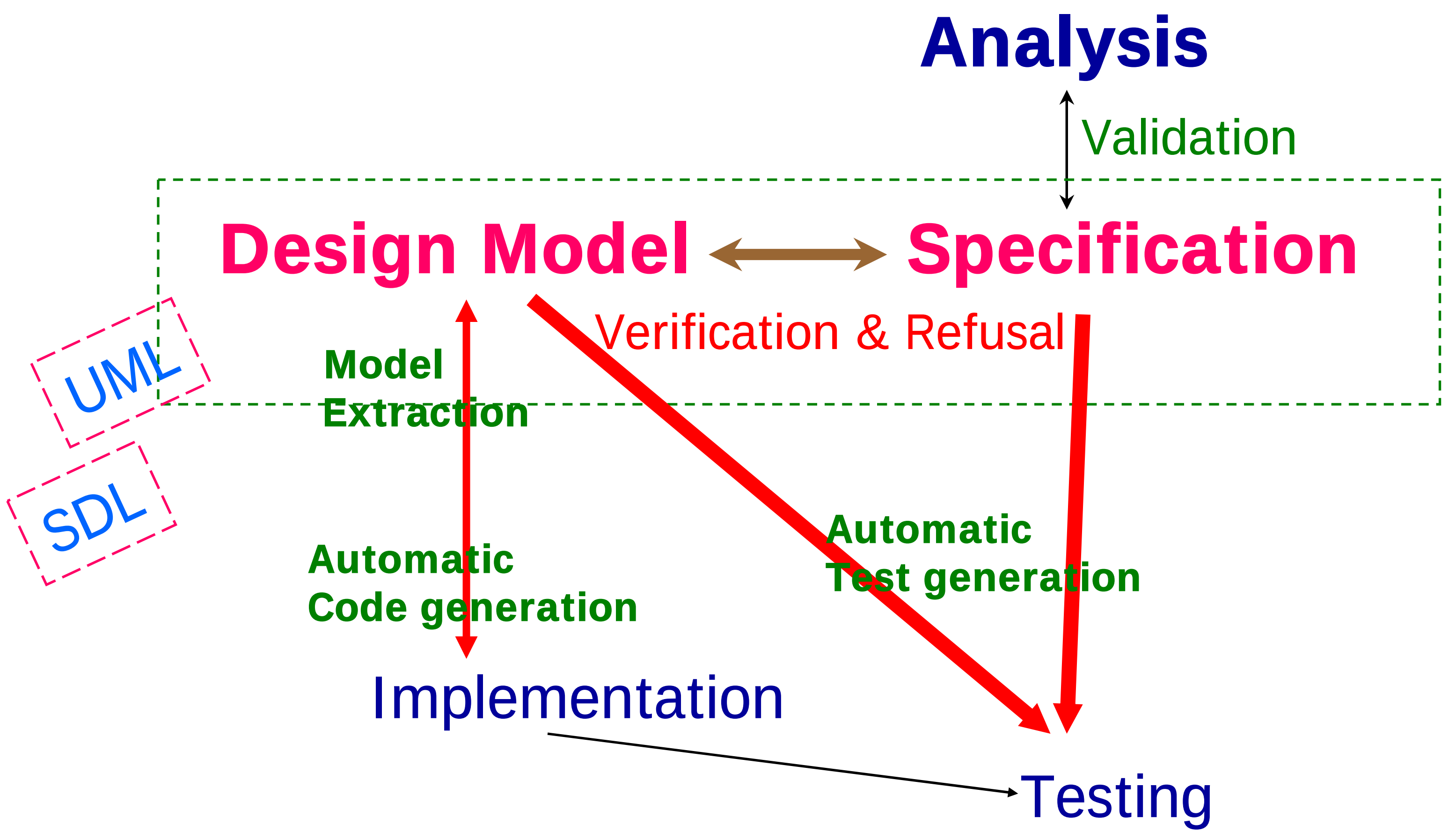
Verification & Validation



Verification & Validation



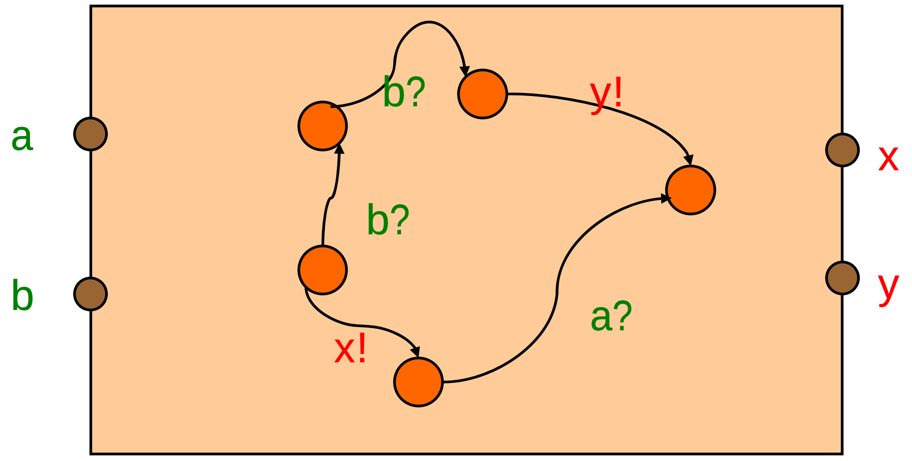
Verification & Validation



How?

Unified Model = State Machine!

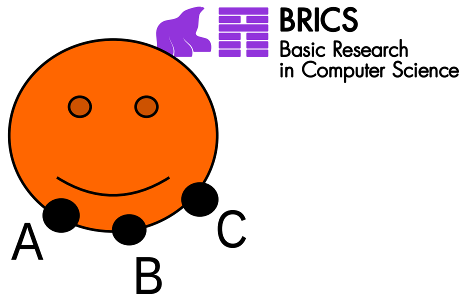
Input ports

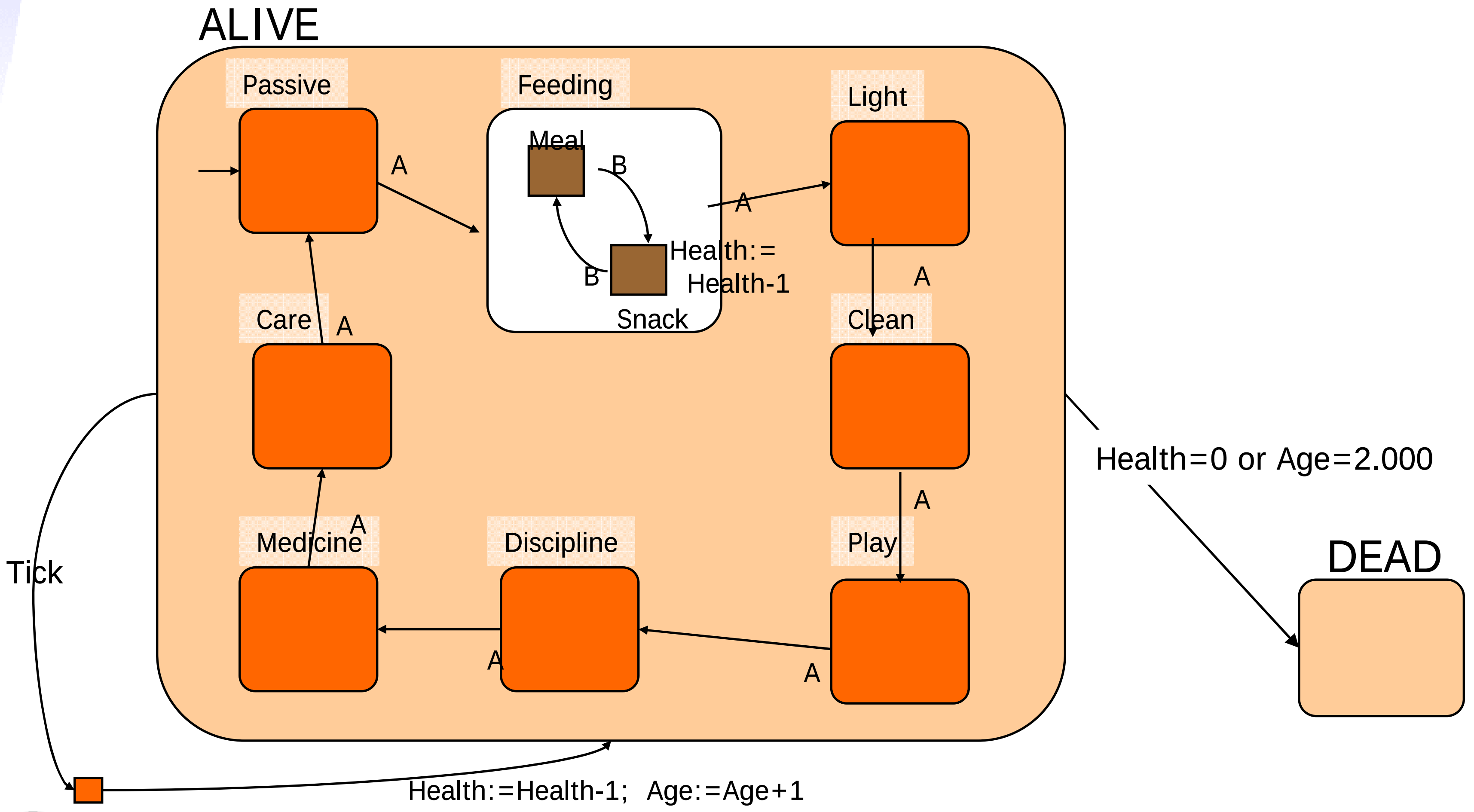


Output ports

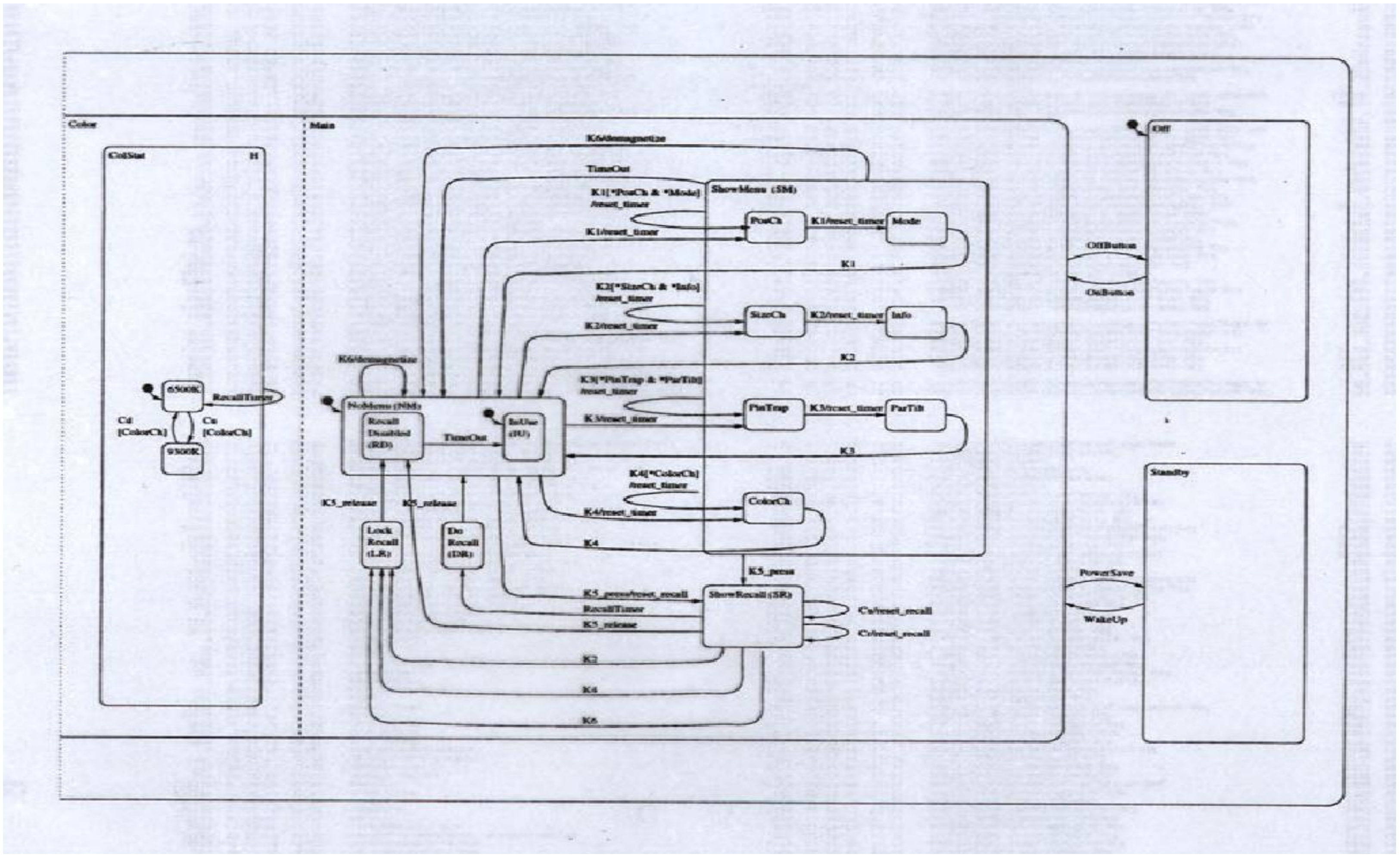
Control states

Tamagotchi

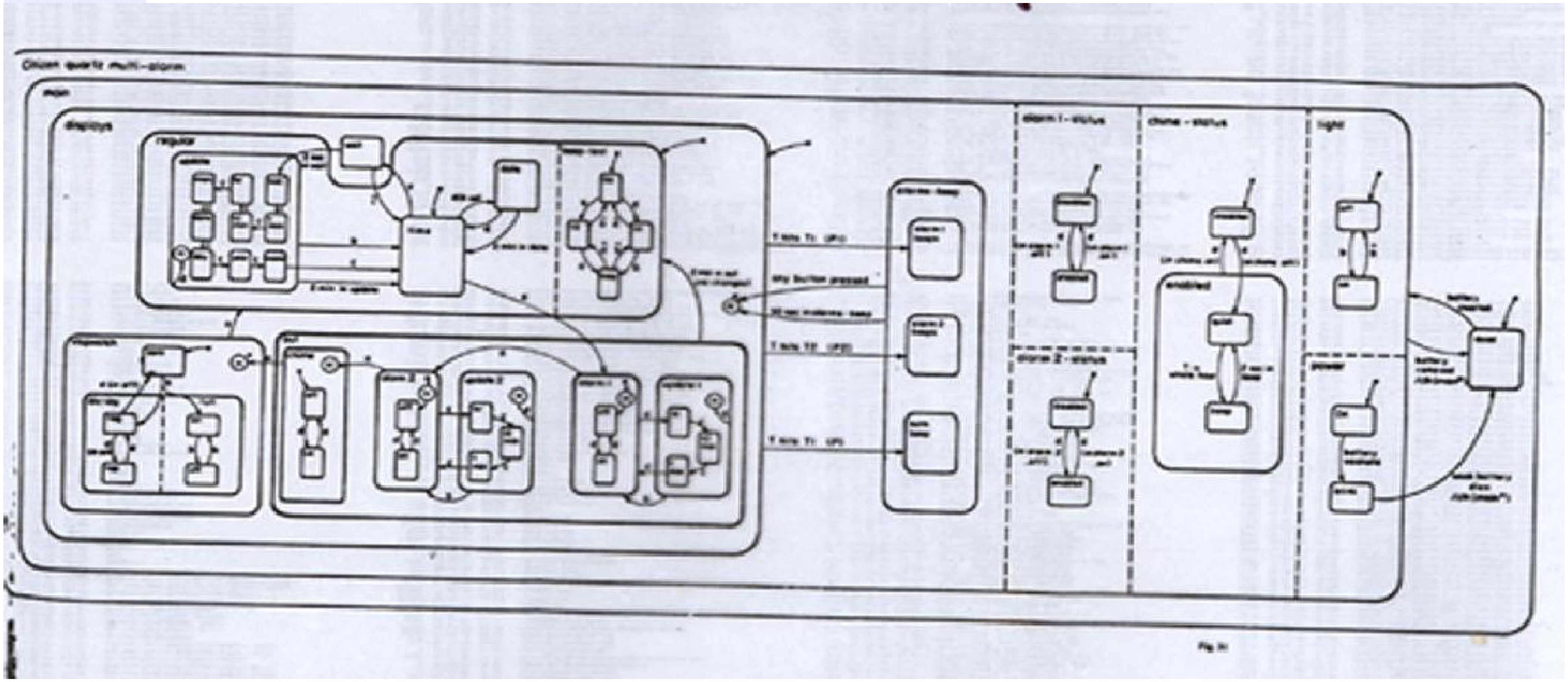

A
B
C


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

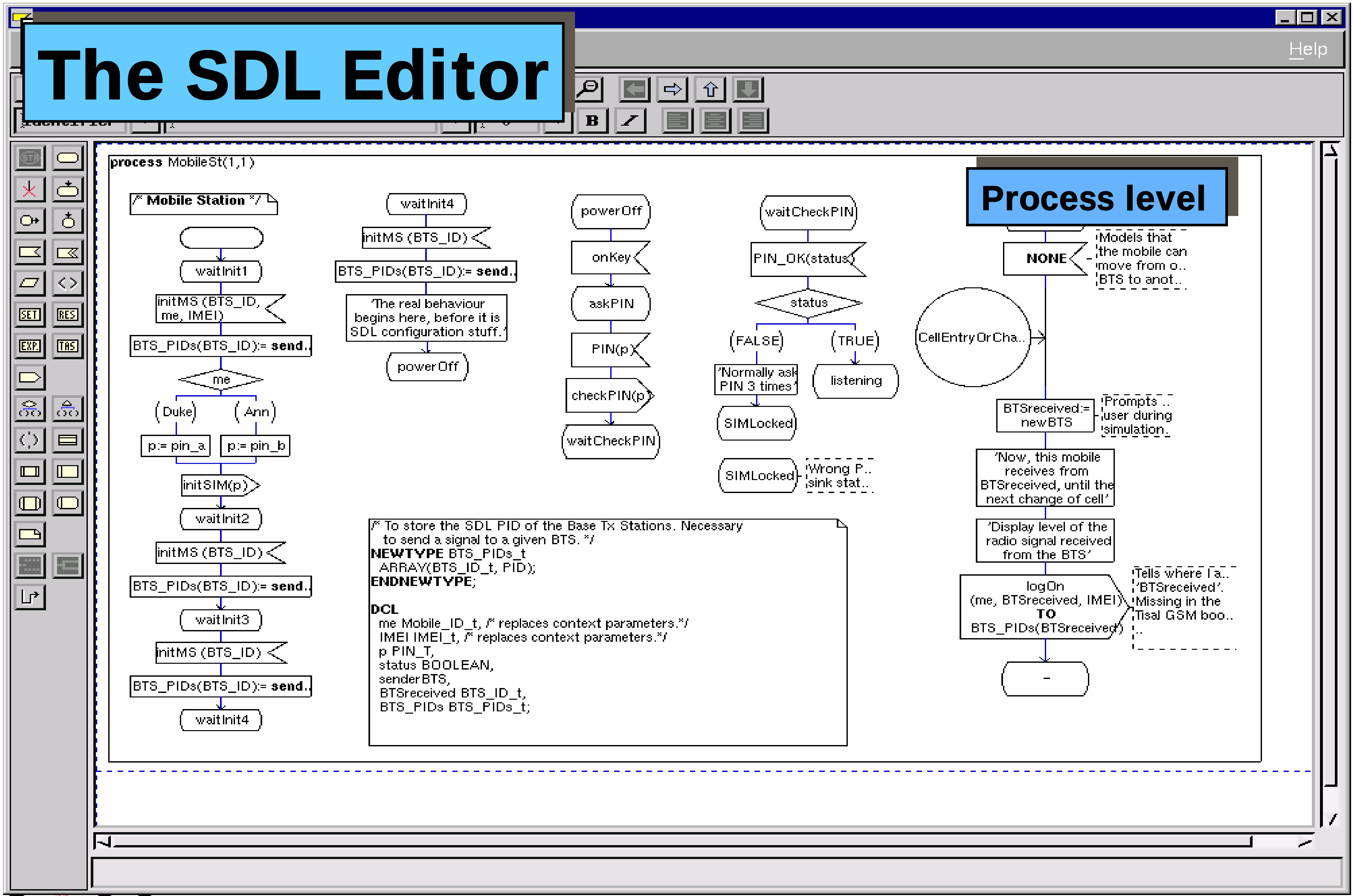
SYNCmaster


Color
ColSet
H
6500K
9300K
RecallTimer
Cd [ColorCh]
Cu [ColorCh]
Main
K6/demagnetize
TimeOut
K1[*PosCh & *Mode] /reset_timer
K1/reset_timer
ShowMenu (SM)
PosCh
K1/reset_timer
Mode
K1
K2[*SizeCh & *Info] /reset_timer
K2/reset_timer
SizeCh
K2/reset_timer
Info
K2
K3[*PinTrap & *ParTilt] /reset_timer
K3/reset_timer
PinTrap
K3/reset_timer
ParTilt
K3
K4[*ColorCh] /reset_timer
K4/reset_timer
ColorCh
K4
NoMenu (NM)
Recall Disabled (RD)
TimeOut
InUse (IU)
K5_release
K5_release
Lock Recall (LR)
Do Recall (DR)
K5_press
K5_press/reset_recall
RecallTimer
K5_release
ShowRecall (SR)
Cu/reset_recall
Cd/reset_recall
K2
K4
K6
Off
OffButton
OnButton
StandBy
PowerSave
WakeUp

Digital Watch



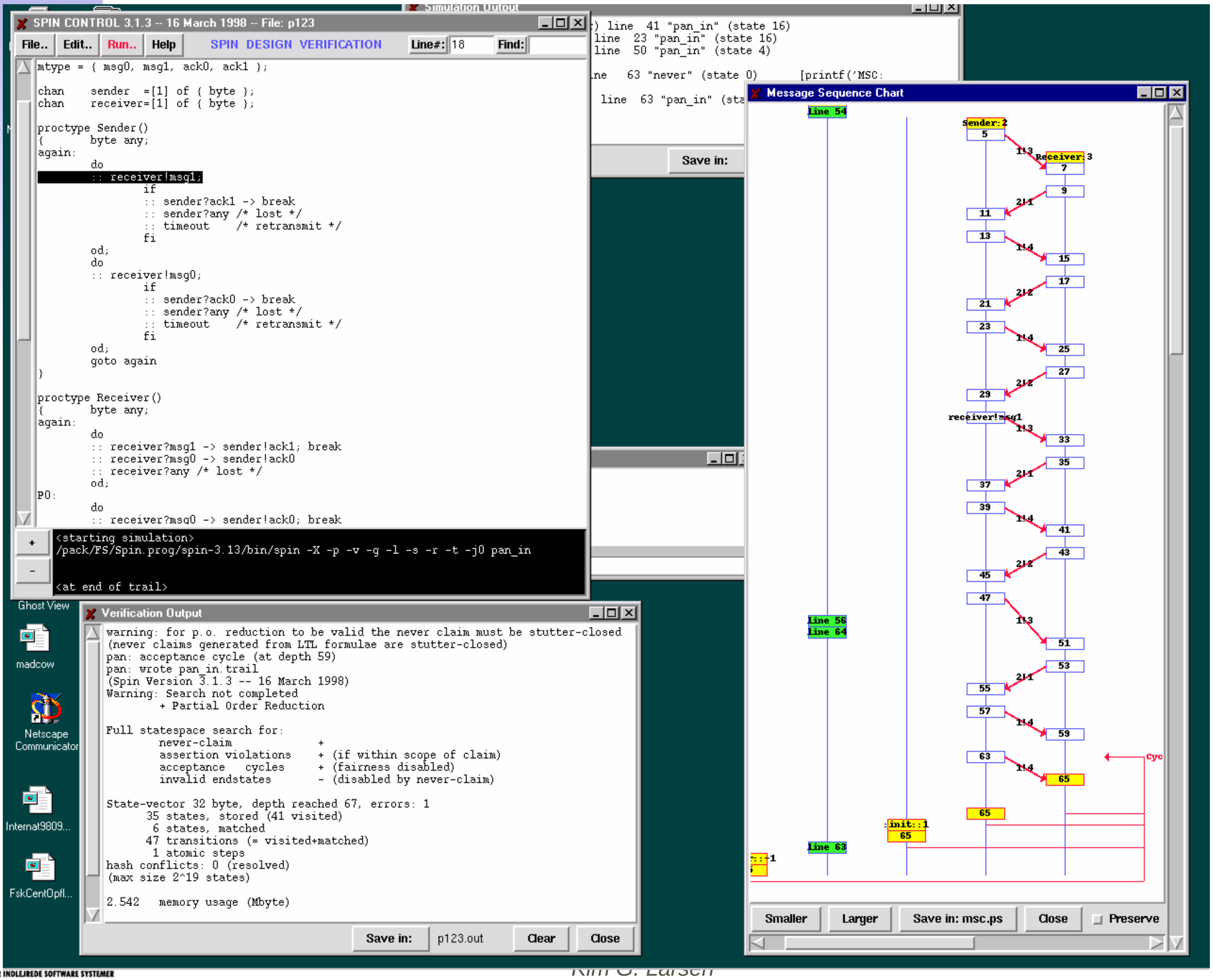
The SDL Editor


Help
identifier
B
process MobileSt(1,1)
/* Mobile Station */
waitInit1
initMS (BTS_ID, me, IMEI)
BTS_PIDs(BTS_ID):= send..
me
Duke
Ann
p:= pin_a
p:= pin_b
initSIM(p)
waitInit2
initMS (BTS_ID)
BTS_PIDs(BTS_ID):= send..
waitInit3
initMS (BTS_ID)
BTS_PIDs(BTS_ID):= send..
waitInit4
waitInit4
initMS (BTS_ID)
BTS_PIDs(BTS_ID):= send..
'The real behaviour begins here, before it is SDL configuration stuff.'
powerOff
powerOff
onKey
askPIN
PIN(p)
checkPIN(p)
waitCheckPIN
waitCheckPIN
PIN_OK(status)
status
FALSE
TRUE
'Normally ask PIN 3 times'
listening
SIMLocked
SIMLocked
Wrong P.. sink stat..
Process level
NONE
Models that the mobile can move from o.. BTS to anot..
CellEntryOrCha..
BTSreceived:= newBTS
Prompts .. user during simulation.
'Now, this mobile receives from BTSreceived, until the next change of cell'
'Display level of the radio signal received from the BTS'
logOn (me, BTSreceived, IMEI) TO BTS_PIDs(BTSreceived)
Tells where I a.. 'BTSreceived'. Missing in the Tisal GSM boo.. ..
-

```
/* To store the SDL PID of the Base Tx Stations. Necessary
   to send a signal to a given BTS. */
NEWTYPE BTS_PIDs_t
  ARRAY(BTS_ID_t, PID);
ENDNEWTYPE;

DCL
  me Mobile_ID_t, /* replaces context parameters.*/
  IMEI IMEI_t, /* replaces context parameters.*/
  p PIN_T,
  status BOOLEAN,
  sender BTS,
  BTSreceived BTS_ID_t,
  BTS_PIDs BTS_PIDs_t;
```

BRICS
Basic Research
in Computer Science

SPIN, Gerald Holzmann AT&T



```
mtype = { msg0, msg1, ack0, ack1 };

chan    sender   =[1] of { byte };
chan    receiver=[1] of { byte };

proctype Sender()
{       byte any;
again:
        do
        :: receiver!msg1;
                if
                :: sender?ack1 -> break
                :: sender?any /* lost */
                :: timeout    /* retransmit */
                fi
        od;
        do
        :: receiver!msg0;
                if
                :: sender?ack0 -> break
                :: sender?any /* lost */
                :: timeout    /* retransmit */
                fi
        od;
        goto again
}

proctype Receiver()
{       byte any;
again:
        do
        :: receiver?msg1 -> sender!ack1; break
        :: receiver?msg0 -> sender!ack0
        :: receiver?any /* lost */
        od;
P0:
        do
        :: receiver?msg0 -> sender!ack0; break
```

```
<starting simulation>
/pack/FS/Spin.prog/spin-3.13/bin/spin -X -p -v -g -l -s -r -t -j0 pan_in

<at end of trail>
```

```
) line  41 "pan_in" (state 16)
 line  23 "pan_in" (state 16)
 line  50 "pan_in" (state 4)
ne   63 "never" (state 0)         [printf('MSC:
 line  63 "pan_in" (sta
```

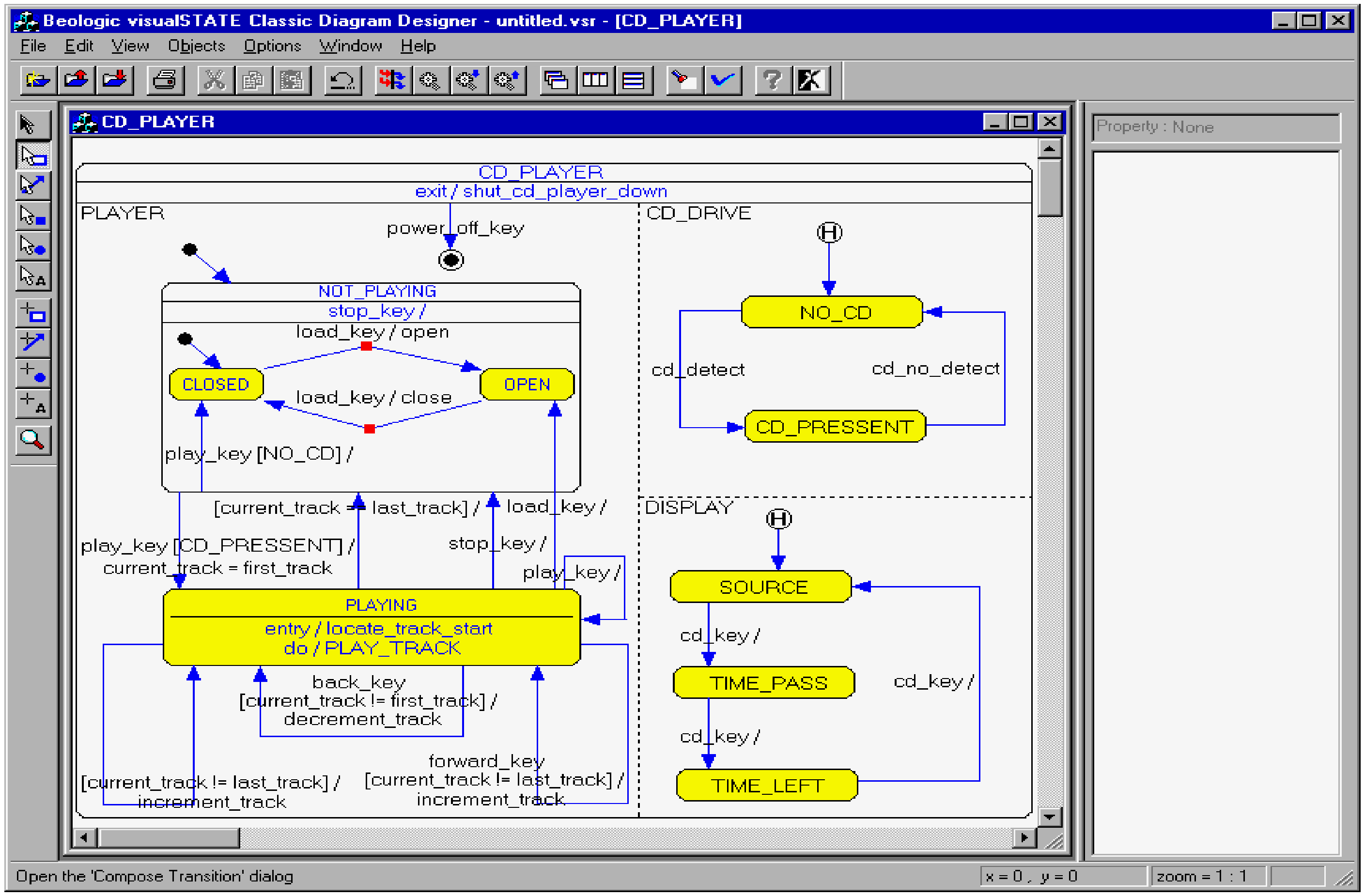
```
warning: for p.o. reduction to be valid the never claim must be stutter-closed
(never claims generated from LTL formulae are stutter-closed)
pan: acceptance cycle (at depth 59)
pan: wrote pan_in.trail
(Spin Version 3.1.3 -- 16 March 1998)
Warning: Search not completed
        + Partial Order Reduction

Full statespace search for:
        never-claim             +
        assertion violations    + (if within scope of claim)
        acceptance   cycles     + (fairness disabled)
        invalid endstates       - (disabled by never-claim)

State-vector 32 byte, depth reached 67, errors: 1
      35 states, stored (41 visited)
       6 states, matched
      47 transitions (= visited+matched)
       1 atomic steps
hash conflicts: 0 (resolved)
(max size 2^19 states)

2.542   memory usage (Mbyte)
```

visualSTATE

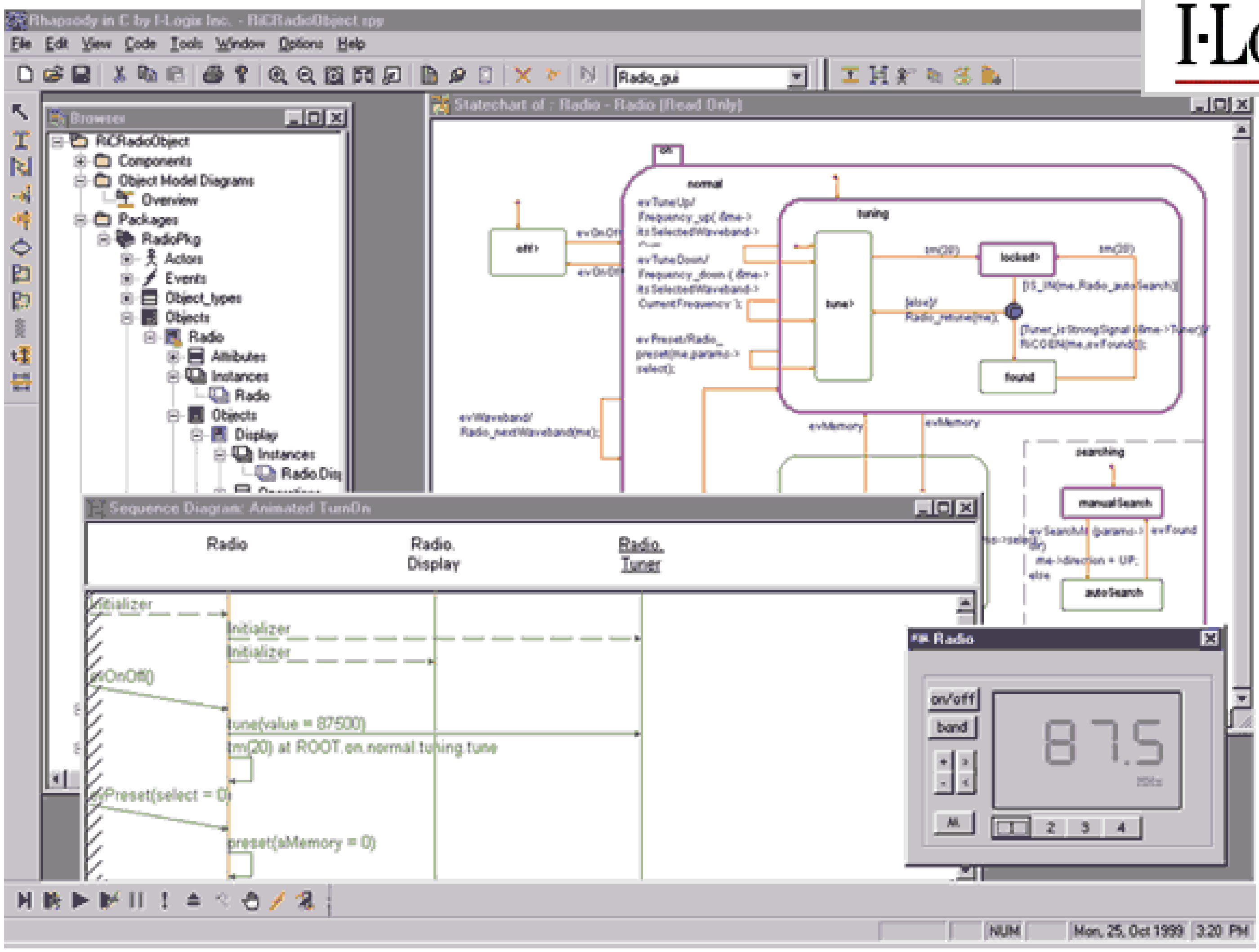
VVS

w Baan Visualstate, DTU (CIT project)

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

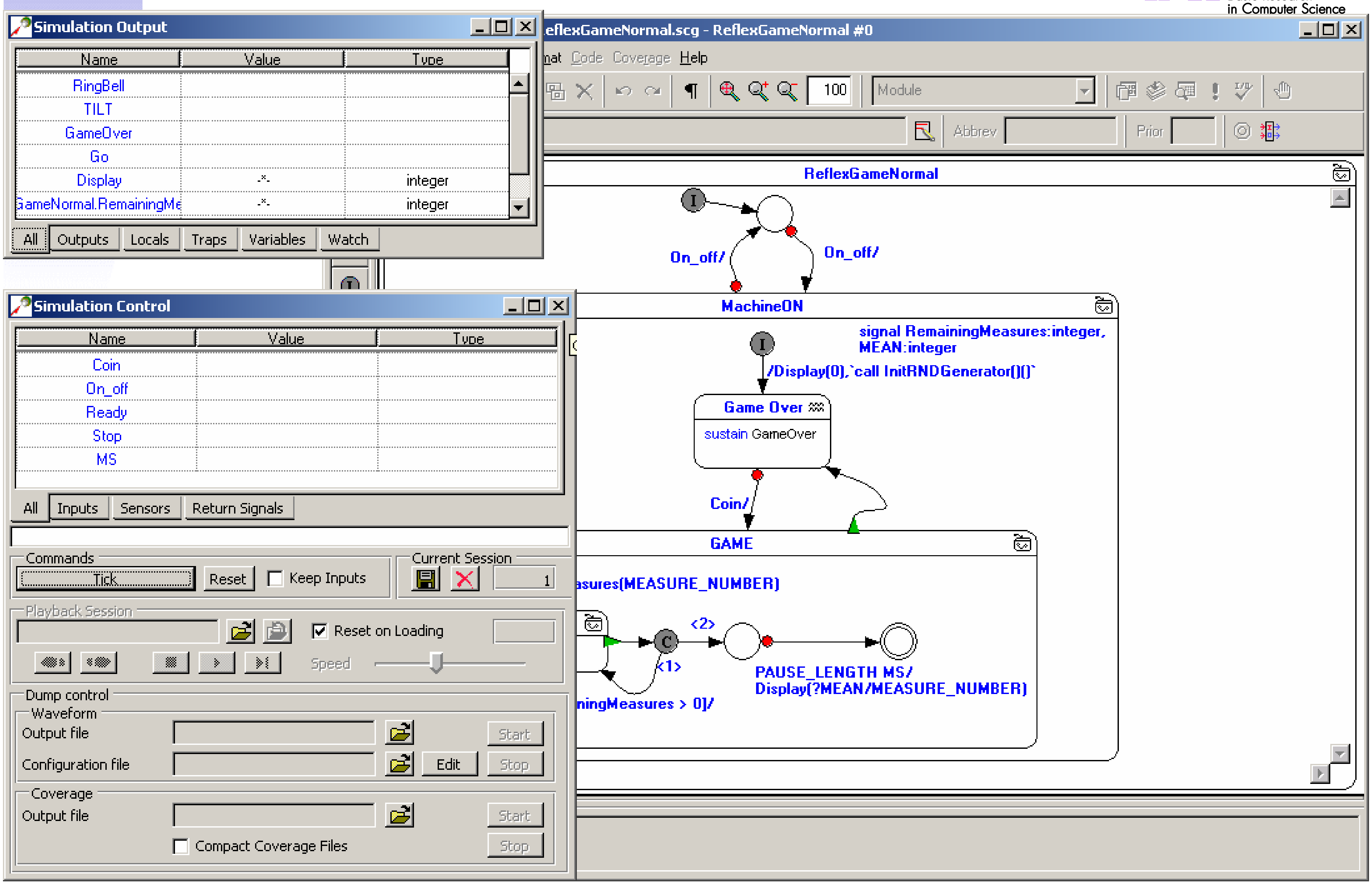


Rhapsody


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

ESTEREL

BRICS Basic Research in Computer Science



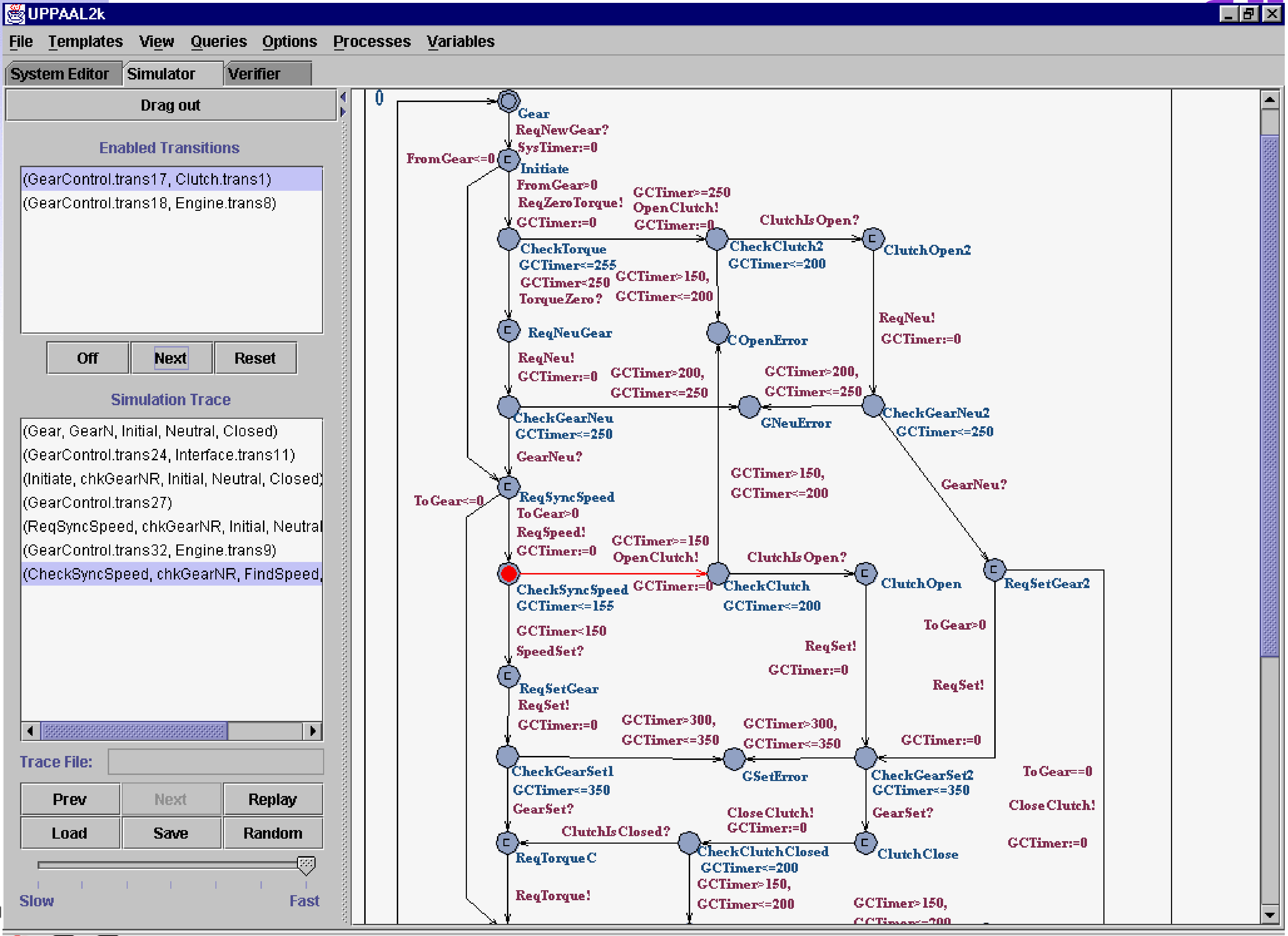
Simulation Output

Name	Value	Type
RingBell		
TILT		
GameOver		
Go		
Display	-*-	integer
GameNormal.RemainingMe	-*-	integer

Simulation Control

Name	Value	Type
Coin		
On_off		
Ready		
Stop		
MS		

UPPAAL



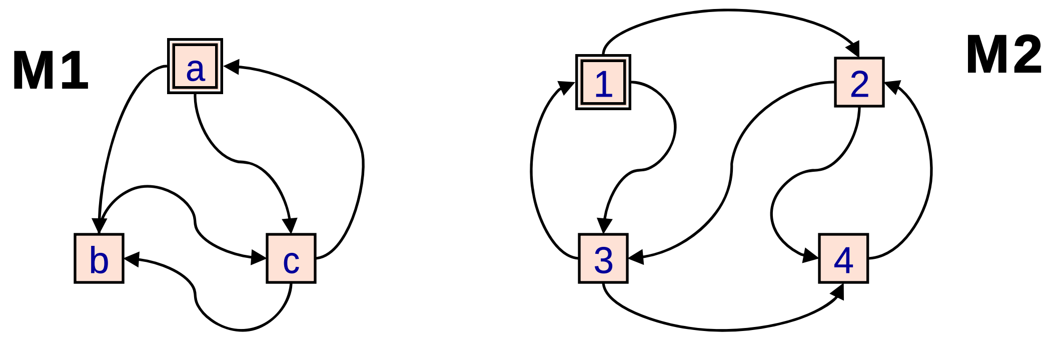
Enabled Transitions

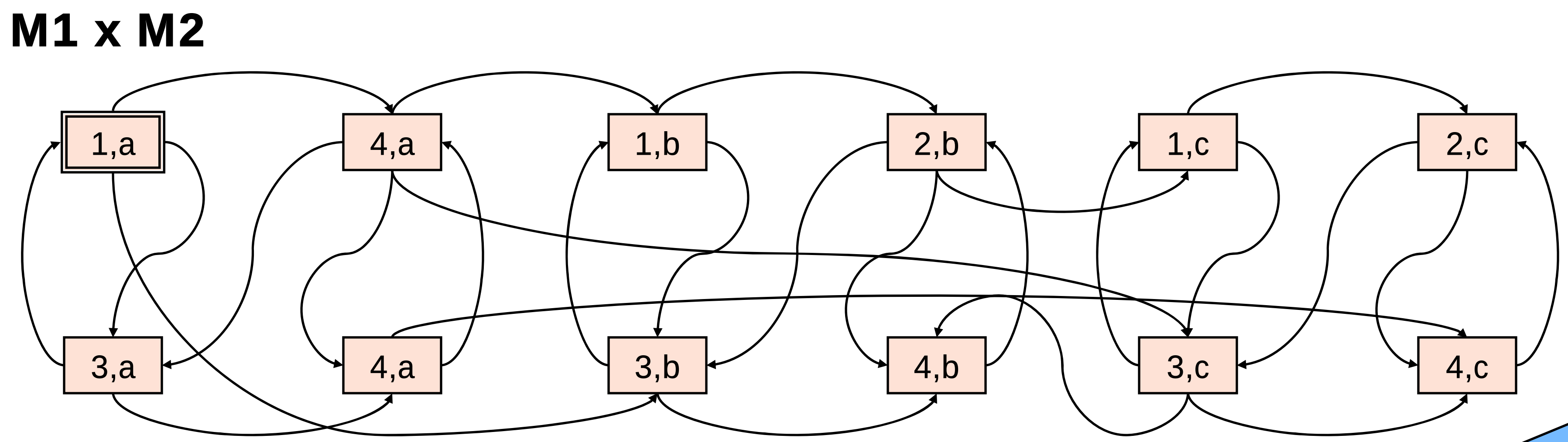
```
(GearControl.trans17, Clutch.trans1)
(GearControl.trans18, Engine.trans8)
```

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,
```

‘State Explosion’ problem





All combinations = exponential in no. of component

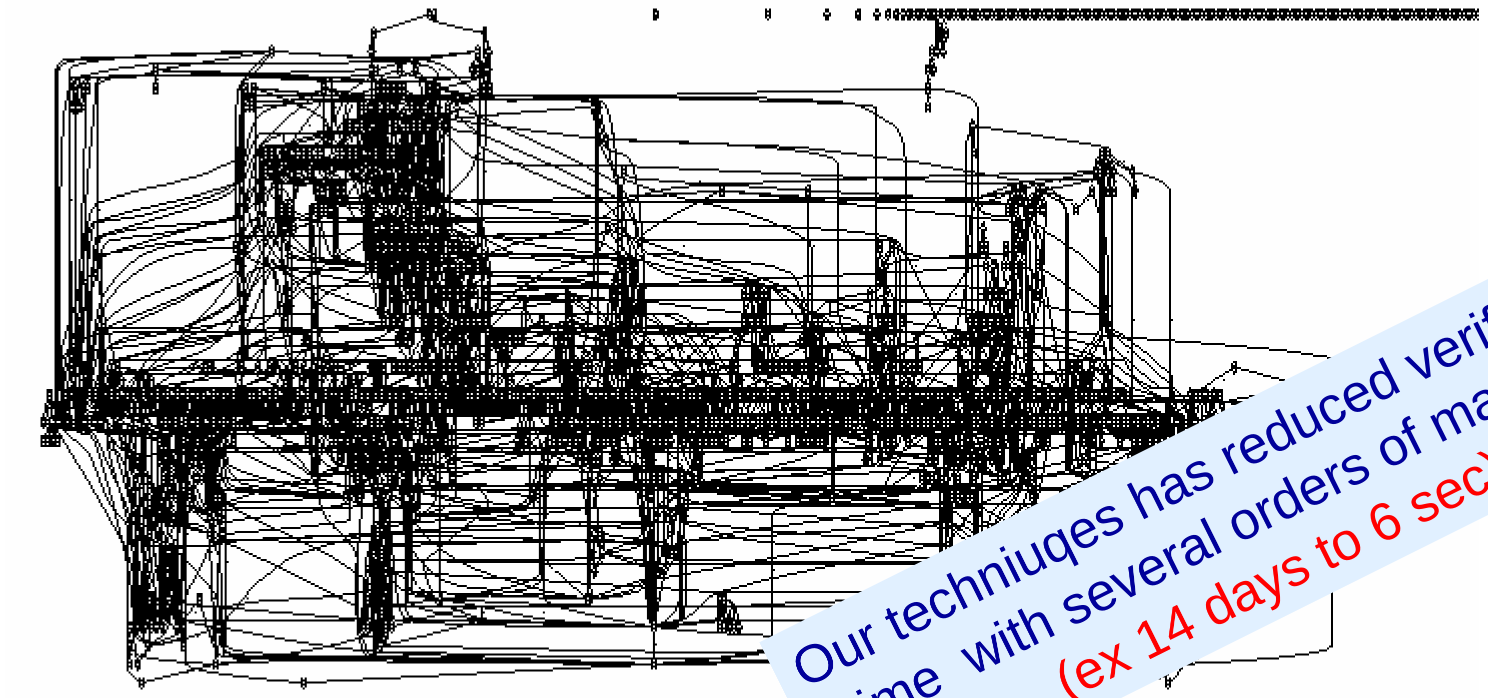
Provably theoretical intractable

Train Simulator

VVS
visualSTATE

1421 machines
11102 transitions
2981 inputs
2667 outputs
3204 local states
Declare state sp.: 10^476

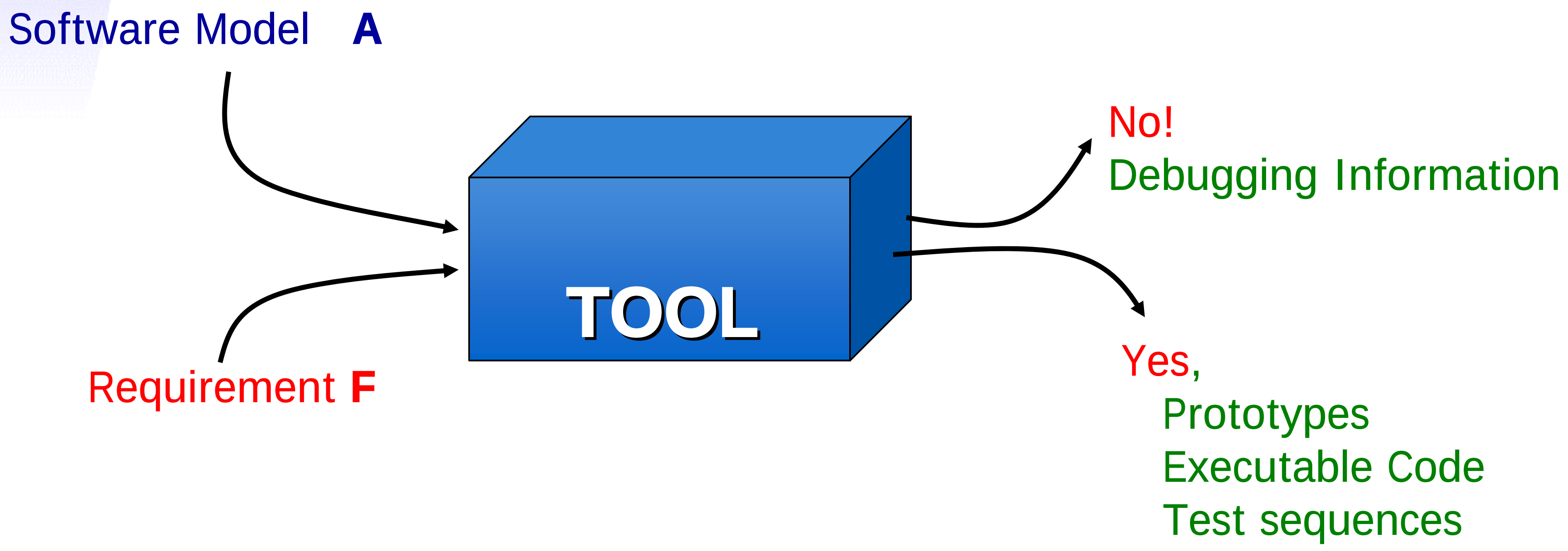
BUGS ?



Our techniuqes has reduced verification time with several orders of magnitude (ex 14 days to 6 sec)

BRICS Basic Research in Computer Science

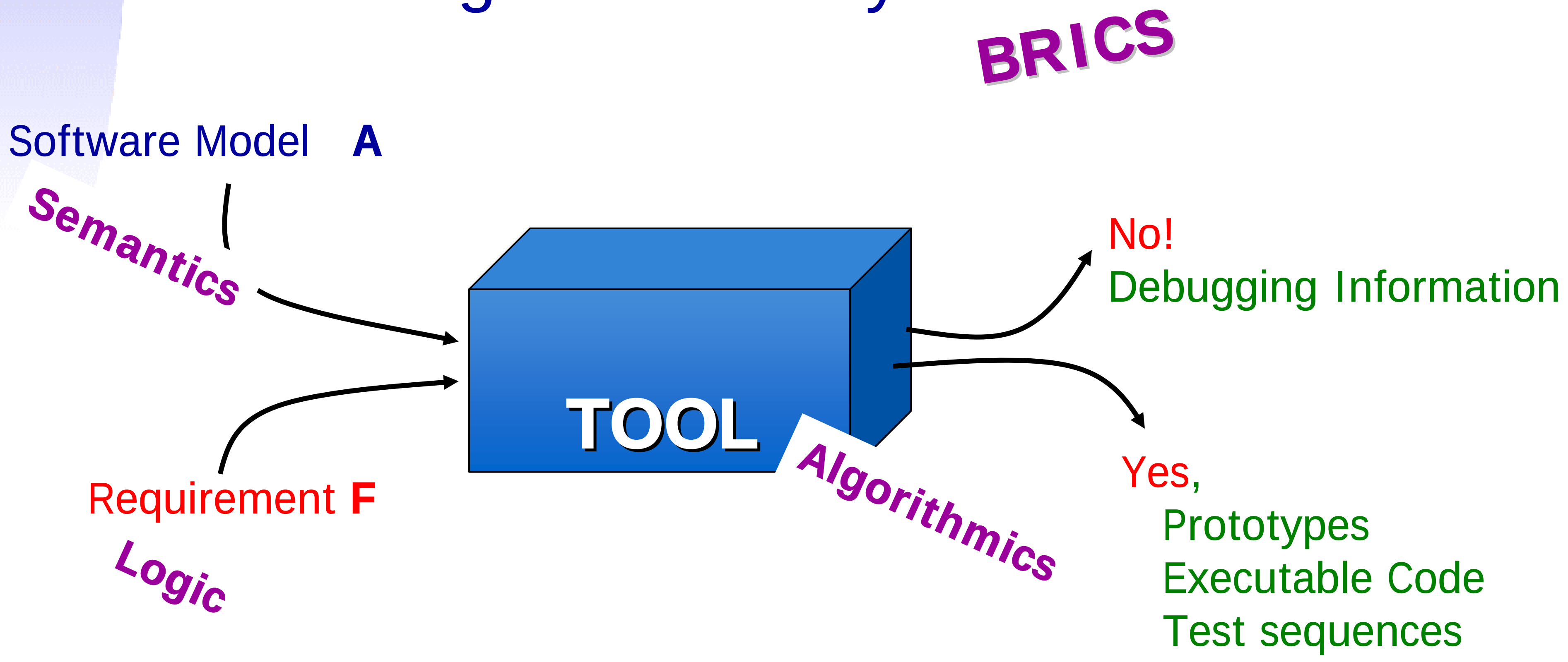
Modelling and Analysis



Tools: UPPAAL, visualSTATE,
ESTEREL, SPIN, Statemate, FormalCheck,
VeriSoft, Java Pathfinder,...

CISS CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

Modelling and Analysis



Tools: UPPAAL, visualSTATE,
ESTEREL, SPIN, Statemate, FormalCheck,
VeriSoft, Java Pathfinder,...

BRICS
Basic Research
in Computer Science

Most fundamentae model in Computer Science:
Kleene og Moore

Finite State Machines

- **Language versus behaviour**
- **Determinism versus non-determinism**
- **Composition and operations**
- **Variants of state machines**
  Moore, Mealy, IO automater, UML ....

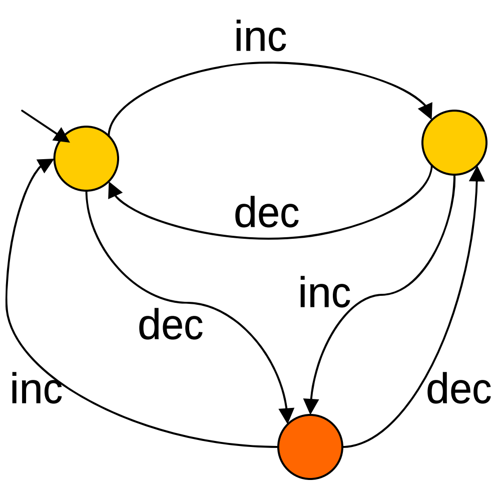
CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

State Machines

Model of Computation

- Set of states
- A start state
- An input-alfabet
- A transition funktion, mapping input symbols and state to next state
- One ore more accept states.
- Computation starts from start state with a given input string (read from left to right)

Modulo 3 counter



inc inc dec inc inc dec inc ☹

inc inc dec inc dec inc dec inc ☺

input string

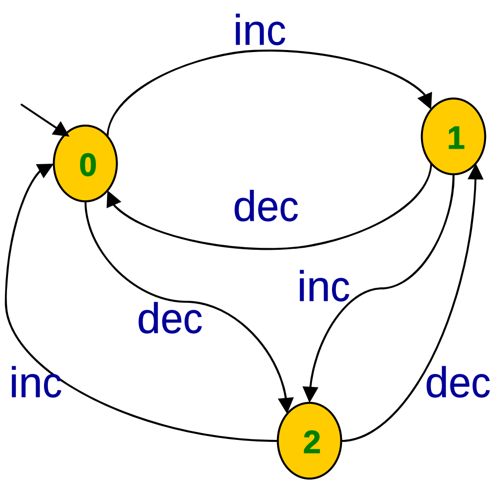
State Machines

Variants

Machines may have actions/output associated with state– Moore Machines.

inputstreng

inc inc dec inc inc dec inc



0 1 2 1 2 0 2 1

outputstreng

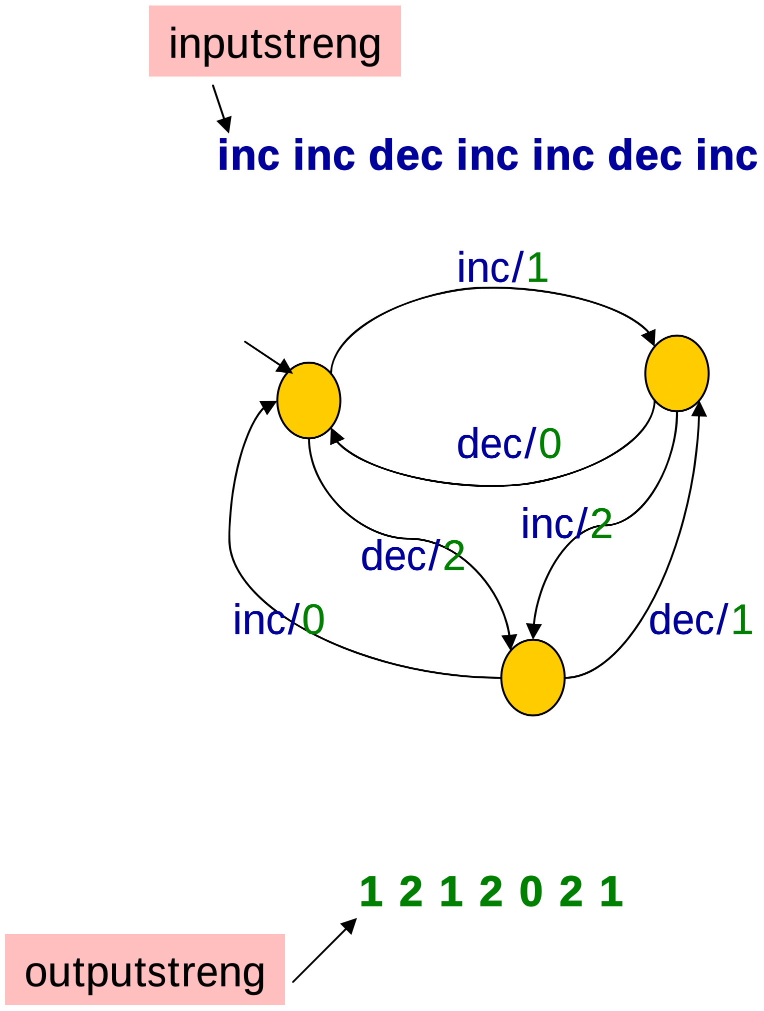
State Machines

Varianter

Machines may have actions/output associated with med transitions – Mealy Maskiner.

Transitions unconditional of af input (nul-transitions).

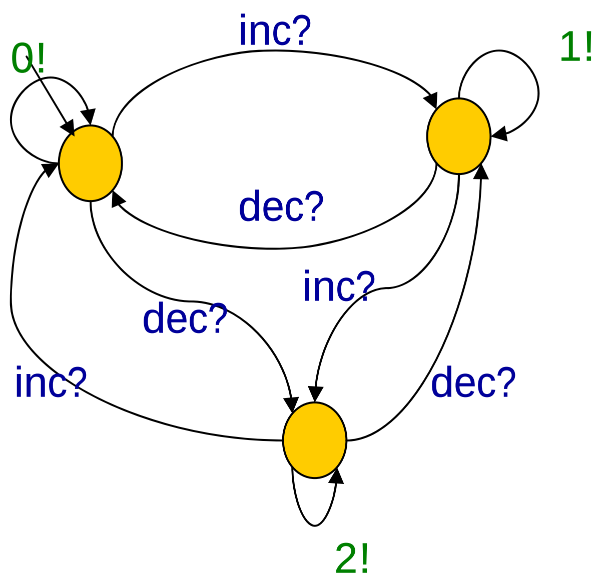
Several transitions for given for input and state (non-determinisme).



State Machines

Variants

Symbols of alphabet patitioned in input- and output-actions (IO-automata)



0! 0! 0! inc? inc? 2! 2! dec? 1!

interaction

Bankbokskode

🟠 O
🔵 B
🟡 G

To open a bank box
the code most contain at least 2 🔵

To open a bank box
the code most end with 🟠 🔵 🟠

To open a bank box
the code most end with 🟠 🔵 🟠
or with 🟡 🔵 🟠

To open a bank box
the code most end with a palindrom
e.g:. 🟠 🔵 🟠
🟡 🟠 🟠 🟡
🟡 🟠 🔵 🟠 🟡
....

Fundamental Results

- Every FSM may be determinized accepting the same language (potential explosion in size).
- For each FSM there exist a language-equivalent *minimal* deterministic FSM.
- FSM's are closed under $\cap$ and $\cup$
- FSM's may be described as regular expressions (and vise versa)

Interacting State Machines

Home-Banking?

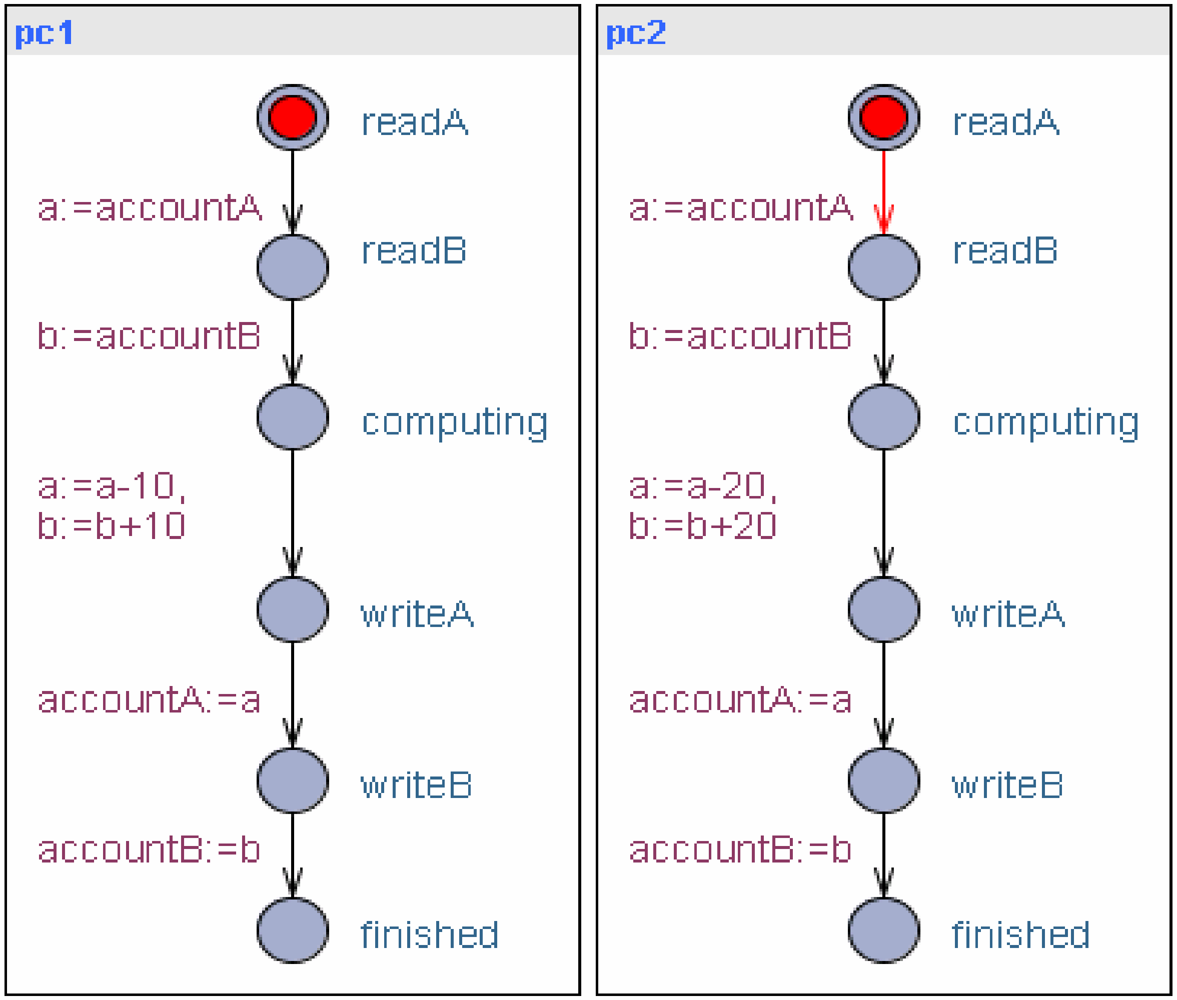
```
int accountA, accountB; //Shared global variables
//Two concurrent bank costumers
```

```
Thread costumer1 () {
  int a,b; //local tmp copy

  a=accountA;
  b=accountB;
  a=a-10;b=b+10;
  accountA=a;
  accountB=b;
}
```

```
Thread costumer2 () {
  int a,b;

  a=accountA;
  b=accountB;
  a=a-20; b=b+20;
  accountA=a;
  accountB=b;
}
```

- Are the accounts in balance after the transactions?

Home Banking



A[] (pc1.finished and pc2.finished) imply (accountA+accountB==200)?

Home Banking

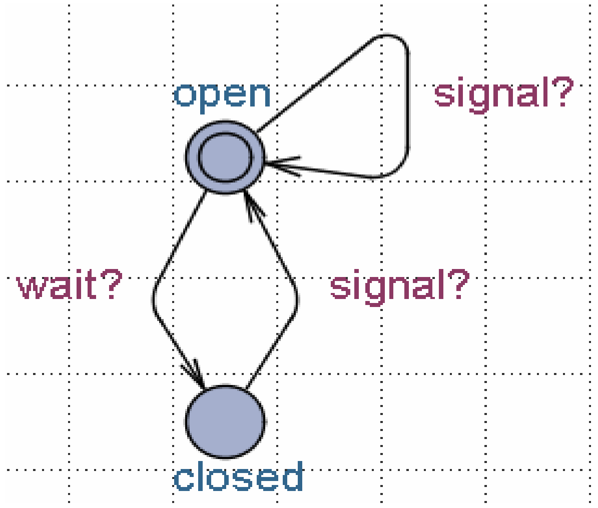
```
int accountA, accountB; //Shared global variables
Semaphore A,B;          //Protected by sem A,B
//Two concurrent bank costumers
```

```
Thread costumer1 () {
  int a,b; //local tmp copy

  wait(A);
  wait(B);
  a=accountA;
  b=accountB;
  a=a-10;b=b+10;
  accountA=a;
  accountB=b;
  signal(A);
  signal(B);
}
```

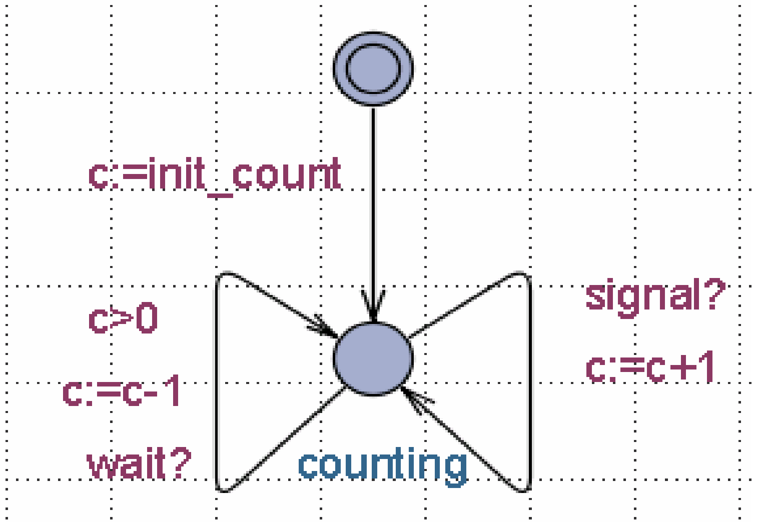
```
Thread costumer2 () {
  int a,b;

  wait(B);
  wait(A);
  a=accountA;
  b=accountB;
  a=a-20; b=b+20;
  accountA=a;
  accountB=b;
  signal(B);
  signal(A);
}
```

Semaphore FSM Model

Binary Semaphore

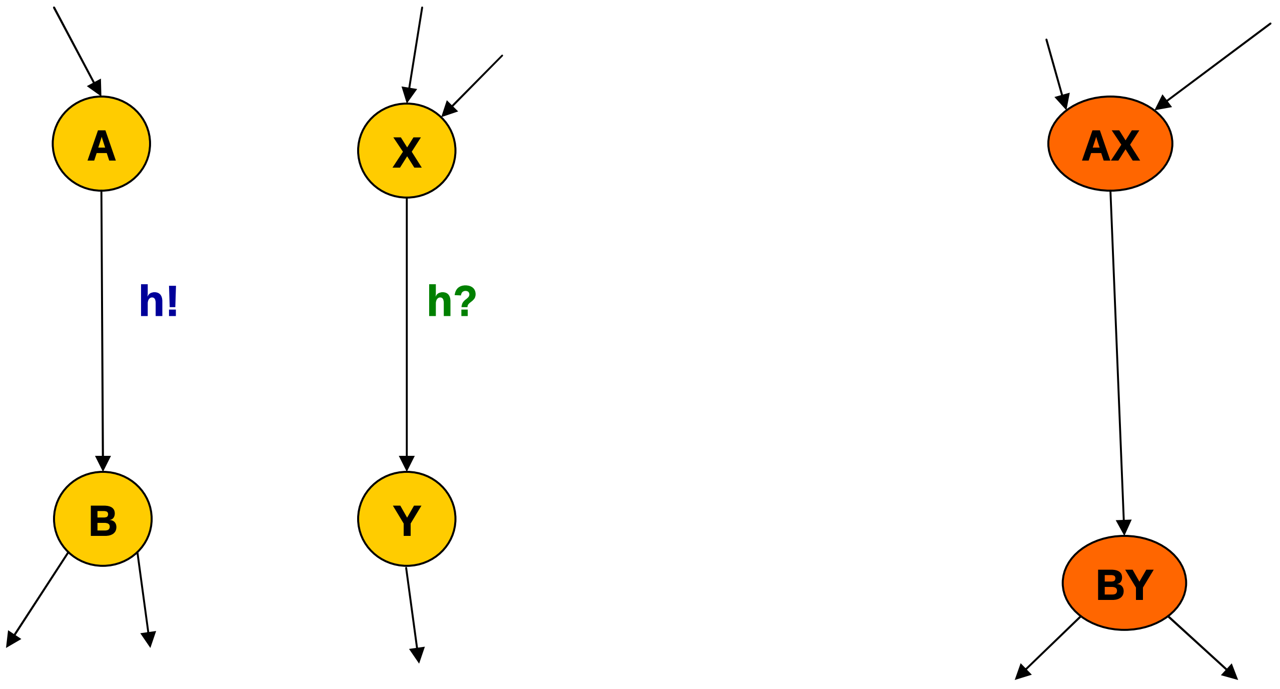


Counting Semaphore



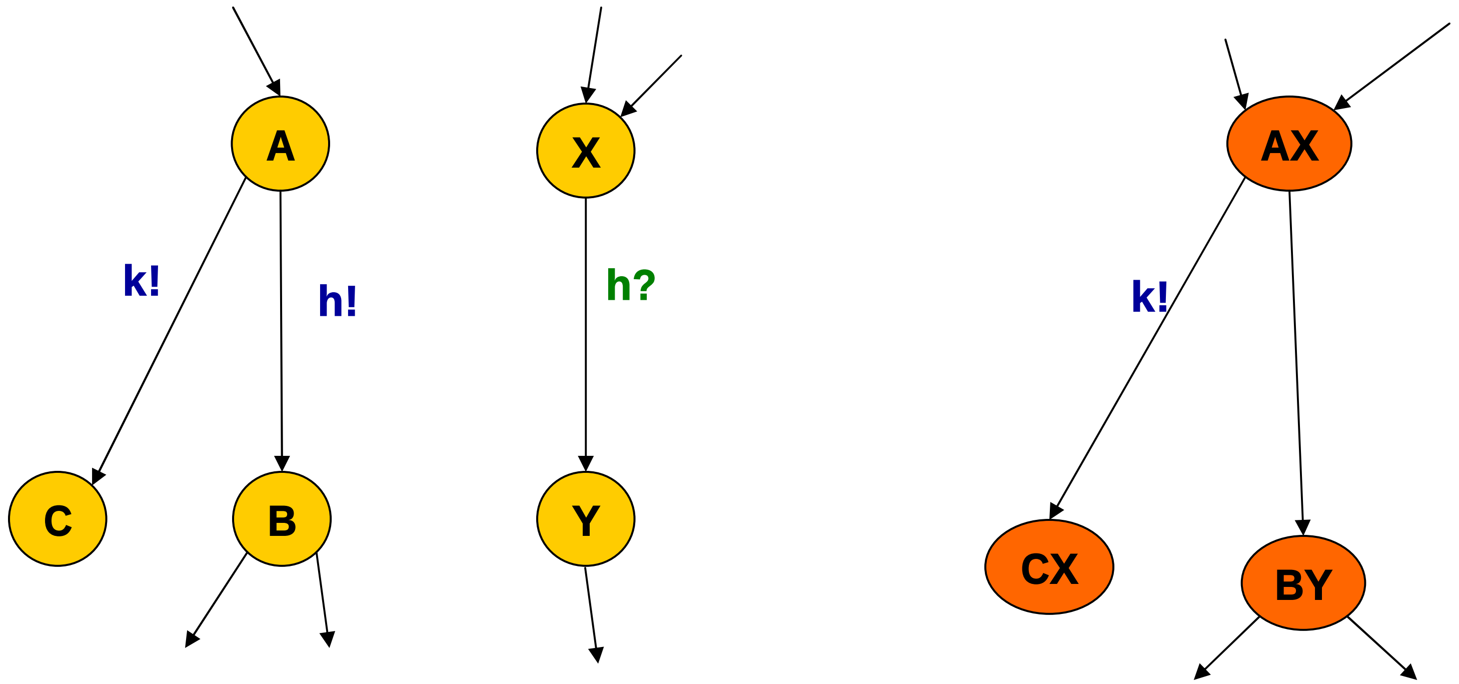
Composition

IO Automater (2-vejs synkronisering)

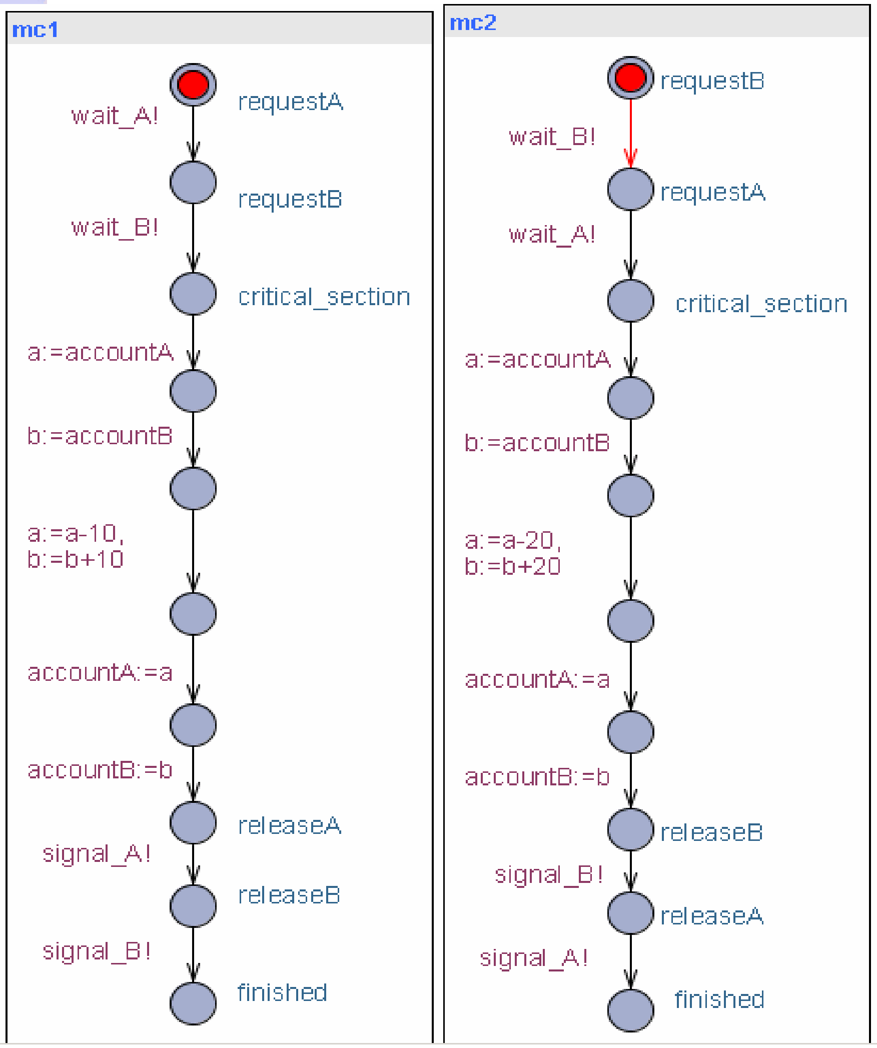


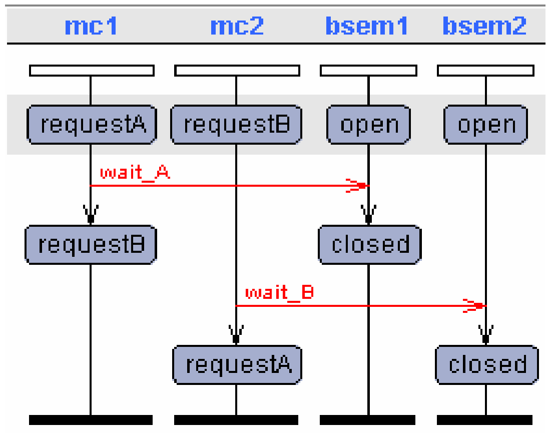
Composition

IO Automater



Semaphore Solution?



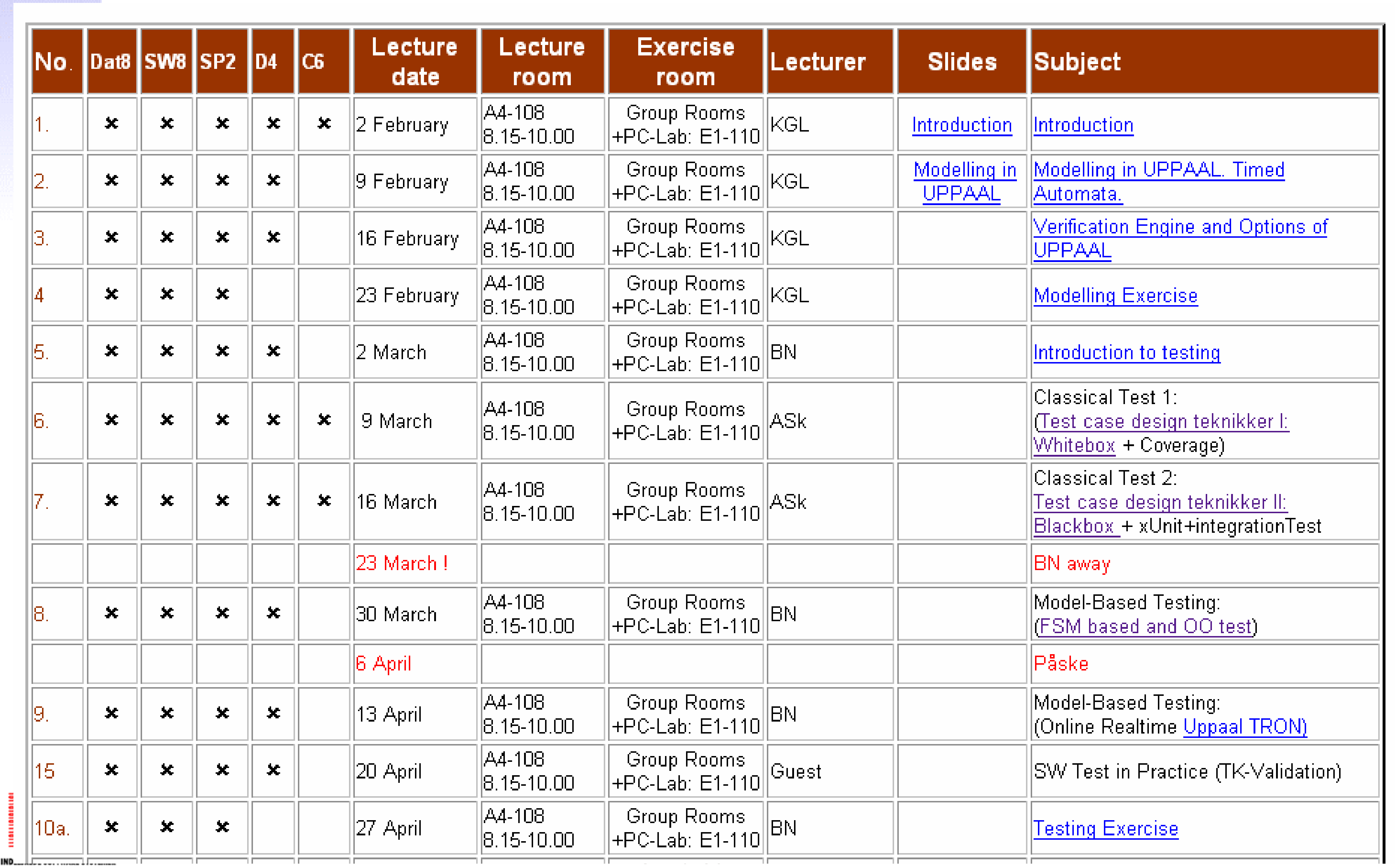


1. Consistency? (Balance)
2. Race conditions?
3. Deadlock?

1. A[] (mc1.finished and mc2.finished) imply (accountA+accountB==200) ✓
2. E<> mc1.critical_section and mc2.critical_section ✓
3. A[] not (mc1.finished and mc2.finished) imply not deadlock ÷

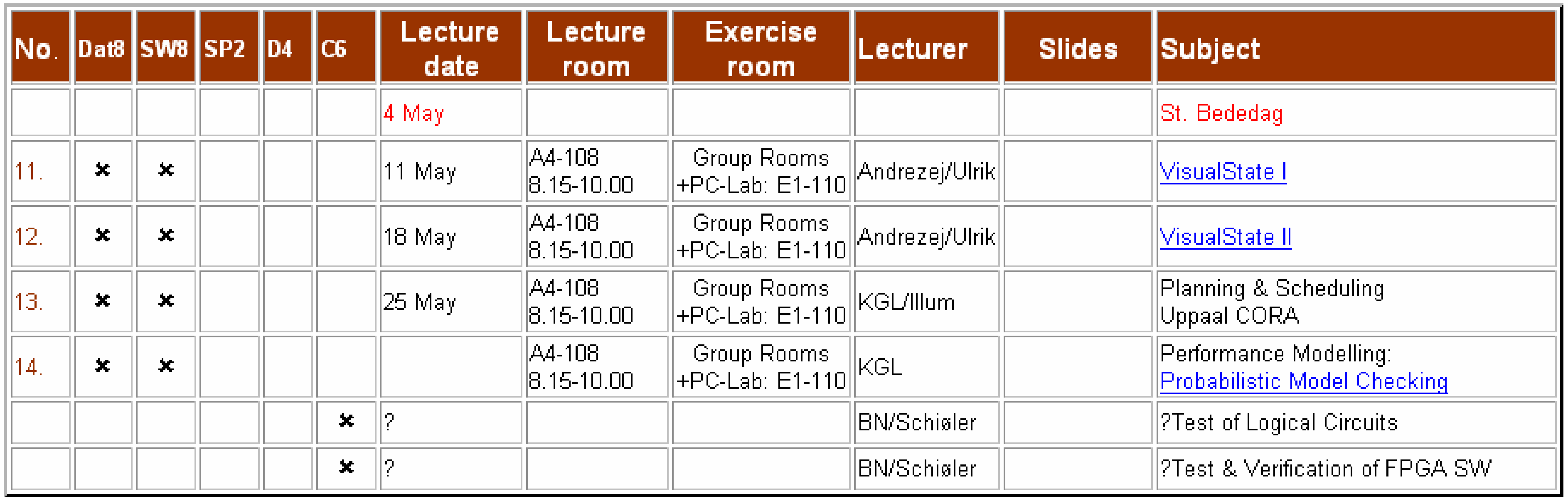
BRICS
Basic Research
in Computer Science

Plan for kursus



No.	Dat8	SW8	SP2	D4	C6	Lecture date	Lecture room	Exercise room	Lecturer	Slides	Subject
1.	×	×	×	×	×	2 February	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	KGL	Introduction	Introduction
2.	×	×	×	×		9 February	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	KGL	Modelling in UPPAAL	Modelling in UPPAAL. Timed Automata.
3.	×	×	×	×		16 February	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	KGL		Verification Engine and Options of UPPAAL
4	×	×	×			23 February	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	KGL		Modelling Exercise
5.	×	×	×	×		2 March	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	BN		Introduction to testing
6.	×	×	×	×	×	9 March	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	ASk		Classical Test 1: (Test case design teknikker I: Whitebox + Coverage)
7.	×	×	×	×	×	16 March	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	ASk		Classical Test 2: Test case design teknikker II: Blackbox + xUnit+integrationTest
						23 March !					BN away
8.	×	×	×	×		30 March	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	BN		Model-Based Testing: (FSM based and OO test)
						6 April					Påske
9.	×	×	×	×		13 April	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	BN		Model-Based Testing: (Online Realtime Uppaal TRON)
15	×	×	×	×		20 April	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	Guest		SW Test in Practice (TK-Validation)
10a.	×	×	×			27 April	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	BN		Testing Exercise

Plan for kursus



No.	Dat8	SW8	SP2	D4	C6	Lecture date	Lecture room	Exercise room	Lecturer	Slides	Subject
						4 May					St. Bededag
11.	×	×				11 May	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	Andrezej/Ulrik		VisualState I
12.	×	×				18 May	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	Andrezej/Ulrik		VisualState II
13.	×	×				25 May	A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	KGL/Illum		Planning & Scheduling Uppaal CORA
14.	×	×					A4-108 8.15-10.00	Group Rooms +PC-Lab: E1-110	KGL		Performance Modelling: Probabilistic Model Checking
					×	?			BN/Schiøler		?Test of Logical Circuits
					×	?			BN/Schiøler		?Test & Verification of FPGA SW