Presentation at Microsoft Research, April 10, 2003

Verification Case Studies with ObjectCheck

Fei Xie

Department of Computer Sciences

The University of Texas at Austin

(Joint work with James C. Browne, Robert P. Kurshan, and Vladimir Levin)

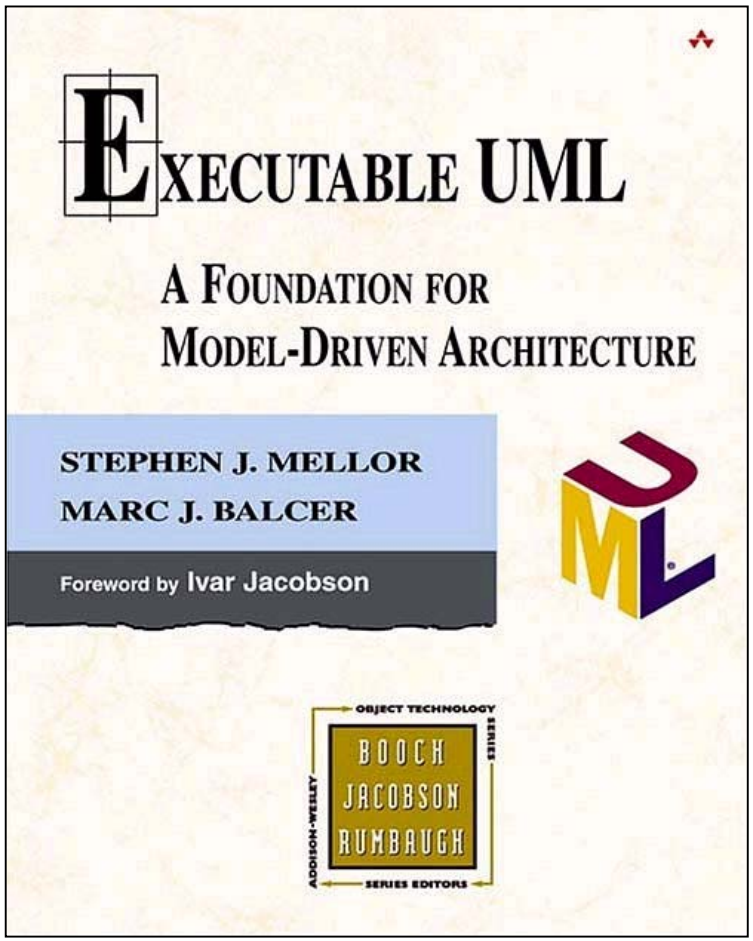
Presentation Outline

- Overview and Architecture of ObjectCheck
- Modeling and Verification of a TinyOS Run-time Image with ObjectCheck
- More Case Studies
- Summary and Future Work
- Work Built on ObjectCheck

ObjectCheck Overview

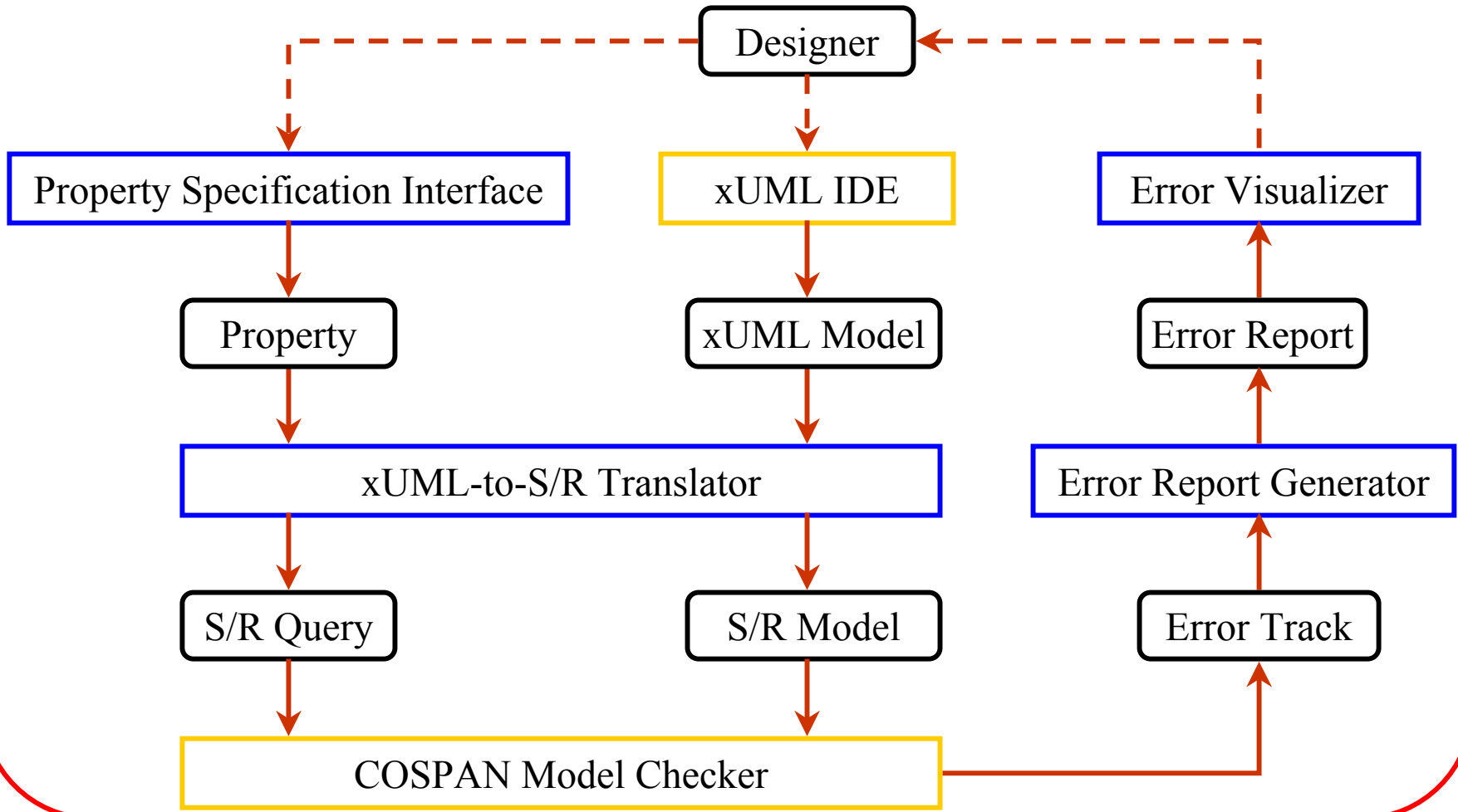
- An integrated development, validation, and mode-checking environment for software system designs;
  - System designs are specified in xUML;
  - xUML is an executable subset of UML;
- Developed in conjunction with
  - FormalCheck (Robert P. Kurshan, et. al.);
  - SDLCheck (Vladimir Levin and Husnu Yenigun);
  - Goal: Software/hardware co-design and co-verification;
    - Sub-goal: Model checking of software system designs in xUML.

Executable UML (xUML)



- Has well-defined *Execution Semantics;*
- Utilizes *UML Action Semantics* recently adopted by OMG;
- Can be compiled to procedural codes;
- Tools provided by:
  - Project Technologies;
  - Kennedy Carter;
  - Hyperformix (SES);
  - …

Architecture and Workflow of ObjectCheck



Presentation Outline

- Overview and Architecture of ObjectCheck
- Modeling and Verification of a TinyOS Run-time Image with ObjectCheck
- More Case Studies
- Summary and Future Work
- Work Built on ObjectCheck

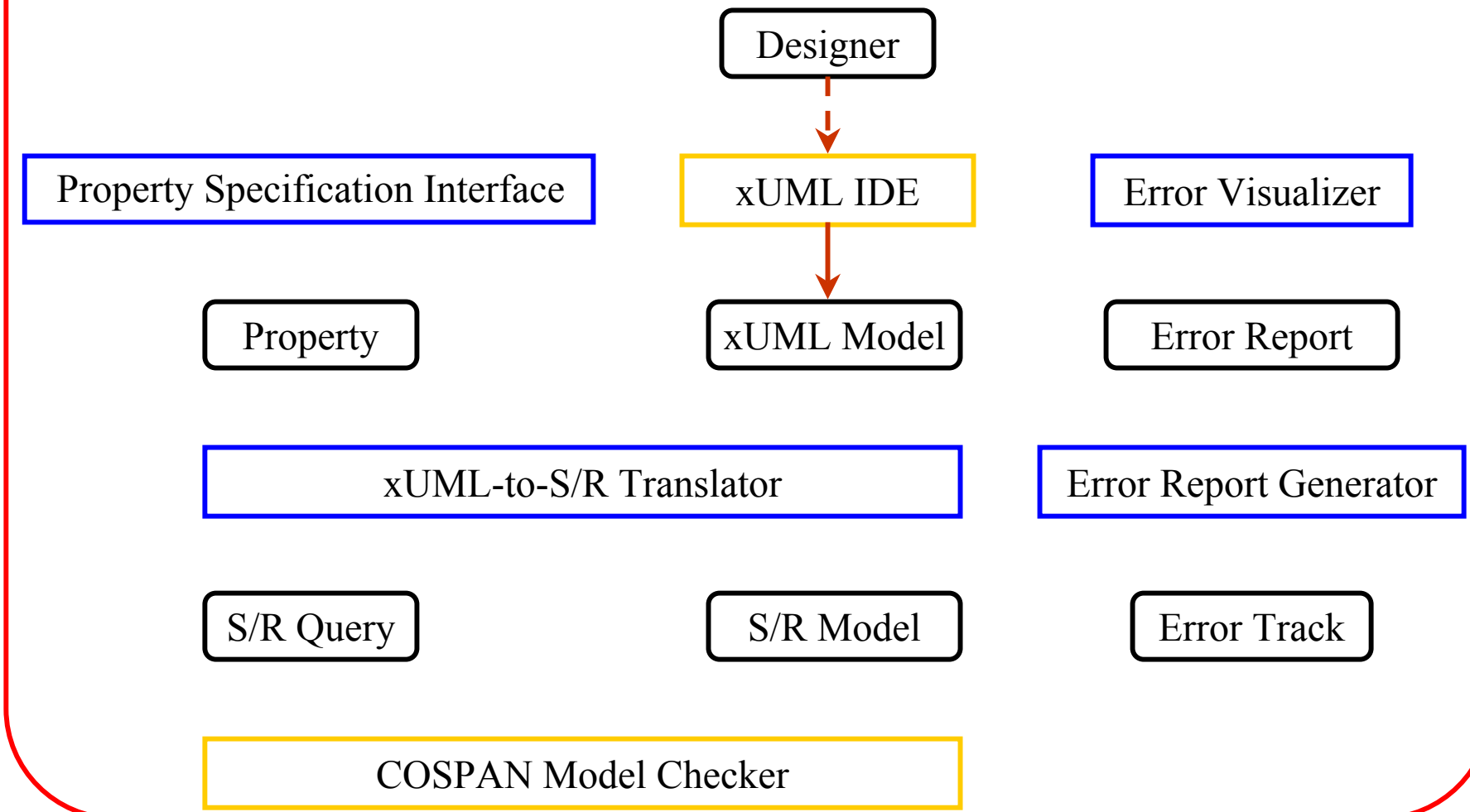
Case Study on TinyOS

- A run-time image of TinyOS, a component-based operating system for networked sensors;
- xUML models for TinyOS components have been constructed from their C source codes to enable:
  - Model-driven development on design level;
  - Software/hardware co-design and co-verification;
- Two properties are to be checked:
  - *Repeated transmission* on physical network;
  - *No consecutive 0's* in any sequence-number sequence.

More on TinyOS

- An OS for networked sensors that have
  - Limited computation ability and energy supply;
  - High concurrency requirements;
  - Diversities in designs and usages;
  - High reliability requirement;
- Is component-based
  - Run-time images are specialized for different sensors;
  - Only necessary components are selected and composed.
- More information -- http://webs.cs.berkeley.edu/tos

Step-by-Step Demonstration



Objectbench

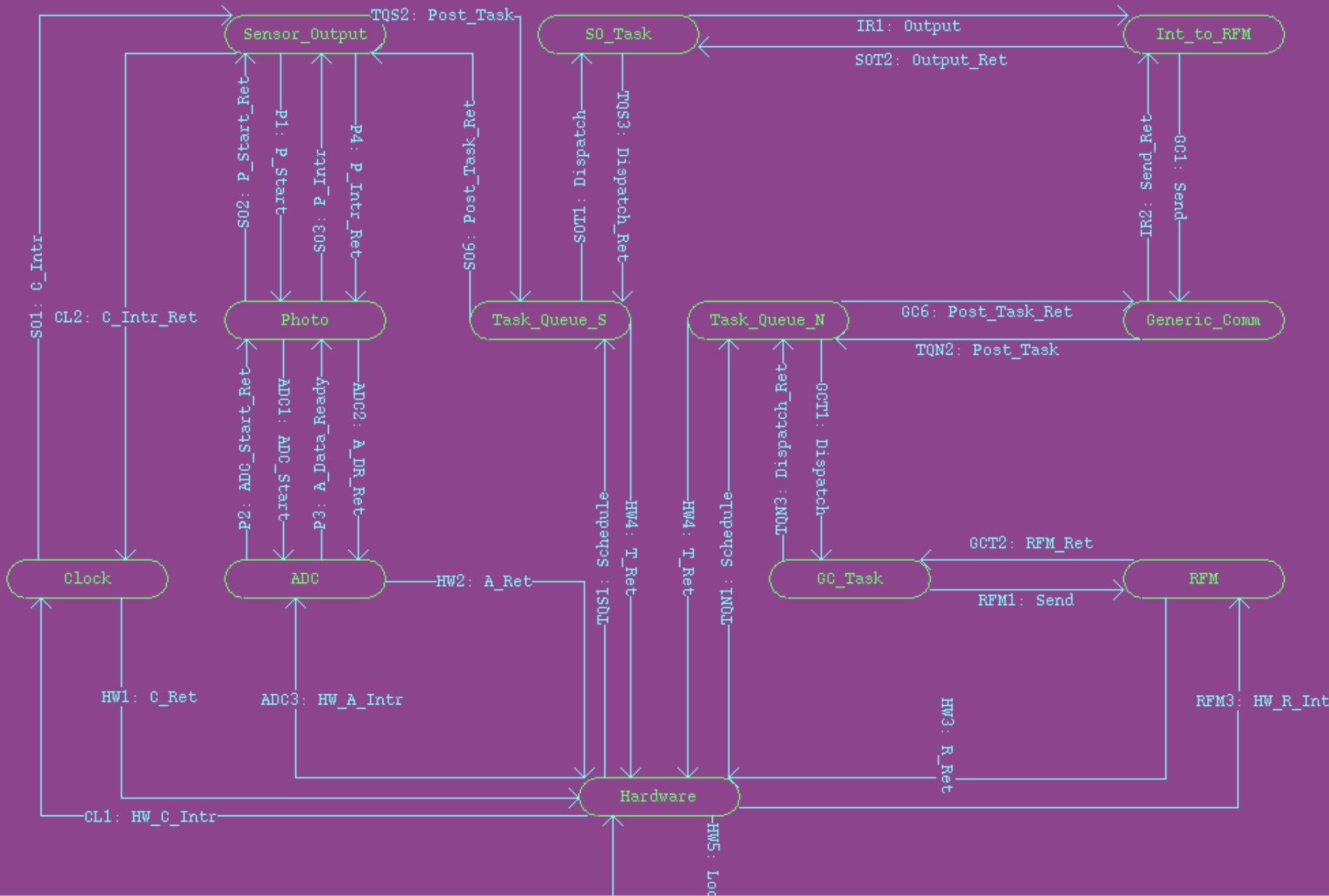
Objectbench 3.1

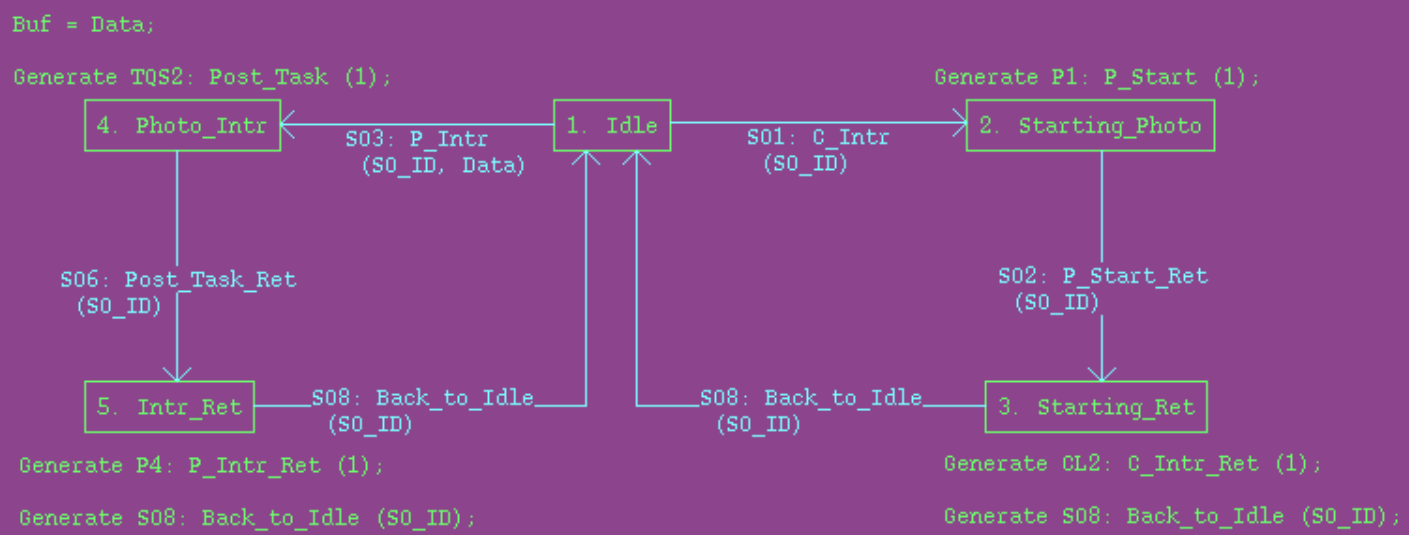
Index | What | How

Subsystem: Sensor_to_Network

Location: OCM

Close

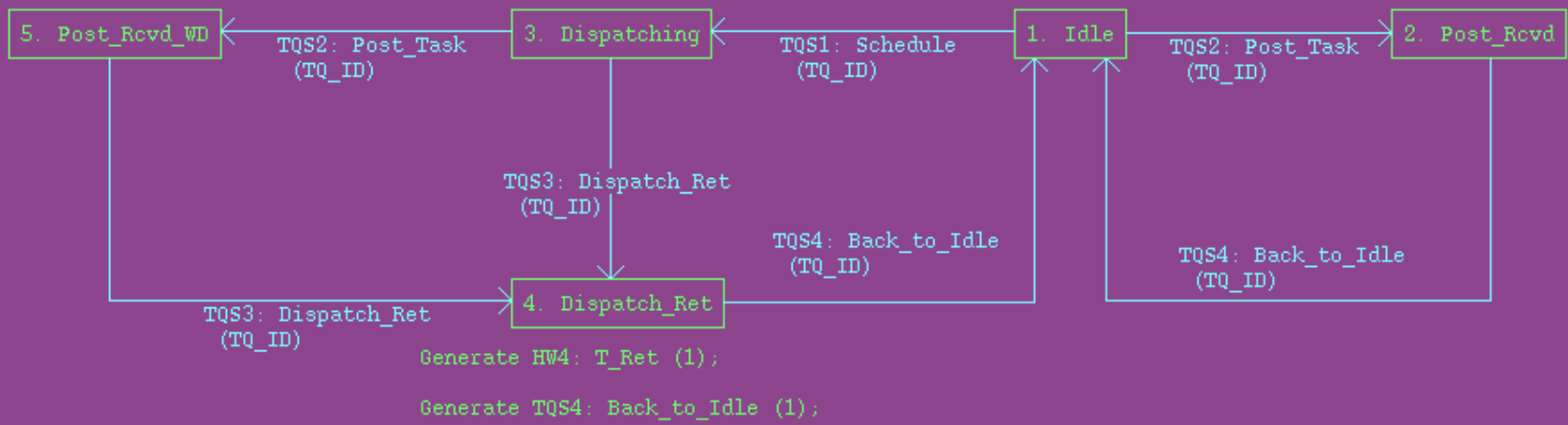


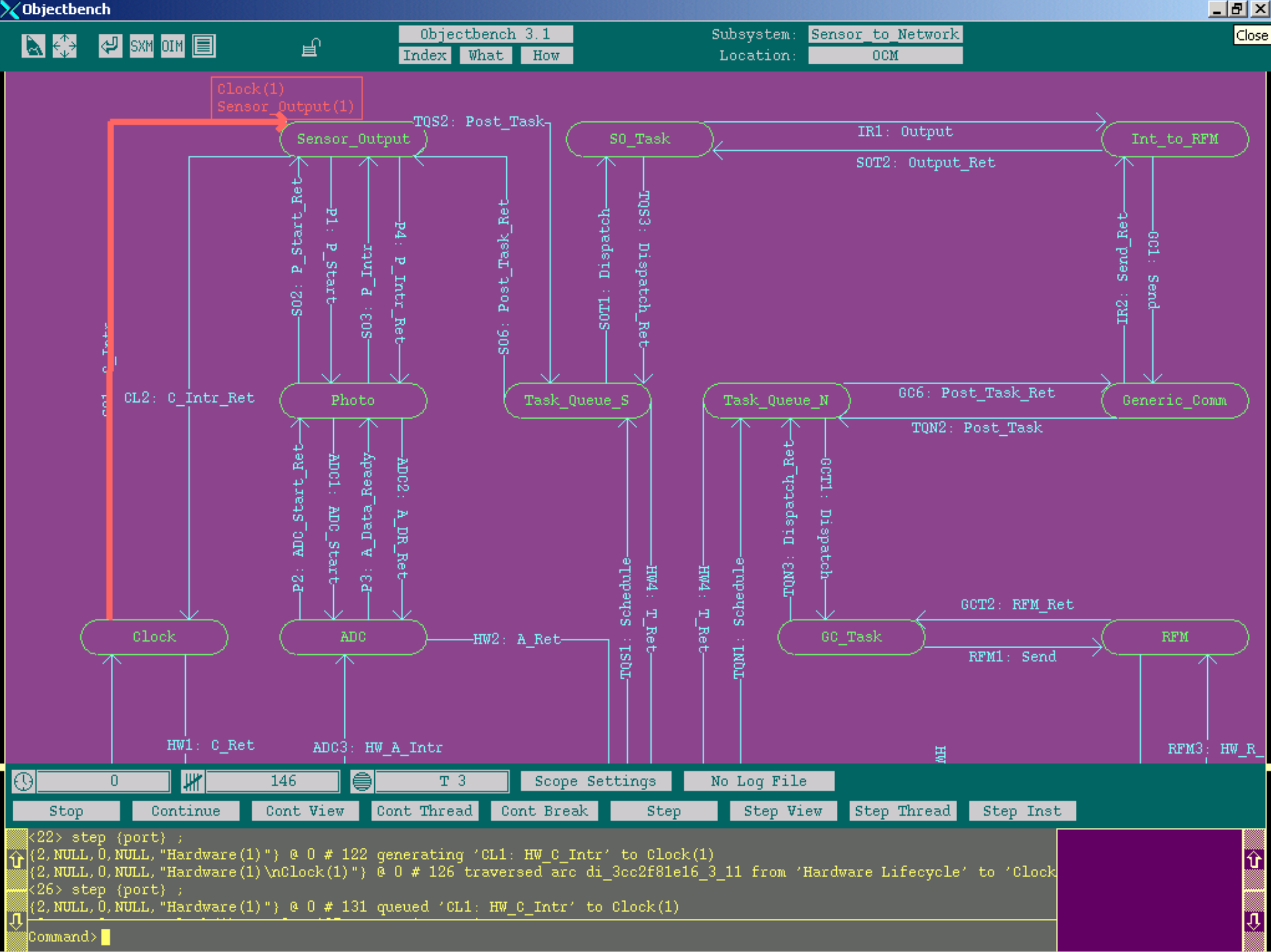

Buf = Data;
Generate TQS2: Post_Task (1);
Generate P1: P_Start (1);
4. Photo_Intr
1. Idle
2. Starting_Photo
S03: P_Intr
(SO_ID, Data)
S01: C_Intr
(SO_ID)
S06: Post_Task_Ret
(SO_ID)
S02: P_Start_Ret
(SO_ID)
5. Intr_Ret
S08: Back_to_Idle
(SO_ID)
S08: Back_to_Idle
(SO_ID)
3. Starting_Ret
Generate P4: P_Intr_Ret (1);
Generate S08: Back_to_Idle (SO_ID);
Generate CL2: C_Intr_Ret (1);
Generate S08: Back_to_Idle (SO_ID);

```
if (Full == FALSE) {
 if (Tail == 5) Tail = 0;
  else Tail = Tail + 1;

 if (Emp == TRUE) Emp = FALSE;

 if (Tail == Head) Full = TRUE;
}

Generate S06: Post_Task_Ret (1);
```

```
if (Emp == FALSE) {
 Generate SOT1: Dispatch (1);

 if (Head == 5) Head = 0;
  else Head = Head + 1;

 if (Full == TRUE) Full = FALSE;

 if (Head == Tail) Emp = TRUE;
}
```

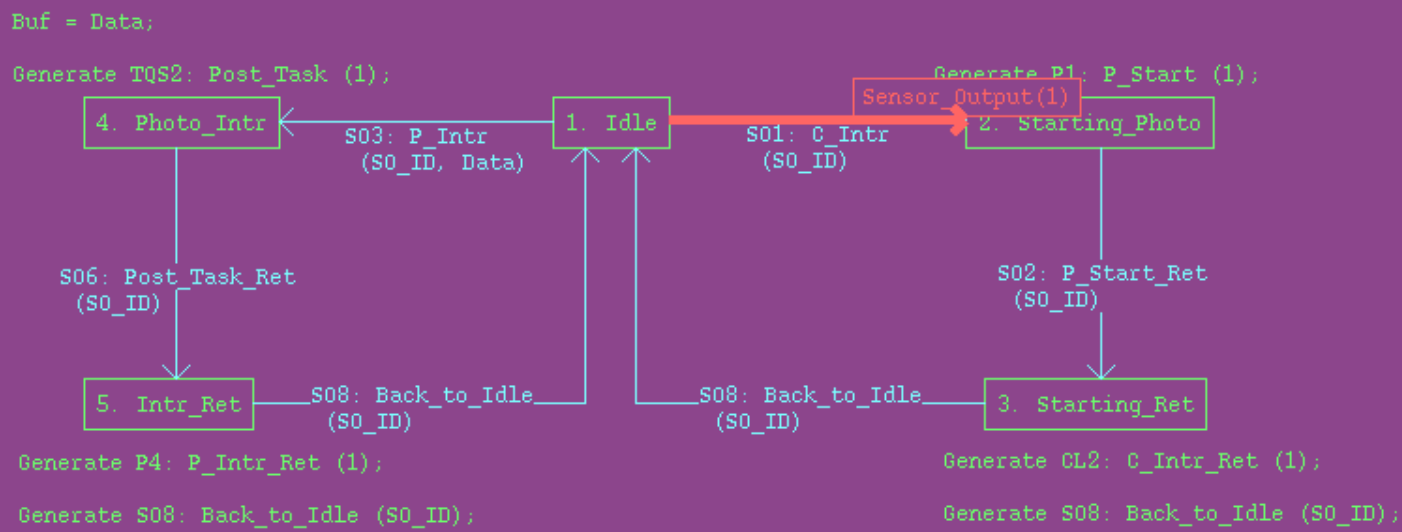
```
if (Full == FALSE) {
 if (Tail == 5) Tail = 0;
   else Tail = Tail + 1;

 if (Emp == TRUE) Emp = FALSE;

 if (Tail == Head) Full = TRUE;
}

Generate S06: Post_Task_Ret (1);
Generate TQS4: Back_to_Idle (1);
```




Objectbench
Objectbench 3.1
Index
What
How
Subsystem: Sensor_to_Network
Location: OCM
Close
Clock(1)
Sensor_Output(1)
Sensor_Output
SO_Task
Int_to_RFM
Photo
Task_Queue_S
Task_Queue_N
Generic_Comm
Clock
ADC
GC_Task
RFM
TQS2: Post_Task
IR1: Output
SOT2: Output_Ret
SO2: P_Start_Ret
P1: P_Start
SO3: P_Intr
P4: P_Intr_Ret
SO6: Post_Task_Ret
SOT1: Dispatch
TQS3: Dispatch_Ret
IR2: Send_Ret
GC1: Send
CL2: C_Intr_Ret
GC6: Post_Task_Ret
TQN2: Post_Task
P2: ADC_Start_Ret
ADC1: ADC_Start
P3: A_Data_Ready
ADC2: A_DR_Ret
TQS1: Schedule
HW4: T_Ret
TQN3: Dispatch_Ret
GCT1: Dispatch
GCT2: RFM_Ret
RFM1: Send
HW2: A_Ret
HW1: C_Ret
ADC3: HW_A_Intr
RFM3: HW_R
0
146
T 3
Scope Settings
No Log File
Stop
Continue
Cont View
Cont Thread
Cont Break
Step
Step View
Step Thread
Step Inst
<22> step {port} ;
(2,NULL,0,NULL,"Hardware(1)") @ 0 # 122 generating 'CL1: HW_C_Intr' to Clock(1)
(2,NULL,0,NULL,"Hardware(1)\nClock(1)") @ 0 # 126 traversed arc di_3cc2f81e16_3_11 from 'Hardware Lifecycle' to 'Clock
<26> step {port} ;
(2,NULL,0,NULL,"Hardware(1)") @ 0 # 131 queued 'CL1: HW_C_Intr' to Clock(1)
Command>

Objectbench

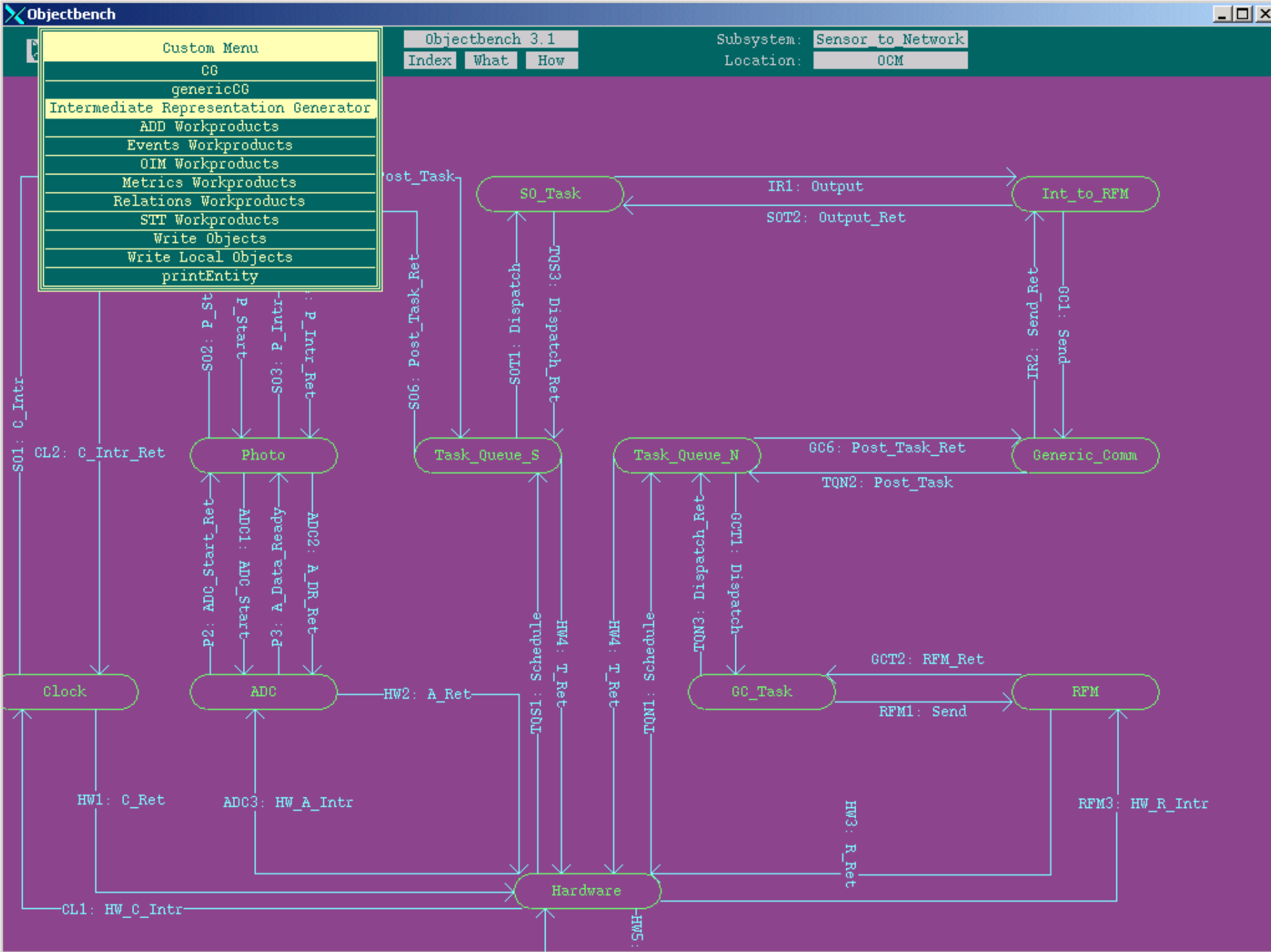
Objectbench 3.1 | Object | Subsystem: Sensor_to_Network
Index | What | How | Events | Location: Sensor_Output Lifecycle
Close

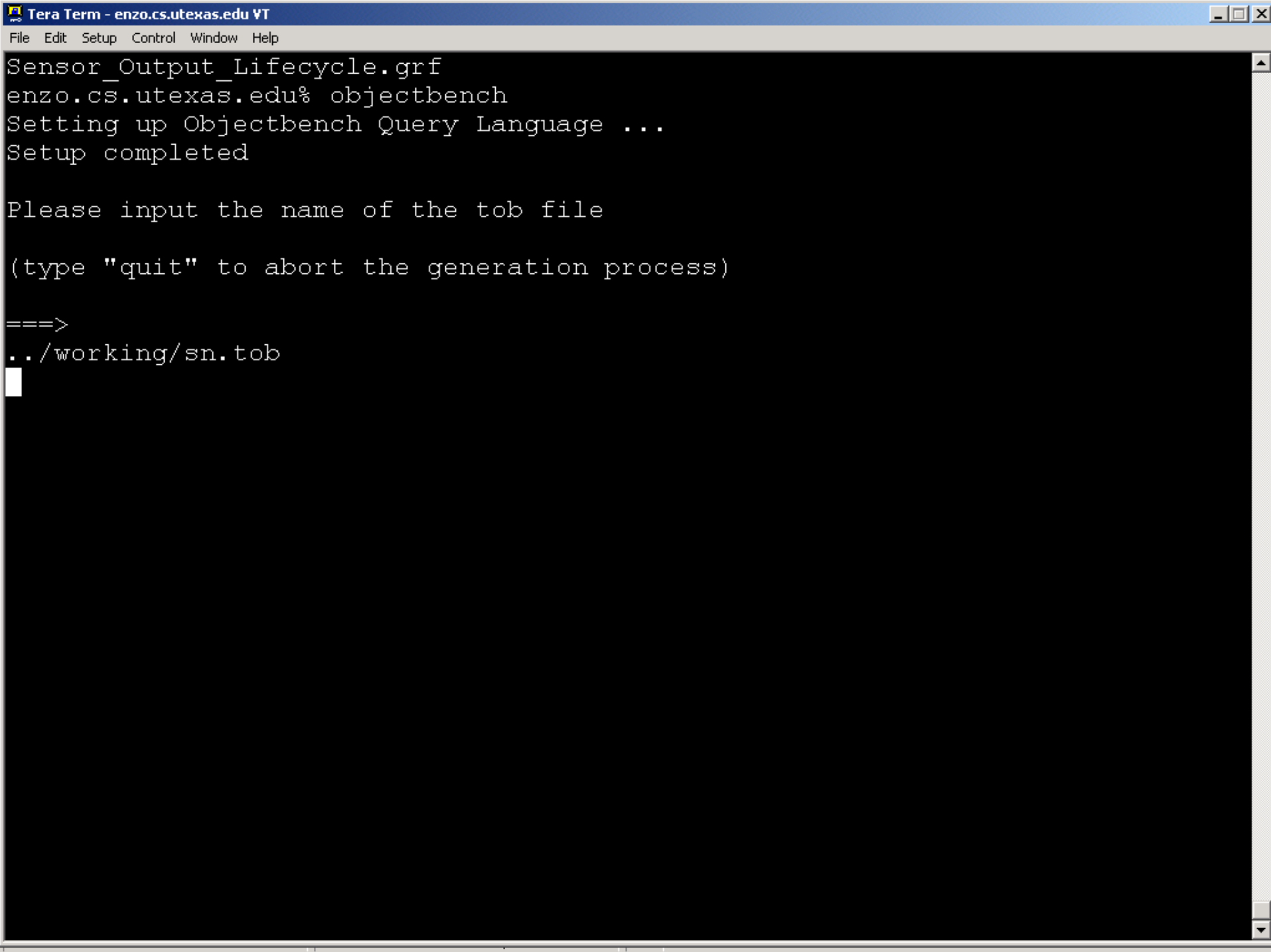


0 | 166 | T 6 | Scope Settings | No Log File

Stop | Continue | Cont View | Cont Thread | Cont Break | Step | Step View | Step Thread | Step Inst

```
{3,NULL,0,NULL,"Clock(1)"} @ 0 # 140 traversed arc from '1. Idle' to '2. Ticking' in submodel 'Clock Lifecycle'
<31> step {port} ;
{3,NULL,0,NULL,"Clock(1)"} @ 0 # 142 generating 'S01: C_Intr' to Sensor_Output(1)
{3,NULL,0,NULL,"Clock(1)\nSensor_Output(1)"} @ 0 # 146 traversed arc di_3cc2f81e16_3_78 from 'Clock Lifecycle' to 'Sen
<35> step {port} ;
{3,NULL,0,NULL,"Clock(1)"} @ 0 # 151 queued 'S01: C_Intr' to Sensor_Output(1)
{6,NULL,0,NULL,"Sensor_Output(1)"} @ 0 # 163 processing 'S01: C_Intr'
Command>
```

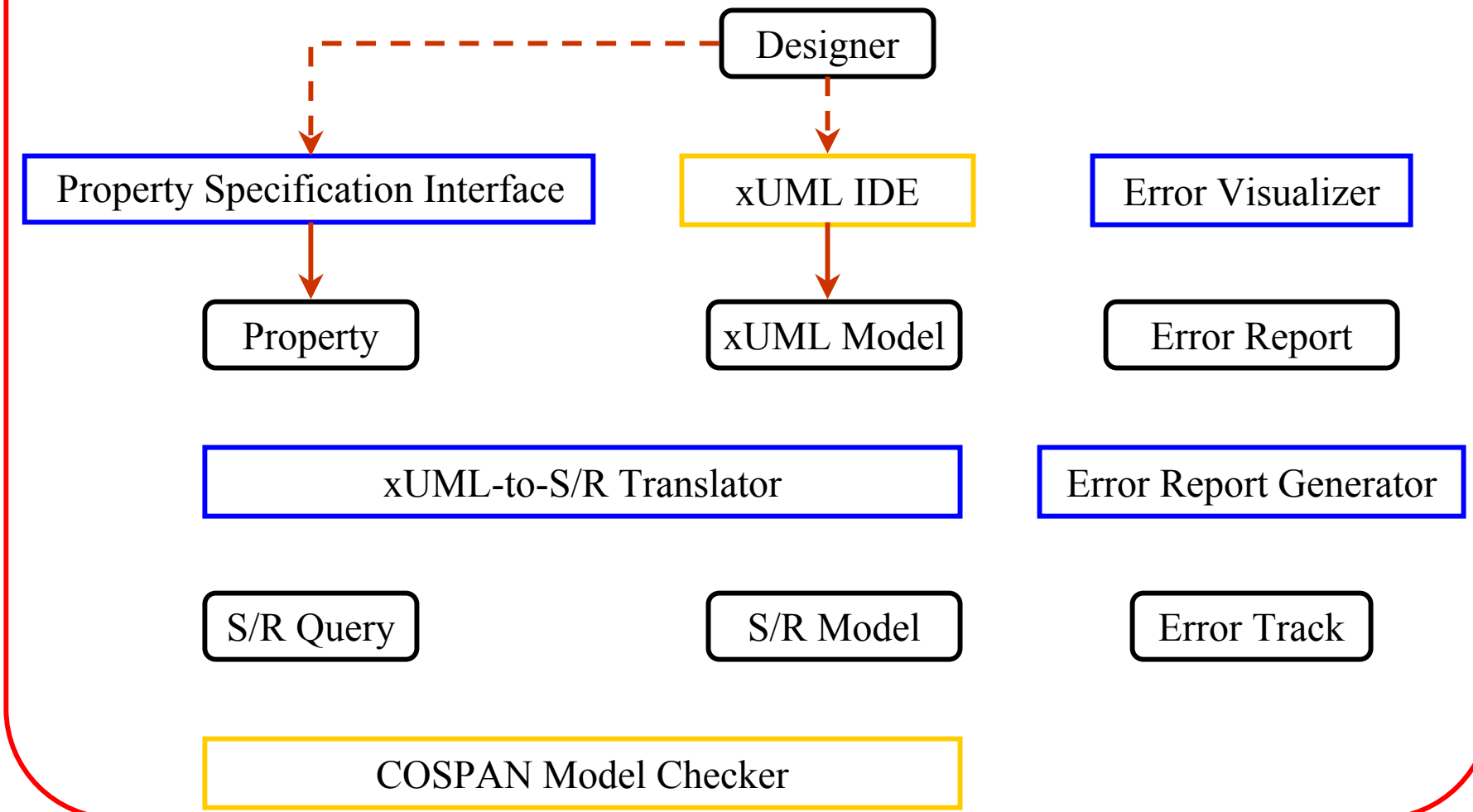

Objectbench
Custom Menu
CG
genericCG
Intermediate Representation Generator
ADD Workproducts
Events Workproducts
OIM Workproducts
Metrics Workproducts
Relations Workproducts
STT Workproducts
Write Objects
Write Local Objects
printEntity
Objectbench 3.1
Index
What
How
Subsystem: Sensor_to_Network
Location: OCM
SO_Task
IR1: Output
SOT2: Output_Ret
Int_to_RFM
SO1: C_Intr
CL2: C_Intr_Ret
SO2: P_St
P_Start
SO3: P_Intr
P_Intr_Ret
SO6: Post_Task_Ret
SOT1: Dispatch
TQS3: Dispatch_Ret
IR2: Send_Ret
GC1: Send
Photo
Task_Queue_S
Task_Queue_N
GC6: Post_Task_Ret
TQN2: Post_Task
Generic_Comm
P2: ADC_Start_Ret
ADC1: ADC_Start
P3: A_Data_Ready
ADC2: A_DR_Ret
TQS1: Schedule
HW4: T_Ret
HW4: T_Ret
TQN1: Schedule
TQN3: Dispatch_Ret
GCT1: Dispatch
GCT2: RFM_Ret
Clock
ADC
HW2: A_Ret
GC_Task
RFM
RFM1: Send
HW1: C_Ret
ADC3: HW_A_Intr
HW3: R_Ret
RFM3: HW_R_Intr
Hardware
CL1: HW_C_Intr
HW5:


Tera Term - enzo.cs.utexas.edu VT
File Edit Setup Control Window Help
Sensor_Output_Lifecycle.grf
enzo.cs.utexas.edu% objectbench
Setting up Objectbench Query Language ...
Setup completed
Please input the name of the tob file
(type "quit" to abort the generation process)
===>
../working/sn.tob

```
project s2n
consistsof
    domain s2n

    syntype Interrupt_Type = integer
        constants [0 4] default 0
    endsyntype

    arraytype Integer_Array = Integer
        6
    endarraytype

    0 0
        subsystem Sensor_to_Network
        0 0
            internal
                object Hardware
                HW
                    Id 'HW_ID' *
                    Interrupt_Type 'Choice' .

                    item
                        HW_Running Yes 1
                        ('HW_ID', 1)
                        ('Choice', 0)
                    enditem

                    1

                    0

                    event HW1: C_Ret ( id key )
                    event HW2: A_Ret ( id key )
                    event HW3: R_Ret ( id key )
                    event HW4: T_Ret ( id key )
                    event HW5: Loop ( id key )

                    state HW_Running HW_SDL_Running
                        0
                        0
                        Choice = ANY("Interrupt_Type")

                        if (Choice == 0) Generate CL1: HW_C_Intr (1)

                        if (Choice == 1) {
                         if (ADC(1).On == TRUE) Generate ADC3: HW_A_Intr (1)
```

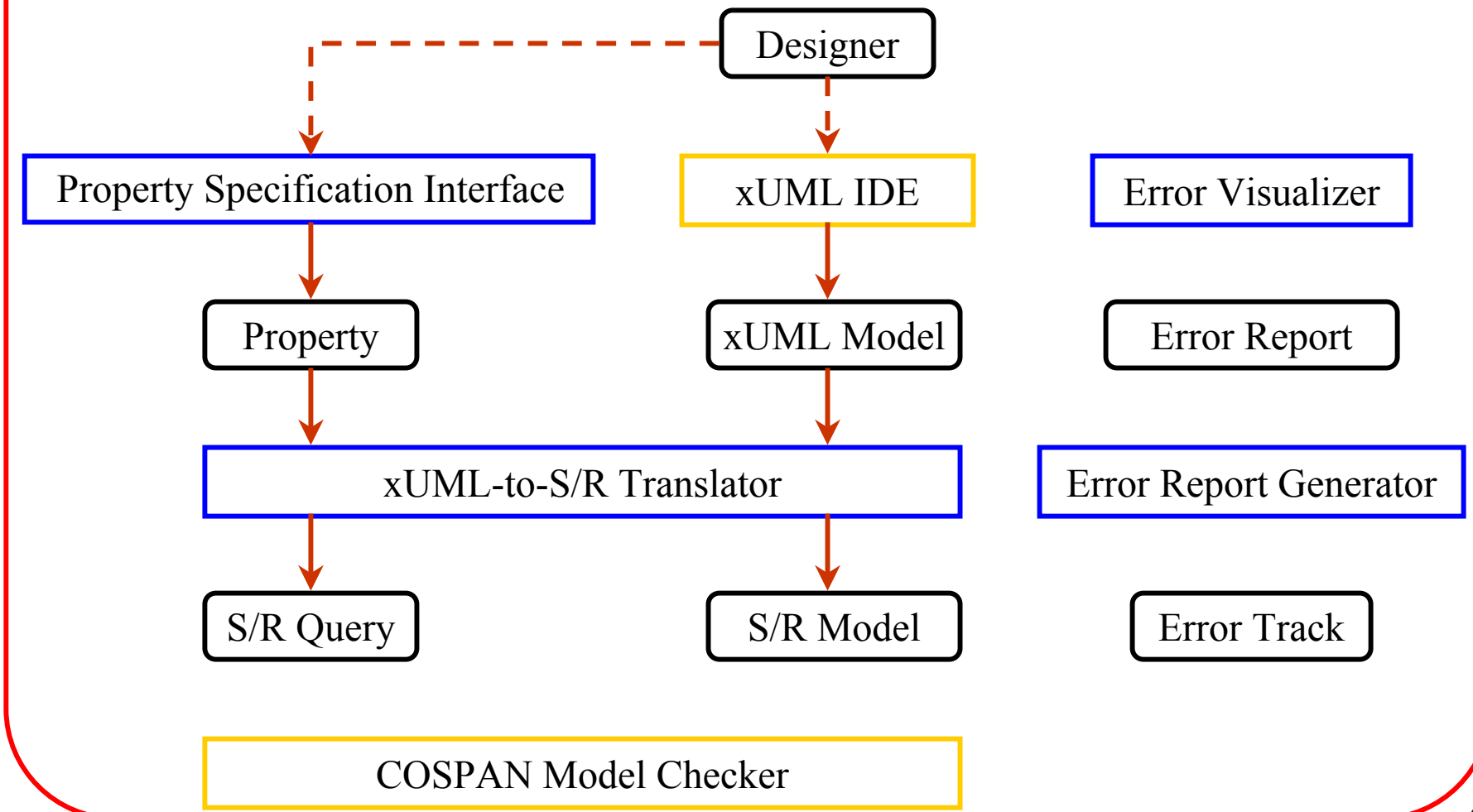

Loading timer...done

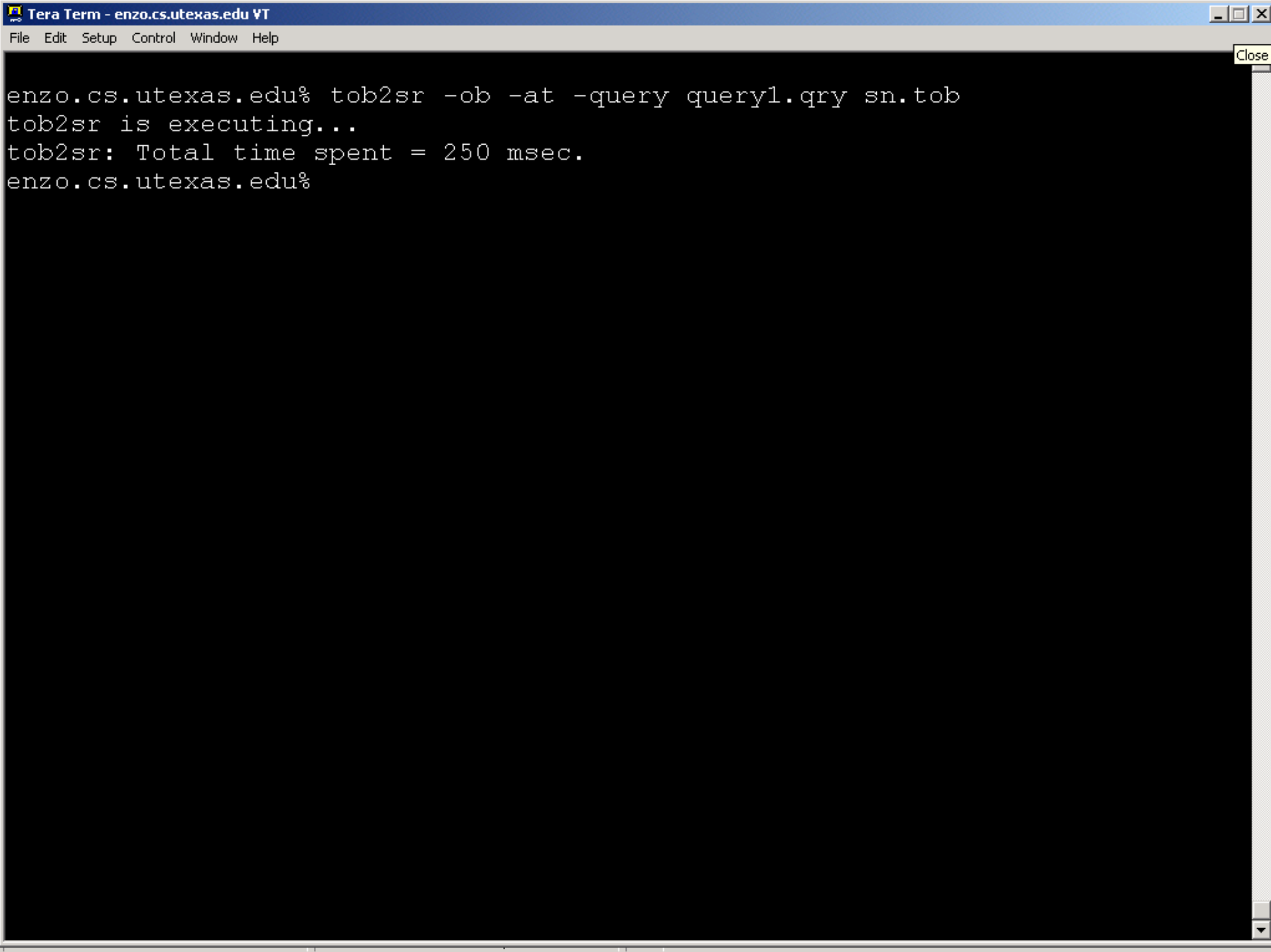
Step-by-Step Demonstration



```
/* Logic Propositions */
DECLARE RFM_Pending <<SYSTEM S2N/BLOCK SENSOR_TO_NETWORK/PROCESS RFM>> PENDING;
DECLARE HW_Choice <<SYSTEM S2N/BLOCK SENSOR_TO_NETWORK/PROCESS HARDWARE>> CHOICE;

/* Properties */
Repeatedly(#RFM_Pending == 1);
Repeatedly(#RFM_Pending != 1);

/* Assumptions */
AssumeRepeatedly(#HW_Choice == 0);
AssumeRepeatedly(#HW_Choice == 1);
AssumeRepeatedly(#HW_Choice == 2);
AssumeRepeatedly(#HW_Choice == 3);
AssumeRepeatedly(#HW_Choice == 4);
```

Step-by-Step Demonstration



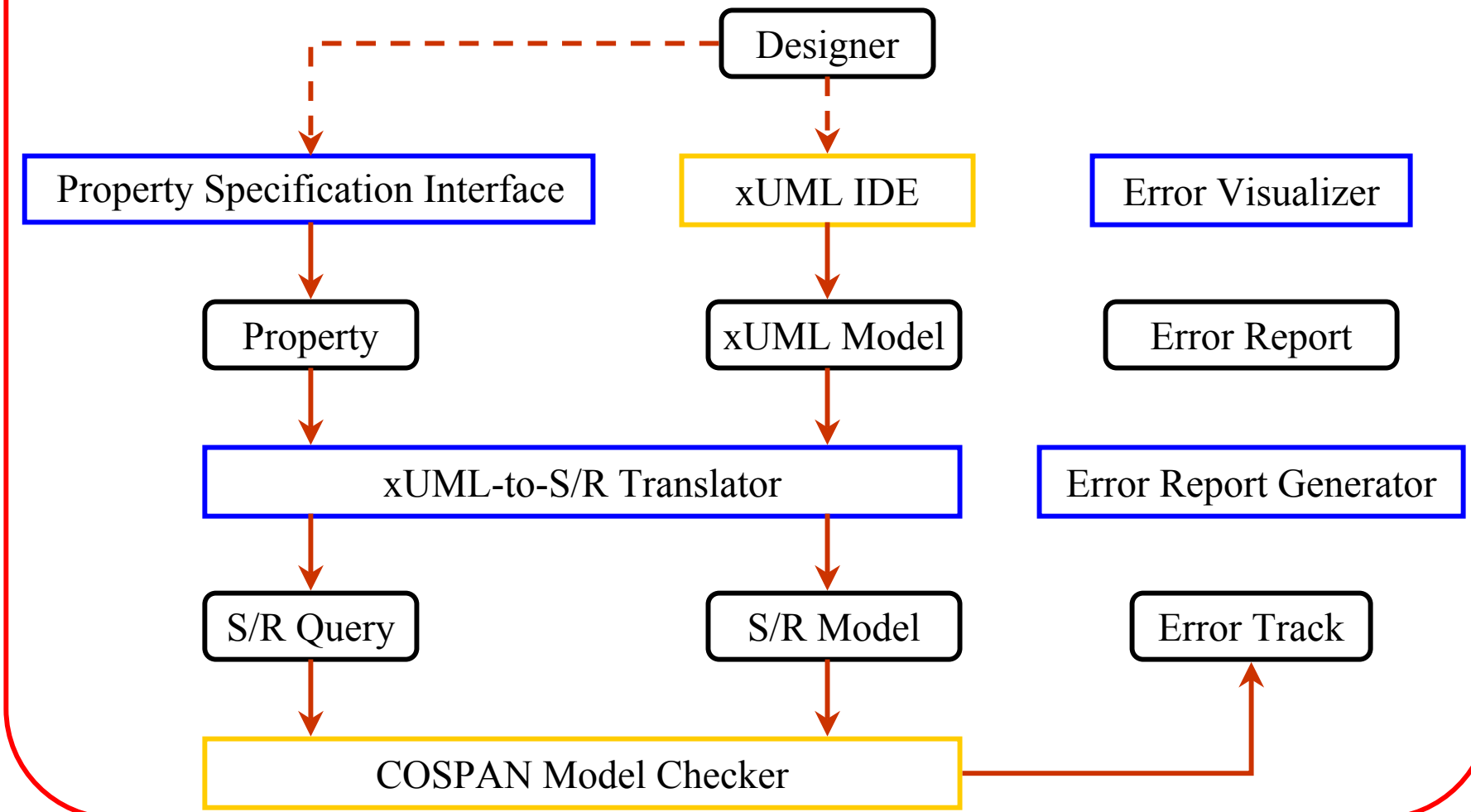

Tera Term - enzo.cs.utexas.edu VT
File Edit Setup Control Window Help
Close
enzo.cs.utexas.edu% tob2sr -ob -at -query query1.qry sn.tob
tob2sr is executing...
tob2sr: Total time spent = 250 msec.
enzo.cs.utexas.edu%

```
/*****************************************************************************
* Generated by tob2sr (Version : 0.37.1) on Mon Mar 24 14:39:00 2003
* The command line was:
*     tob2sr -ob -at -query query1.qry sn.tob
*****************************************************************************/

#if ! defined(__QUERY_LOCAL)

/*************************************/
/*        SDL SYSTEM DESCRIPTION     */
/*************************************/

/*************************************/
/*          TYPE  DEFINITIONS        */
/*************************************/

# define __DummySignal 0

/* incoming signals of processes and channels */

# define Process_HARDWARE_HW5 1
# define Process_HARDWARE_HW4 2
# define Process_HARDWARE_HW3 3
# define Process_HARDWARE_HW2 4
# define Process_HARDWARE_HW1 5
type Process_HARDWARE_Signals : ( __DummySignal, Process_HARDWARE_HW5, Process_HARDWARE_HW4, Process_HARDWARE_\
HW3, Process_HARDWARE_HW2, Process_HARDWARE_HW1 )

# define Process_CLOCK_CL2 1
# define Process_CLOCK_CL1 2
type Process_CLOCK_Signals : ( __DummySignal, Process_CLOCK_CL2, Process_CLOCK_CL1 )

# define Process_ADC_ADC3 1
# define Process_ADC_ADC2 2
# define Process_ADC_ADC1 3
type Process_ADC_Signals : ( __DummySignal, Process_ADC_ADC3, Process_ADC_ADC2, Process_ADC_ADC1 )

# define Process_PHOTO_P4 1
# define Process_PHOTO_P3 2
# define Process_PHOTO_P2 3
# define Process_PHOTO_P1 4
type Process_PHOTO_Signals : ( __DummySignal, Process_PHOTO_P4, Process_PHOTO_P3, Process_PHOTO_P2, Process_PH\
OTO_P1 )

# define Process_SENSOR_OUTPUT_SO6 1
# define Process_SENSOR_OUTPUT_SO3 2
```


Loading timer...done

```
          Process_TASK_QUEUE_N_TQN1 ? (__Sender = p_Process_HARDWARE) |
          __DummySignal
       )
}

selvar __InputSig : Process_TASK_QUEUE_N_Signals
asgn __InputSig := __SigArray[__SlotToBeConsumed]
end BufferProcess_TASK_QUEUE_N



#endif

#if ! defined(__MODEL_LOCAL)

/*****************************************/
/*              QUERY PART               */
/*****************************************/

#define DIRECTSUMof 2

#include <QRY.h>

/*****************************************/
/*              USER QUERY               */
/*****************************************/

/* Query variable definitions */

#define __QV__RFM_Pending (Process_RFM.V_PENDING)
#define __QV__HW_Choice (Process_HARDWARE.V_CHOICE)

/* Properties */
monitor __Property__1 : Repeatedly_ (0, (__QV__RFM_Pending=1))
monitor __Property__2 : Repeatedly_ (1, (__QV__RFM_Pending~=1))

/* Assumptions from the query file */
monitor __Assumption__1 : AssumeRepeatedly_ ((__QV__HW_Choice=0))
monitor __Assumption__2 : AssumeRepeatedly_ ((__QV__HW_Choice=1))
monitor __Assumption__3 : AssumeRepeatedly_ ((__QV__HW_Choice=2))
monitor __Assumption__4 : AssumeRepeatedly_ ((__QV__HW_Choice=3))
monitor __Assumption__5 : AssumeRepeatedly_ ((__QV__HW_Choice=4))
#endif
```

Step-by-Step Demonstration

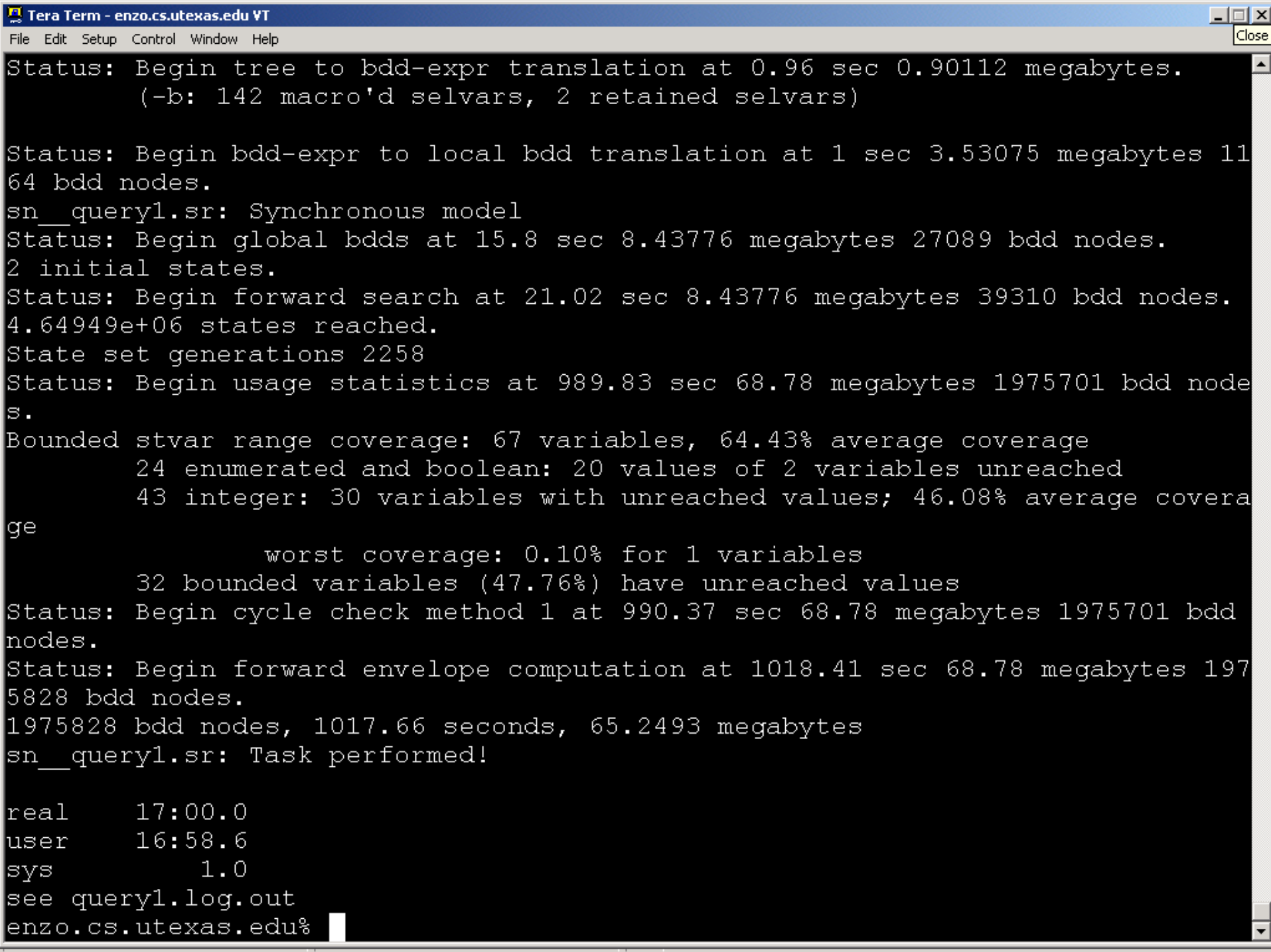


Tera Term - enzo.cs.utexas.edu VT

File Edit Setup Control Window Help

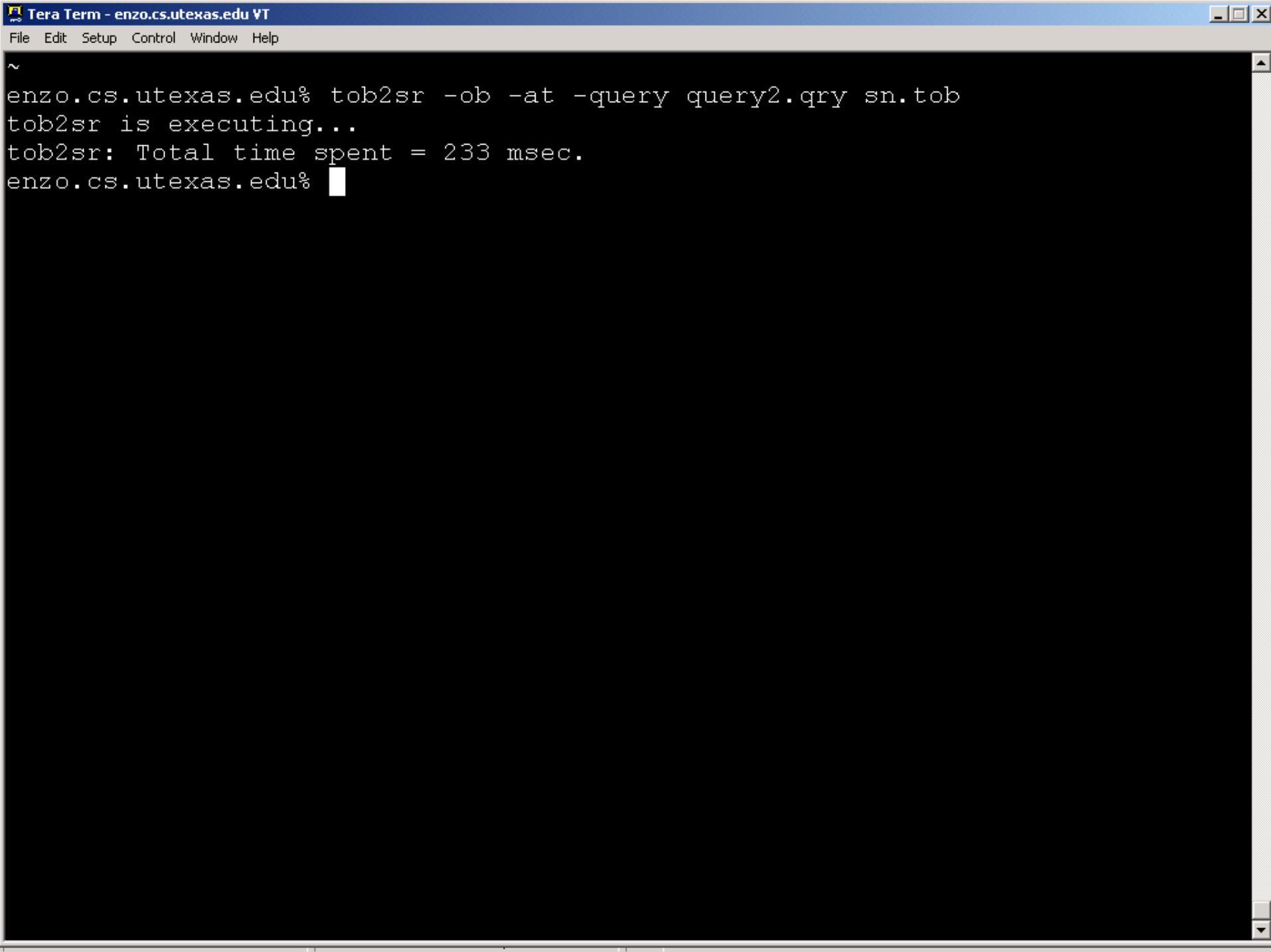
Close

```
================================================================
Mon Apr  7 17:16:31 CDT 2003
enzo.cs.utexas.edu [SunOS 5.8 sun4u]: /v/filer1a/v0q031/feixie/project/OBChec
k/demo/tos/working1
cospan -bq1 -Kt -#status -#dupstvars -#time -D__MODEL_LOCAL sn__query1.sr -D_
_QUERY_LOCAL sn__query1.sr

cospan: Version 8.23.206 (Bell Laboratories)  1 Feb 2002
        Single pass for option -q1
cospan: Version 8.23.206 (Bell Laboratories)  1 Feb 2002
+ time sr_F -I/projects/formalcheck/COSPAN/SUN/include -#status -#dupstvars -
I/projects/formalcheck/COSPAN/SUN/include -b -#reduction -Kt -#status -#dupst
vars -D__MODEL_LOCAL sn__query1.sr -D__QUERY_LOCAL sn__query1.sr -#qtree -#st
atus -#dupstvars -#reduction -#status -#dupstvars -#qtree
sr_F: -#bddversion=d1
Status: Begin parsing at 0 sec 0 megabytes.
sn__query1.sr: Mon Apr  7 17:16:17 2003
/projects/formalcheck/COSPAN/SUN/include/QRY.h: Mon Mar 18 10:42:35 2002
/projects/formalcheck/COSPAN/SUN/include/QRY+.h: Mon Mar 18 10:42:35 2002
Status: Begin checks and tree rewrites at 0.1 sec 0.73728 megabytes.
sn__query1.rf: list entry count:
        93 pruned, 211 active, 0 freed by reduction
        39 data variables declared or with width >= -#databits=4
        144 selection/local variables
        0 resized variables
        67 bounded state variables: 2.58e53 states
        0 unbounded state variables
        5 boolean cysets
        2 boolean recurs
        2 free selection/local variables: 60 selections/state
        0 pausing processes
```


Tera Term - enzo.cs.utexas.edu VT
File Edit Setup Control Window Help
Close
Status: Begin tree to bdd-expr translation at 0.96 sec 0.90112 megabytes.
(-b: 142 macro'd selvars, 2 retained selvars)
Status: Begin bdd-expr to local bdd translation at 1 sec 3.53075 megabytes 11
64 bdd nodes.
sn__query1.sr: Synchronous model
Status: Begin global bdds at 15.8 sec 8.43776 megabytes 27089 bdd nodes.
2 initial states.
Status: Begin forward search at 21.02 sec 8.43776 megabytes 39310 bdd nodes.
4.64949e+06 states reached.
State set generations 2258
Status: Begin usage statistics at 989.83 sec 68.78 megabytes 1975701 bdd node
s.
Bounded stvar range coverage: 67 variables, 64.43% average coverage
24 enumerated and boolean: 20 values of 2 variables unreached
43 integer: 30 variables with unreached values; 46.08% average covera
ge
worst coverage: 0.10% for 1 variables
32 bounded variables (47.76%) have unreached values
Status: Begin cycle check method 1 at 990.37 sec 68.78 megabytes 1975701 bdd
nodes.
Status: Begin forward envelope computation at 1018.41 sec 68.78 megabytes 197
5828 bdd nodes.
1975828 bdd nodes, 1017.66 seconds, 65.2493 megabytes
sn__query1.sr: Task performed!
real 17:00.0
user 16:58.6
sys 1.0
see query1.log.out
enzo.cs.utexas.edu%

```
/* Logic Propositions */
DECLARE HW_Choice <<SYSTEM S2N/BLOCK SENSOR_TO_NETWORK/PROCESS HARDWARE>> CHOICE;

DECLARE RFM_Buf <<SYSTEM S2N/BLOCK SENSOR_TO_NETWORK/PROCESS RFM>> BUF;
DECLARE RFM_Data <<SYSTEM S2N/BLOCK SENSOR_TO_NETWORK/PROCESS RFM>> DATA;
DECLARE RFM_Trasmitting <<SYSTEM S2N/BLOCK SENSOR_TO_NETWORK/PROCESS RFM>> $RFM_TRANSMITTING;

/* Properties */
Never ((#RFM_Data == 0) AND (#RFM_Buf == 0) AND #RFM_Trasmitting);

/* Assumptions */
AssumeRepeatedly(#HW_Choice == 0);
AssumeRepeatedly(#HW_Choice == 1);
AssumeRepeatedly(#HW_Choice == 2);
AssumeRepeatedly(#HW_Choice == 3);
AssumeRepeatedly(#HW_Choice == 4);
```


Loading timer...done


Tera Term - enzo.cs.utexas.edu VT
File Edit Setup Control Window Help
~
enzo.cs.utexas.edu% tob2sr -ob -at -query query2.qry sn.tob
tob2sr is executing...
tob2sr: Total time spent = 233 msec.
enzo.cs.utexas.edu%

Tera Term - enzo.cs.utexas.edu VT

File Edit Setup Control Window Help

Close

```
================================================================
Mon Apr  7 17:16:55 CDT 2003
enzo.cs.utexas.edu [SunOS 5.8 sun4u]: /v/filer1a/v0q031/feixie/project/OBChec
k/demo/tos/working2
cospan -bq1 -Kt -#status -#dupstvars -#time -D__MODEL_LOCAL sn__query2.sr -D_
_QUERY_LOCAL sn__query2.sr

cospan: Version 8.23.206 (Bell Laboratories)  1 Feb 2002
        Single pass for option -q1
cospan: Version 8.23.206 (Bell Laboratories)  1 Feb 2002
+ time sr_F -I/projects/formalcheck/COSPAN/SUN/include -#status -#dupstvars -
I/projects/formalcheck/COSPAN/SUN/include -b -#reduction -Kt -#status -#dupst
vars -D__MODEL_LOCAL sn__query2.sr -D__QUERY_LOCAL sn__query2.sr -#qtree -#st
atus -#dupstvars -#reduction -#status -#dupstvars -#qtree
sr_F: -#bddversion=d1
Status: Begin parsing at 0 sec 0 megabytes.
sn__query2.sr: Mon Apr  7 17:16:53 2003
/projects/formalcheck/COSPAN/SUN/include/QRY.h: Mon Mar 18 10:42:35 2002
/projects/formalcheck/COSPAN/SUN/include/QRY+.h: Mon Mar 18 10:42:35 2002
Status: Begin checks and tree rewrites at 0.11 sec 0.73728 megabytes.
sn__query2.rf: list entry count:
        60 pruned, 243 active, 0 freed by reduction
        39 data variables declared or with width >= -#databits=4
        153 selection/local variables
        1 resized variables
        90 bounded state variables: 1.82e126 states
        0 unbounded state variables
        6 boolean cysets
        0 boolean recurs
        2 free selection/local variables: 60 selections/state
        1 variable reference clippings, 0 expression clippings
```

Tera Term - enzo.cs.utexas.edu VT

File Edit Setup Control Window Help

```
Status: Begin tree to bdd-expr translation at 1.04 sec 0.94208 megabytes.
        (-b: 151 macro'd selvars, 2 retained selvars)

Status: Begin bdd-expr to local bdd translation at 1.16 sec 3.56352 megabytes
 1652 bdd nodes.
sn__query2.sr: Synchronous model
Status: Begin global bdds at 16.07 sec 8.81459 megabytes 30959 bdd nodes.
1 initial states.
Status: Begin forward search at 23.74 sec 8.81459 megabytes 68301 bdd nodes.
Stop at error:
  No transition enabled in process .__Property__1
3916 states reached.
State set generations 248
Status: Begin error track at 59.72 sec 11.092 megabytes 164852 bdd nodes.
Status: Begin usage statistics at 64.58 sec 11.1002 megabytes 164852 bdd node
s.
Bounded stvar range coverage: 89 variables, 47.11% average coverage
        24 enumerated and boolean: 30 values of 4 variables unreached
        65 integer: 52 variables with unreached values; 30.50% average covera
ge
                worst coverage: 0.05% for 10 variables
        56 bounded variables (62.92%) have unreached values
156202 bdd nodes, 64.91 seconds, 7.81517 megabytes
sn__query2.sr: Task failed (tree).
sn__query2.sr: (tree), exit 0

real     1:06.6
user     1:06.1
sys         0.1
see query2.log.out
enzo.cs.utexas.edu%
```

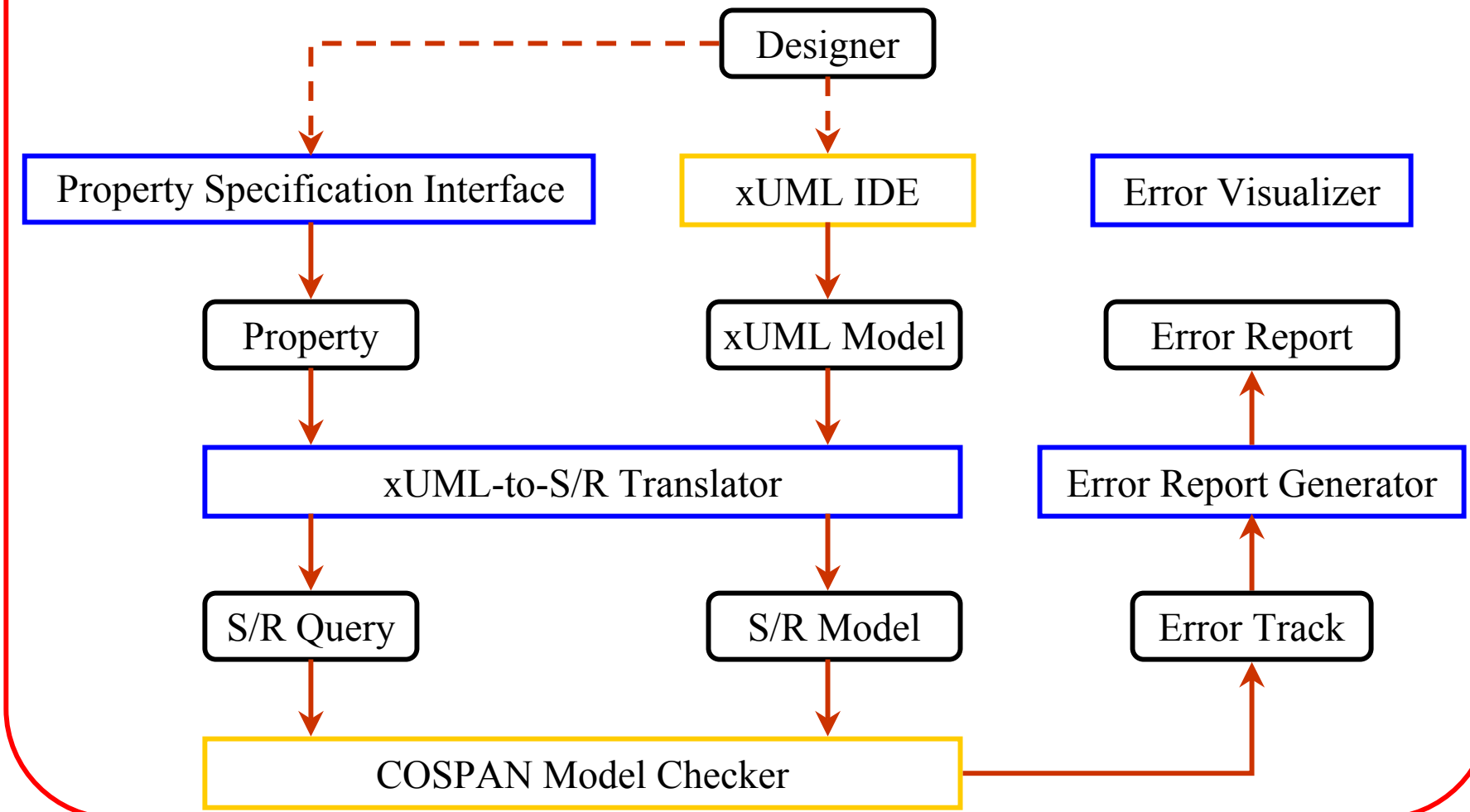
Tera Term - enzo.cs.utexas.edu VT

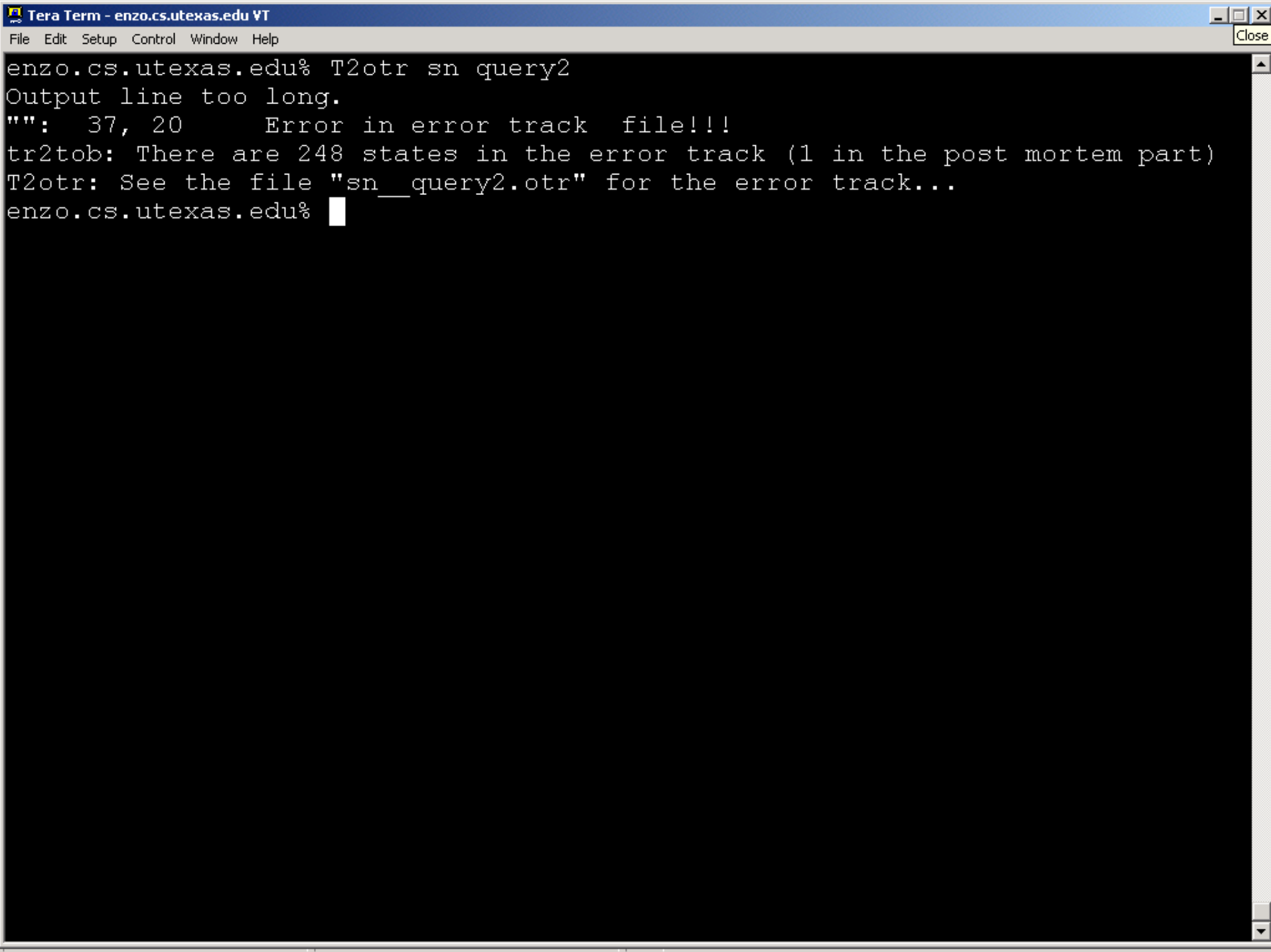
File Edit Setup Control Window Help

Close

```
enzo.cs.utexas.edu% more sn__query2.T
0(0)     .Process_HARDWARE.$=state_START_1        .Process_HARDWARE.V_CHOICE=0.
BufferProcess_HARDWARE.__SigArray[0]=0   .BufferProcess_HARDWARE.__SigArray[1]
=0       .BufferProcess_HARDWARE.__SigArray[2]=0 .BufferProcess_HARDWARE.__Sig
Array[3]=0       .Process_CLOCK.$=state_CL_SDL_IDLE_2     .BufferProcess_CLOCK.
__SigArray[0]=0 .BufferProcess_CLOCK.__SigArray[1]=0     .BufferProcess_CLOCK.
__SigArray[2]=0 .Process_ADC.$=state_ADC_SDL_IDLE_3      .Process_ADC.V_ON=0 .
Process_ADC.V_READING=0 .BufferProcess_ADC.__SigArray[0]=0       .Buff
erProcess_ADC.__SigArray[1]=0    .BufferProcess_ADC.__SigArray[2]=0       .Proc
ess_PHOTO.$=state_P_SDL_IDLE_3   .Process_PHOTO.V_DATA=0 .BufferProcess_PHOTO.
__SigArray[0]=0 .BufferProcess_PHOTO.__SigArray[1]=0     .BufferProcess_PHOTO.
__SigArray[2]=0 .BufferProcess_PHOTO.__ParArray[0][0]=0 .BufferProcess_PHOTO.
__ParArray[1][0]=0       .BufferProcess_PHOTO.__ParArray[2][0]=0 .Process_SENS
OR_OUTPUT.$=state_SO_SDL_IDLE_3 .Process_SENSOR_OUTPUT.V_BUF=0   .Process_SENS
OR_OUTPUT.V_DATA=0       .BufferProcess_SENSOR_OUTPUT.__SigArray[0]=0     .Buff
erProcess_SENSOR_OUTPUT.__SigArray[1]=0 .BufferProcess_SENSOR_OUTPUT.__SigArr
ay[2]=0 .BufferProcess_SENSOR_OUTPUT.__ParArray[0][0]=0 .BufferProcess_SENSOR
_OUTPUT.__ParArray[1][0]=0       .BufferProcess_SENSOR_OUTPUT.__ParArray[2][0]
=0       .Process_RFM.$=state_RFM_SDL_IDLE_3      .Process_RFM.V_PENDING=0     .
Process_RFM.V_BUF=-1     .Process_RFM.V_DATA=0    .BufferProcess_RFM.__SigArray
[0]=0    .BufferProcess_RFM.__SigArray[1]=0       .BufferProcess_RFM.__SigArray
[2]=0    .BufferProcess_RFM.__ParArray[0][0]=0    .BufferProcess_RFM.__ParArray
[1][0]=0         .BufferProcess_RFM.__ParArray[2][0]=0    .Process_GENERIC_COMM
.$=state_GC_SDL_IDLE_2   .Process_GENERIC_COMM.V_BUF=0    .Process_GENERIC_COMM
.V_DATA=0        .BufferProcess_GENERIC_COMM.__SigArray[0]=0      .BufferProces
s_GENERIC_COMM.__SigArray[1]=0   .BufferProcess_GENERIC_COMM.__SigArray[2]=0 .
BufferProcess_GENERIC_COMM.__ParArray[0][0]=0    .BufferProcess_GENERIC_COMM._
_ParArray[1][0]=0        .BufferProcess_GENERIC_COMM.__ParArray[2][0]=0   .Proc
ess_INT_TO_RFM.$=state_IR_SDL_IDLE_2     .Process_INT_TO_RFM.V_DATA=0     .Buff
erProcess_INT_TO_RFM.__SigArray[0]=0     .BufferProcess_INT_TO_RFM.__SigArray[
--More--(2%)
```

Step-by-Step Demonstration



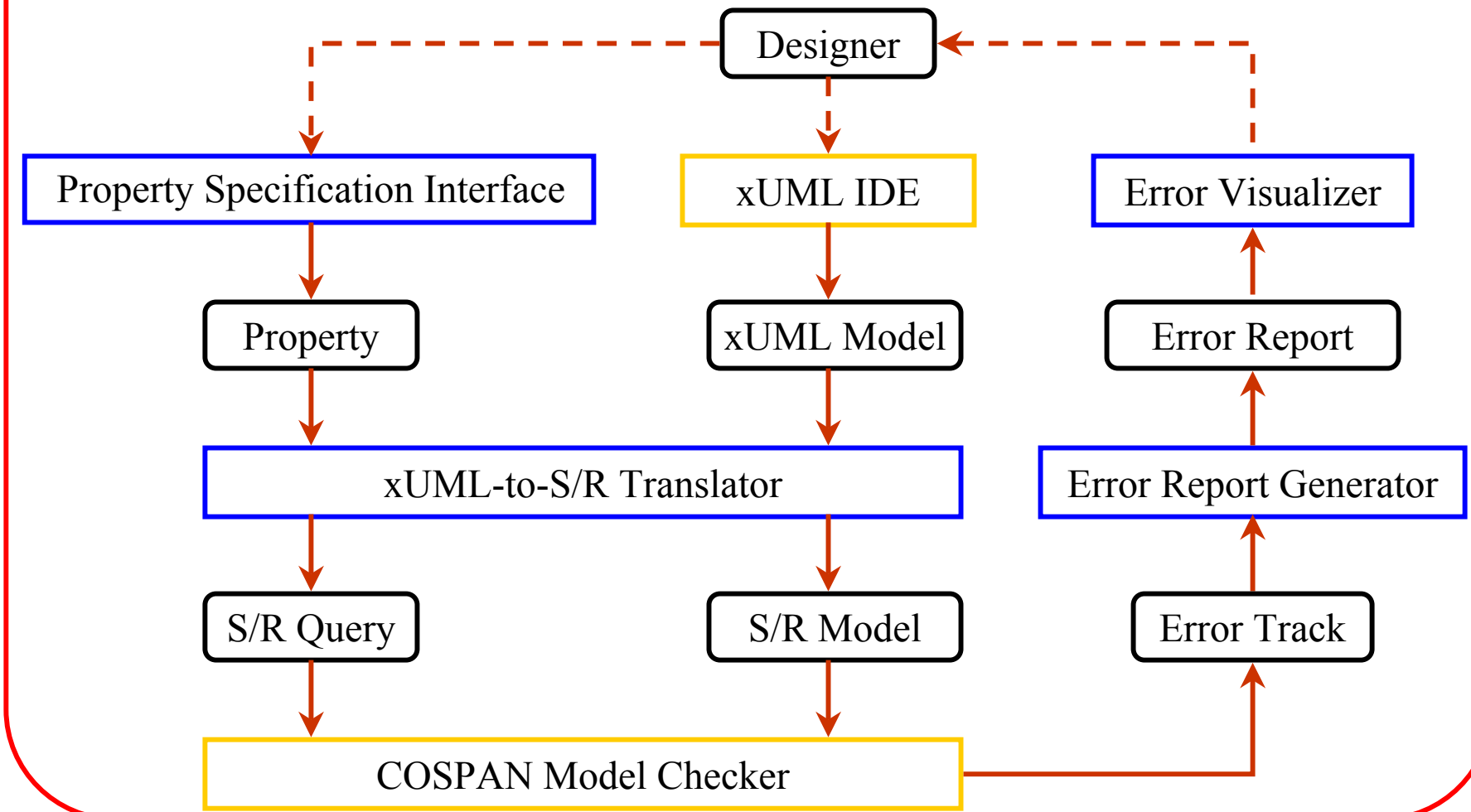

Tera Term - enzo.cs.utexas.edu VT
File Edit Setup Control Window Help
Close
enzo.cs.utexas.edu% T2otr sn query2
Output line too long.
"": 37, 20 Error in error track file!!!
tr2tob: There are 248 states in the error track (1 in the post mortem part)
T2otr: See the file "sn__query2.otr" for the error track...
enzo.cs.utexas.edu%

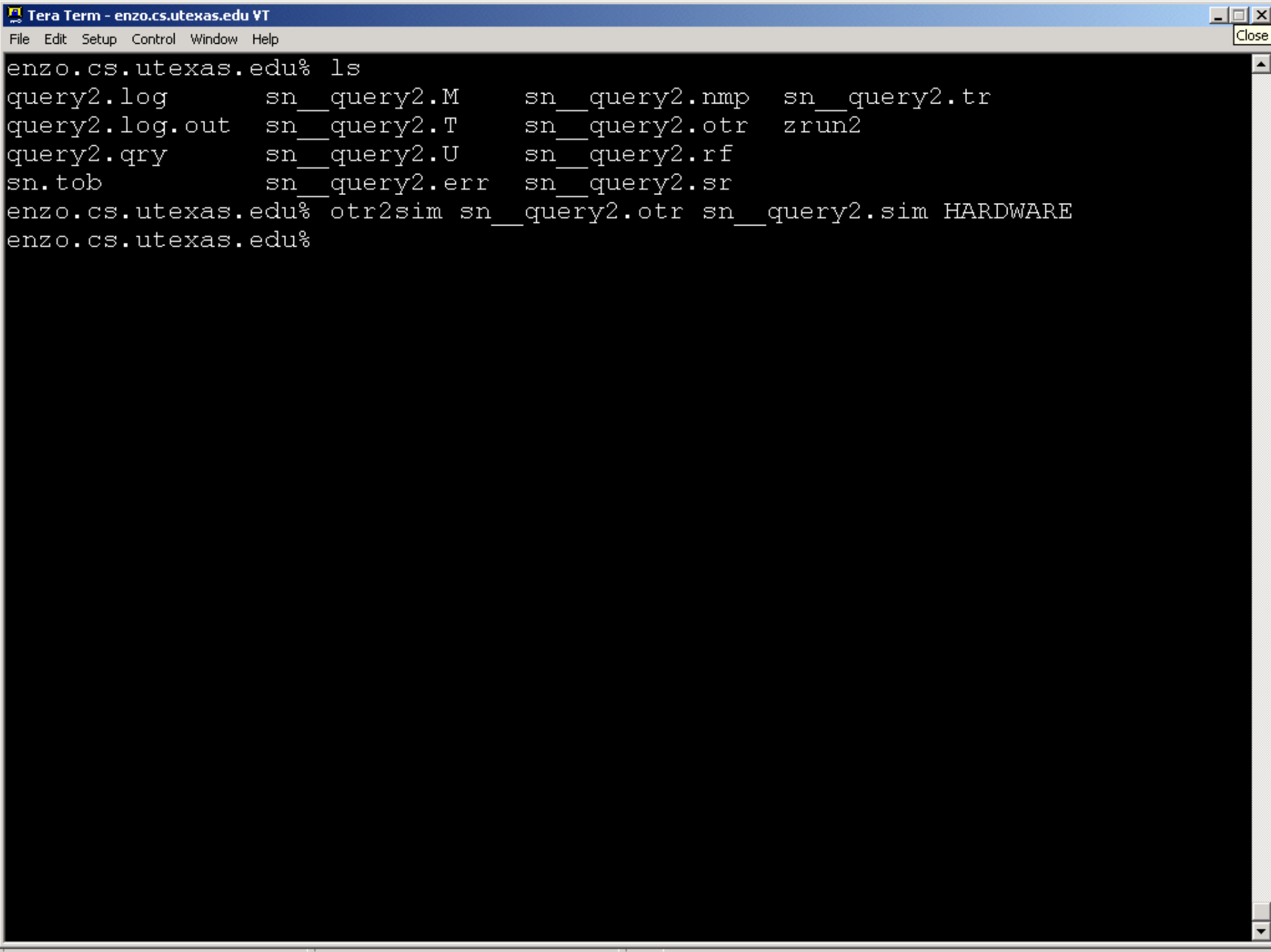
Tera Term - enzo.cs.utexas.edu VT

File Edit Setup Control Window Help

```
enzo.cs.utexas.edu% more sn__query2.otr
==========================
------- State 1 -------
==========================
PROCESS HARDWARE is initially at STATE START
   <<SYSTEM S2N / BLOCK SENSOR_TO_NETWORK / PROCESS HARDWARE>>CHOICE=0
PROCESS CLOCK is initially at STATE CL_SDL_IDLE
PROCESS ADC is initially at STATE ADC_SDL_IDLE
   <<SYSTEM S2N / BLOCK SENSOR_TO_NETWORK / PROCESS ADC>>ON=0
   <<SYSTEM S2N / BLOCK SENSOR_TO_NETWORK / PROCESS ADC>>READING=0
PROCESS PHOTO is initially at STATE P_SDL_IDLE
   <<SYSTEM S2N / BLOCK SENSOR_TO_NETWORK / PROCESS PHOTO>>DATA=0
PROCESS SENSOR_OUTPUT is initially at STATE SO_SDL_IDLE
   <<SYSTEM S2N / BLOCK SENSOR_TO_NETWORK / PROCESS SENSOR_OUTPUT>>BUF=0
   <<SYSTEM S2N / BLOCK SENSOR_TO_NETWORK / PROCESS SENSOR_OUTPUT>>DATA=0
PROCESS RFM is initially at STATE RFM_SDL_IDLE
   <<SYSTEM S2N / BLOCK SENSOR_TO_NETWORK / PROCESS RFM>>PENDING=0
   <<SYSTEM S2N / BLOCK SENSOR_TO_NETWORK / PROCESS RFM>>BUF=1
   <<SYSTEM S2N / BLOCK SENSOR_TO_NETWORK / PROCESS RFM>>DATA=0
PROCESS GENERIC_COMM is initially at STATE GC_SDL_IDLE
   <<SYSTEM S2N / BLOCK SENSOR_TO_NETWORK / PROCESS GENERIC_COMM>>BUF=0
   <<SYSTEM S2N / BLOCK SENSOR_TO_NETWORK / PROCESS GENERIC_COMM>>DATA=0
PROCESS INT_TO_RFM is initially at STATE IR_SDL_IDLE
   <<SYSTEM S2N / BLOCK SENSOR_TO_NETWORK / PROCESS INT_TO_RFM>>DATA=0
PROCESS TASK_QUEUE_S is initially at STATE TQS_SDL_IDLE
   <<SYSTEM S2N / BLOCK SENSOR_TO_NETWORK / PROCESS TASK_QUEUE_S>>FULL=0
   <<SYSTEM S2N / BLOCK SENSOR_TO_NETWORK / PROCESS TASK_QUEUE_S>>EMP=1
   <<SYSTEM S2N / BLOCK SENSOR_TO_NETWORK / PROCESS TASK_QUEUE_S>>HEAD=0
   <<SYSTEM S2N / BLOCK SENSOR_TO_NETWORK / PROCESS TASK_QUEUE_S>>TAIL=0
PROCESS SO_TASK is initially at STATE SOT_SDL_IDLE
--More--(4%)
```

Step-by-Step Demonstration



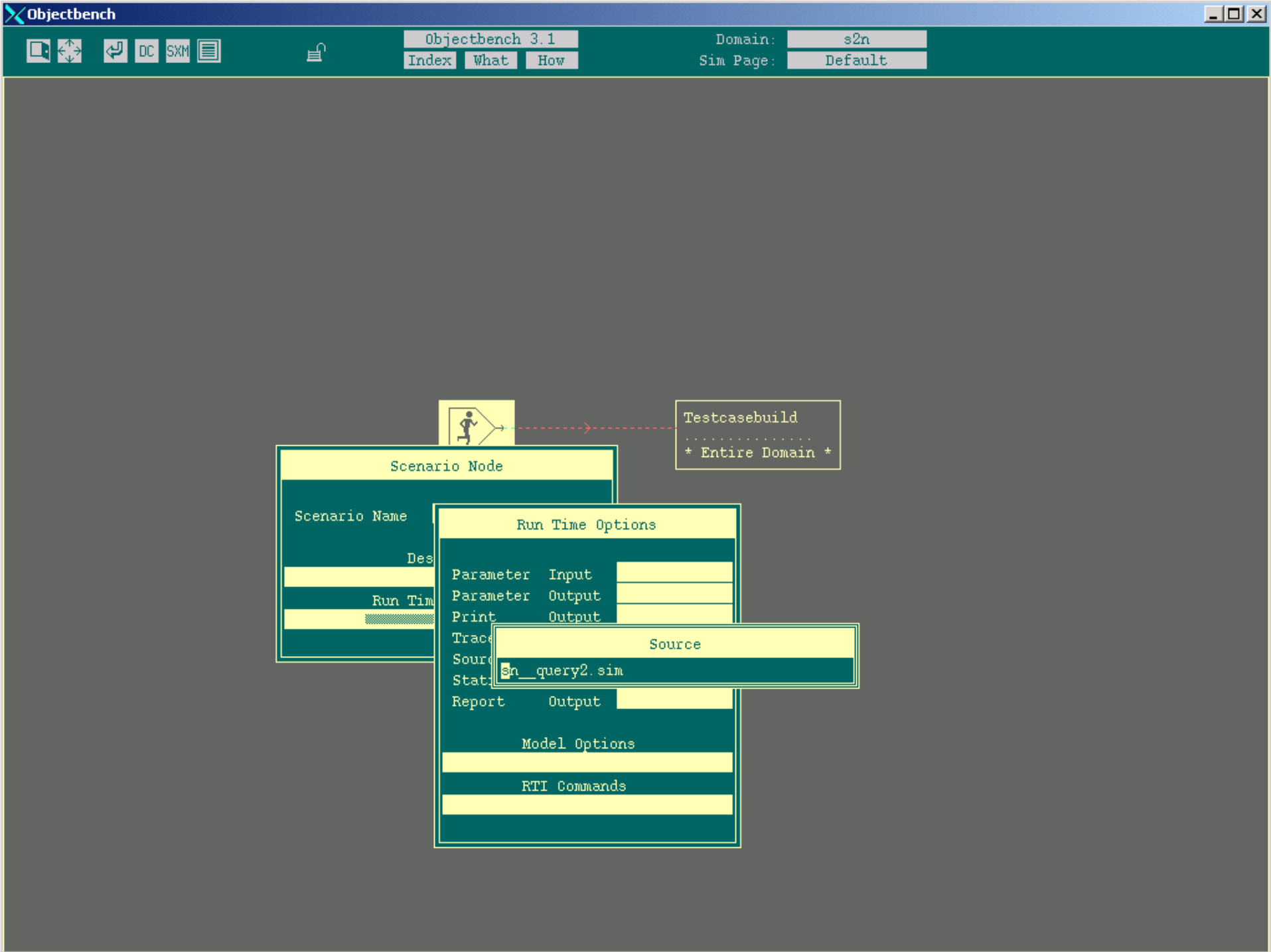

Tera Term - enzo.cs.utexas.edu VT
File Edit Setup Control Window Help
Close
enzo.cs.utexas.edu% ls
query2.log sn__query2.M sn__query2.nmp sn__query2.tr
query2.log.out sn__query2.T sn__query2.otr zrun2
query2.qry sn__query2.U sn__query2.rf
sn.tob sn__query2.err sn__query2.sr
enzo.cs.utexas.edu% otr2sim sn__query2.otr sn__query2.sim HARDWARE
enzo.cs.utexas.edu%

Tera Term - enzo.cs.utexas.edu VT

File Edit Setup Control Window Help

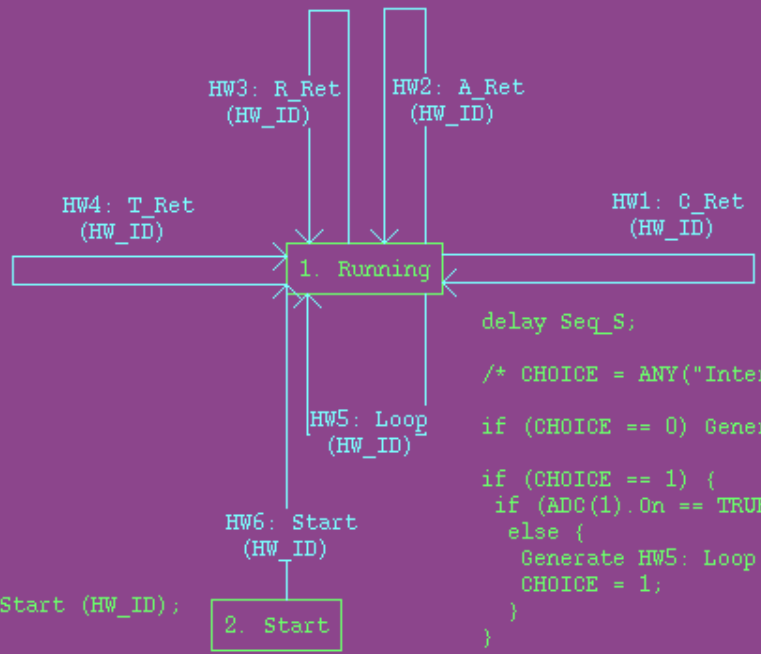
Close

```
/****************** Interleaving orders: ********************/

delay 0;
HARDWARE(1).Seq_S = 1;
HARDWARE(1).CHOICE = 0;

delay 1;
HARDWARE(1).Seq_T = 15;

delay 1;
CLOCK(1).Seq_S = 0;

delay 1;
CLOCK(1).Seq_T = 0;

delay 1;
SENSOR_OUTPUT(1).Seq_S = 0;

delay 1;
SENSOR_OUTPUT(1).Seq_T = 0;

delay 1;
PHOTO(1).Seq_S = 0;

delay 1;
PHOTO(1).Seq_T = 0;

delay 1;
ADC(1).Seq_S = 0;
--More--(16%)
```


Objectbench
DC
SXM
Objectbench 3.1
Index
What
How
Domain:
s2n
Sim Page:
Default
Testcasebuild
* Entire Domain *
Scenario Node
Scenario Name
Run Time Options
Parameter Input
Parameter Output
Print Output
Report Output
Source
sn__query2.sim
Model Options
RTI Commands

Objectbench

Objectbench 3.1 | Index | What | How | Object | Events | Subsystem: Sensor_to_Network | Location: HARDWARE LIFECYCLE

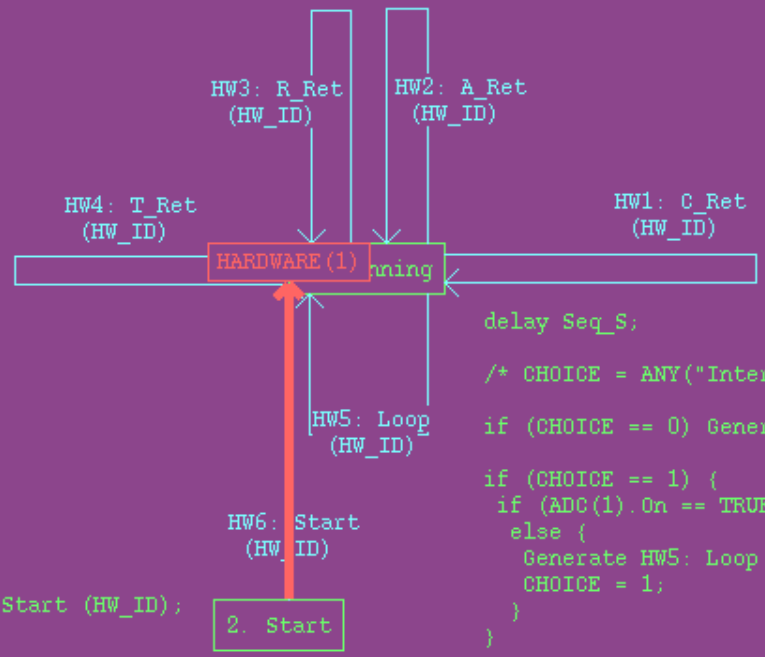



```
Generate HW6: Start (HW_ID);

delay Seq_T;
```

```
delay Seq_S;

/* CHOICE = ANY("Interrupt_Type"); */

if (CHOICE == 0) Generate CL1: HW_C_Intr (1);

if (CHOICE == 1) {
 if (ADC(1).On == TRUE) Generate ADC3: HW_A_Intr (1);
  else {
   Generate HW5: Loop (HW_ID);
   CHOICE = 1;
  }
}

if (CHOICE == 2) {
 if (RFM(1).Pending == TRUE) Generate RFM3: HW_R_Intr (1);
  else {
   Generate HW5: Loop (HW_ID);
   CHOICE = 2;
  }
}

if (CHOICE == 3) {
 if (TASK_QUEUE_S(1).Emp == FALSE) Generate TQS1: Schedule (1);
  else {
   Generate HW5: Loop (HW_ID);
   CHOICE = 3;
  }
}

if (CHOICE == 4) {
 if (TASK_QUEUE_N(1).Emp == FALSE) Generate TQN1: Schedule (1);
  else {
   Generate HW5: Loop (HW_ID);
   CHOICE = 4;
  }
```

Objectbench

Objectbench 3.1 | Index | What | How | Object | Events | Subsystem: Sensor_to_Network | Location: HARDWARE LIFECYCLE



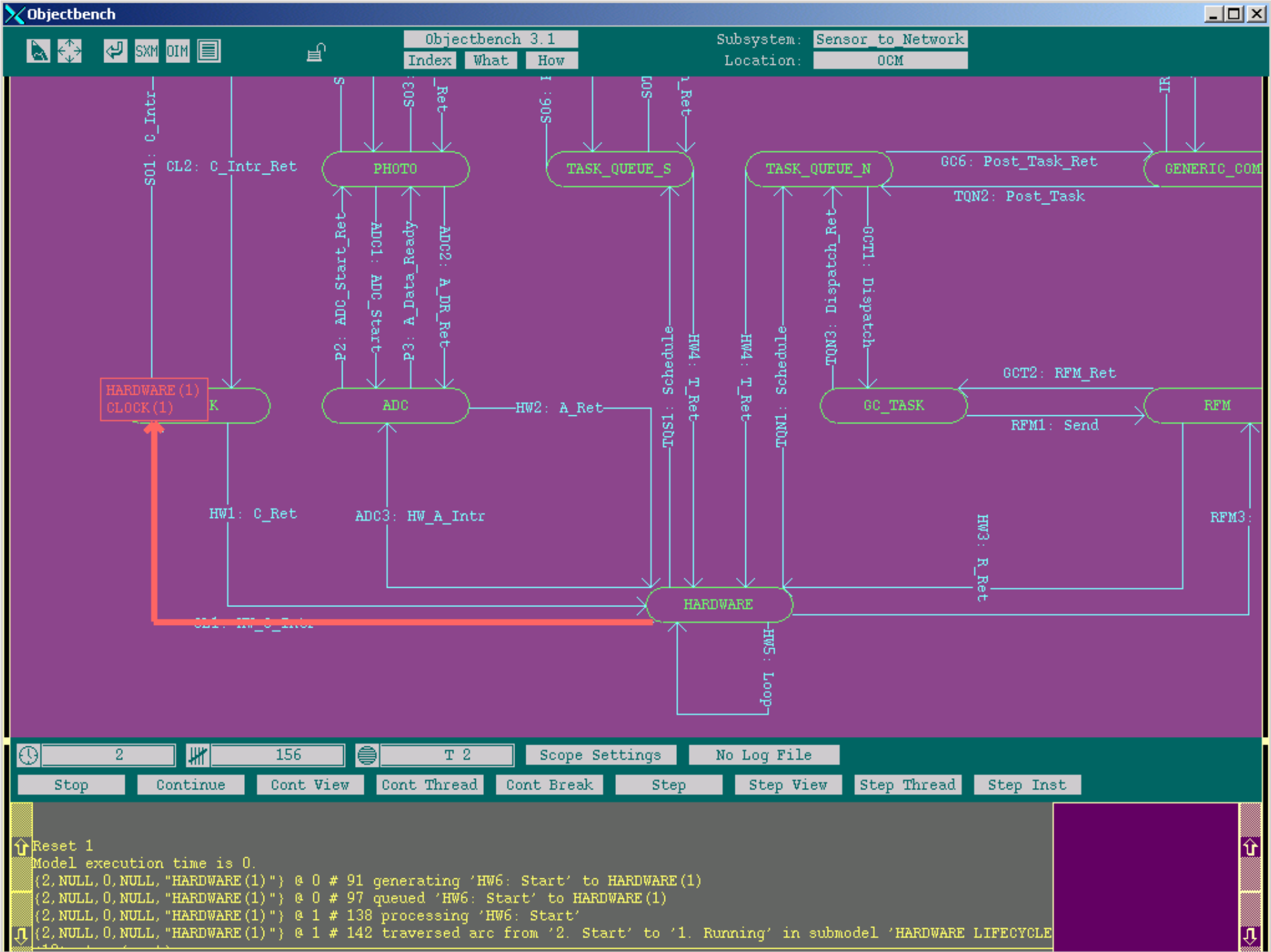

```
Generate HW6: Start (HW_ID);

delay Seq_T;
```

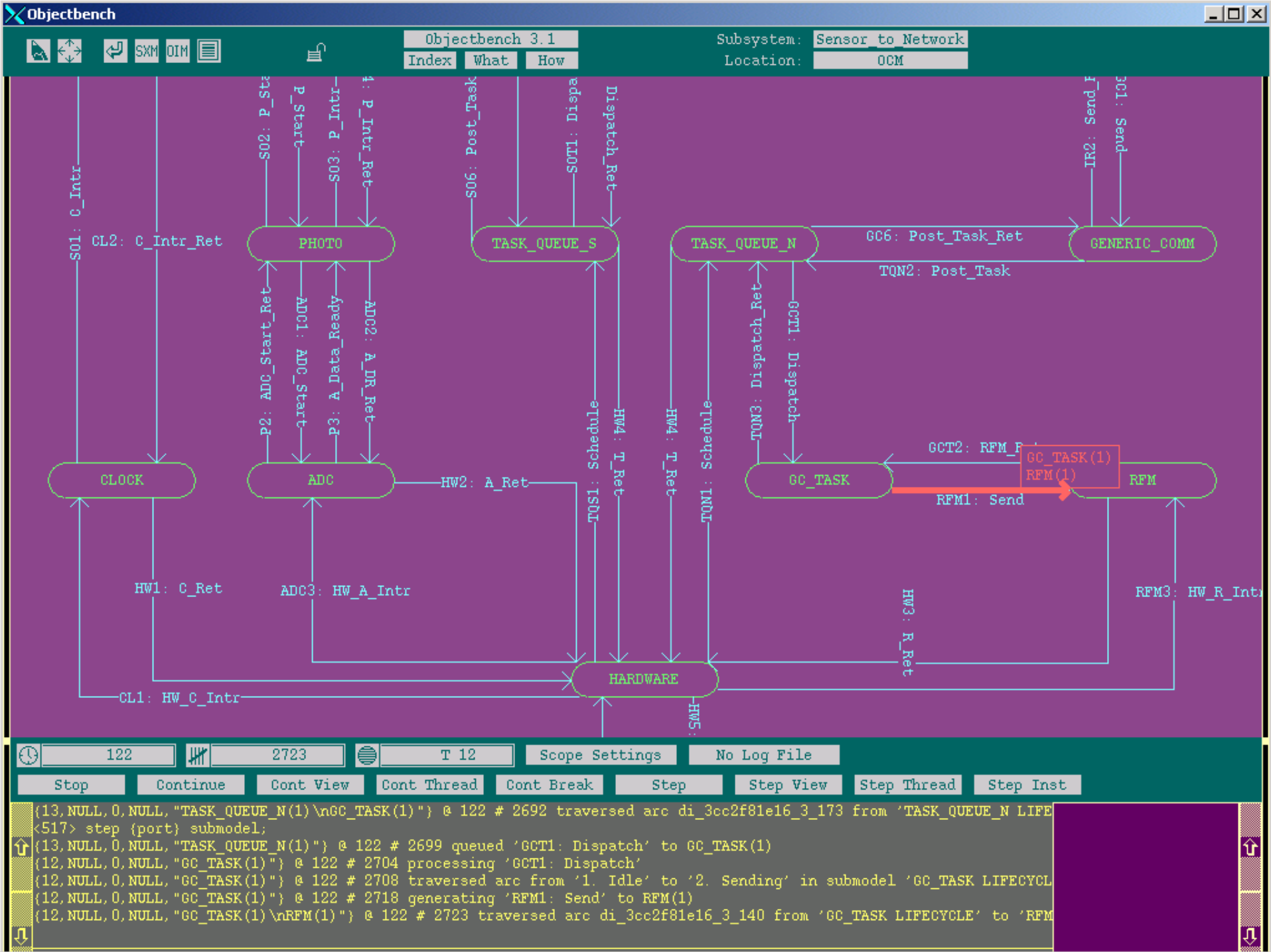
```
delay Seq_S;

/* CHOICE = ANY("Interrupt_Type"); */

if (CHOICE == 0) Generate CL1: HW_C_Intr (1);

if (CHOICE == 1) {
 if (ADC(1).On == TRUE) Generate ADC3: HW_A_Intr (1);
  else {
   Generate HW5: Loop (HW_ID);
   CHOICE = 1;
  }
}

if (CHOICE == 2) {
 if (RFM(1).Pending == TRUE) Generate RFM3: HW_R_Intr (1);
  else {
   Generate HW5: Loop (HW_ID);
   CHOICE = 2;
  }
```

1 | 142 | T 2 | Scope Settings | No Log File

Stop | Continue | Cont View | Cont Thread | Cont Break | Step | Step View | Step Thread | Step Inst

```
<4> step (port) ;


Reset 1
Model execution time is 0.
(2,NULL,0,NULL,"HARDWARE(1)") @ 0 # 91 generating 'HW6: Start' to HARDWARE(1)
(2,NULL,0,NULL,"HARDWARE(1)") @ 0 # 97 queued 'HW6: Start' to HARDWARE(1)
(2,NULL,0,NULL,"HARDWARE(1)") @ 1 # 138 processing 'HW6: Start'
(2,NULL,0,NULL,"HARDWARE(1)") @ 1 # 142 traversed arc from '2. Start' to '1. Running' in submodel 'HARDWARE LIFECYCLE
```


Objectbench
Objectbench 3.1
Index
What
How
Subsystem: Sensor_to_Network
Location: OCM
SXM
OIM
S01: C_Intr
CL2: C_Intr_Ret
PHOTO
P2: ADC_Start_Ret
ADC1: ADC_Start
P3: A_Data_Ready
ADC2: A_DR_Ret
TASK_QUEUE_S
TQS1: Schedule
HW4: T_Ret
TASK_QUEUE_N
TQN3: Dispatch_Ret
GCT1: Dispatch
TQN1: Schedule
HW4: T_Ret
GC6: Post_Task_Ret
TQN2: Post_Task
GENERIC_COM
HARDWARE(1)
CLOCK(1)
ADC
HW2: A_Ret
GC_TASK
GCT2: RFM_Ret
RFM1: Send
RFM
HW1: C_Ret
ADC3: HW_A_Intr
HW3: R_Ret
RFM3:
HARDWARE
HW5: Loop
2
156
T 2
Scope Settings
No Log File
Stop
Continue
Cont View
Cont Thread
Cont Break
Step
Step View
Step Thread
Step Inst
Reset 1
Model execution time is 0.
{2,NULL,0,NULL,"HARDWARE(1)"} @ 0 # 91 generating 'HW6: Start' to HARDWARE(1)
{2,NULL,0,NULL,"HARDWARE(1)"} @ 0 # 97 queued 'HW6: Start' to HARDWARE(1)
{2,NULL,0,NULL,"HARDWARE(1)"} @ 1 # 138 processing 'HW6: Start'
{2,NULL,0,NULL,"HARDWARE(1)"} @ 1 # 142 traversed arc from '2. Start' to '1. Running' in submodel 'HARDWARE LIFECYCLE


Objectbench
Objectbench 3.1
Index
What
How
Subsystem: Sensor_to_Network
Location: OCM
SXM
OIM
S01: C_Intr
CL2: C_Intr_Ret
S02: P_Start
P_Start
S03: P_Intr
P_Intr_Ret
S06: Post_Task
SGT1: Dispa
Dispatch_Ret
IR2: Send_I
IC1: Send
PHOTO
TASK_QUEUE_S
TASK_QUEUE_N
GC6: Post_Task_Ret
TQN2: Post_Task
GENERIC_COMM
P2: ADC_Start_Ret
ADC1: ADC_Start
P3: A_Data_Ready
ADC2: A_DR_Ret
TQS1: Schedule
HW4: T_Ret
HW4: T_Ret
TQN1: Schedule
TQN3: Dispatch_Ret
GCT1: Dispatch
GCT2: RFM_P
GC_TASK(1)
RFM(1)
RFM1: Send
CLOCK
ADC
HW2: A_Ret
GC_TASK
RFM
HW1: C_Ret
ADC3: HW_A_Intr
HW3: R_Ret
RFM3: HW_R_Intr
HARDWARE
CL1: HW_C_Intr
HW5:
122
2723
T 12
Scope Settings
No Log File
Stop
Continue
Cont View
Cont Thread
Cont Break
Step
Step View
Step Thread
Step Inst
(13,NULL,0,NULL,"TASK_QUEUE_N(1)\nGC_TASK(1)") @ 122 # 2692 traversed arc di_3cc2f81e16_3_173 from 'TASK_QUEUE_N LIFE
<517> step {port} submodel;
(13,NULL,0,NULL,"TASK_QUEUE_N(1)") @ 122 # 2699 queued 'GCT1: Dispatch' to GC_TASK(1)
(12,NULL,0,NULL,"GC_TASK(1)") @ 122 # 2704 processing 'GCT1: Dispatch'
(12,NULL,0,NULL,"GC_TASK(1)") @ 122 # 2708 traversed arc from '1. Idle' to '2. Sending' in submodel 'GC_TASK LIFECYCL
(12,NULL,0,NULL,"GC_TASK(1)") @ 122 # 2718 generating 'RFM1: Send' to RFM(1)
(12,NULL,0,NULL,"GC_TASK(1)\nRFM(1)") @ 122 # 2723 traversed arc di_3cc2f81e16_3_140 from 'GC_TASK LIFECYCLE' to 'RFM

Statistics from Model Checking "Repeated Output" Property

- Four model checking runs with different combinations of reduction algorithms
  - Start at the same time on the same host;
  - The host is a SUN with 8 CPUs and 2GB memory.

SPOR	SMC	Memory Usage	Time Usage
Off	Off	596M	19370S
Off	On	80M	1384S
On	Off	596M	17438S
On	On	102M	1379S

Conclusion: SMC helps and SPOR doesn't.

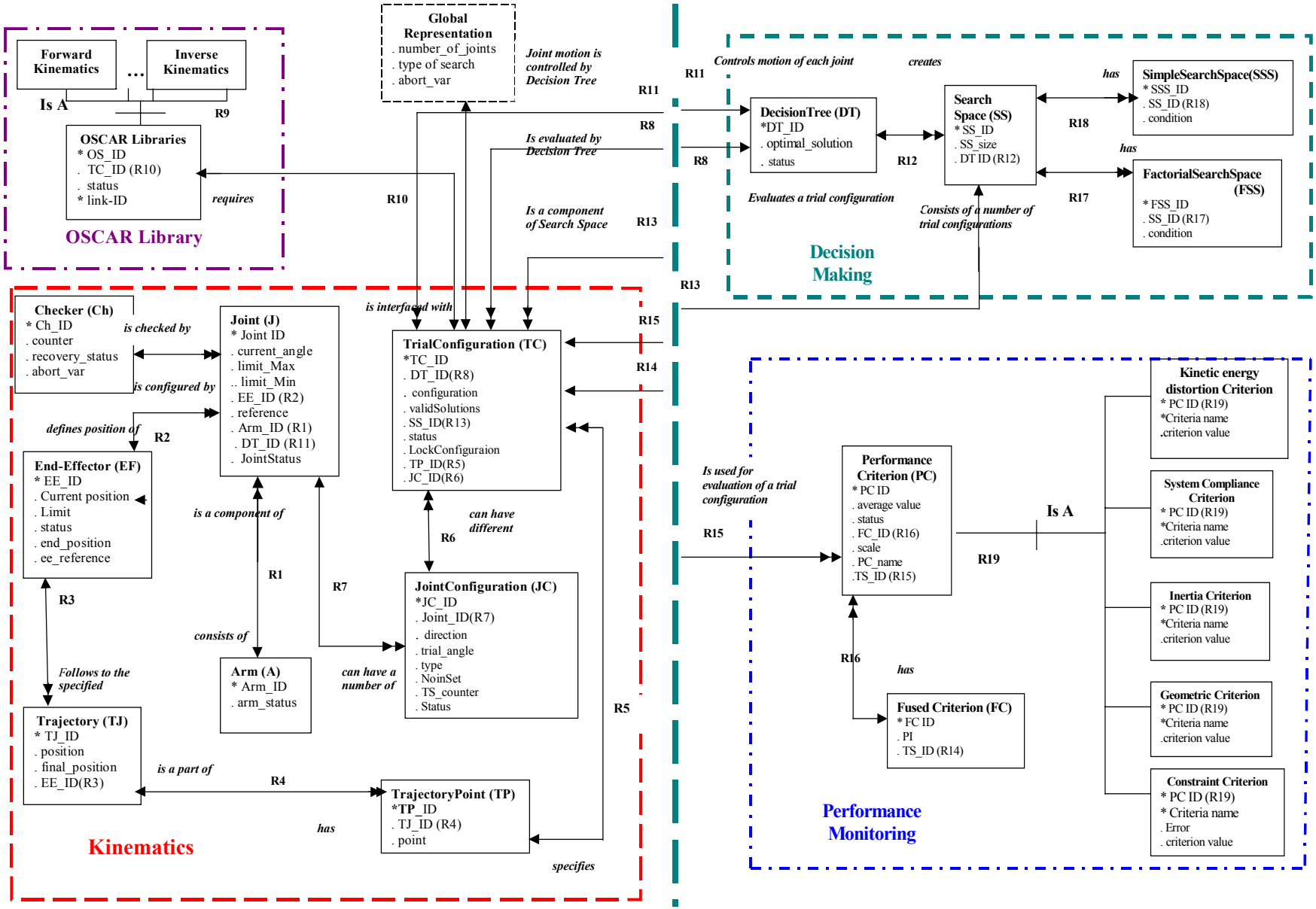
Presentation Outline

- Overview and Architecture of ObjectCheck
- Modeling and Verification of a TinyOS Run-time Image with ObjectCheck
- More Case Studies
- Summary and Future Work
- Work Built on ObjectCheck

Model Checking of NASA robot Controller

- A typical control-intensive embedded system;
- Conducted by Natasha Shyrigina using ObjectCheck;
  - 37 properties were checked.
  - 22 properties were successfully checked.
  - 6 bugs were found.

NASA Robot Controller



Properties to be Model checked

Table 4.4: Verification properties

N	Property	Robotic Description	Formal Description
1	EventuallyAlways(p=1)	Eventually the robot control terminates	Eventually permanently p=1
2	AlwaysUntil(p=0, t=1 OR (s=1 AND v=1))	The program terminates when it either completes the task or reaches the state where there is no solution for the fault recovery	p=0 holds at any execution of the program until occurrence of either t=1 or the combination of s=1 and v=1
3	AfterAlwaysUntil(q=i, r=l, p=l)	If the EE reaches an undesired position than the program terminates prior to a new move of the EE	At any point in the execution if q=l than it is followed by r=l until p is set to 1
4	Always(r=1 → u=1)	Whenever the EE is in the "Following_Desired_Trajectory" state than the Arm is in the "Valid" state	At any time during execution of the program when r=1 than u=1
5	AlwaysUntil(s=0, (($a_{1i} > max_{1i}$ OR $a_{1i} < min_{1i}$) OR ... OR ($a_{6i} > max_{6i}$ OR $a_{6i} < min_{6i}$)))	Fault recovery is executed when any of the joint angles does not satisfy the allowed limits	s=0 holds at any execution until any of the following does not holds: ($a_{1i} > max_{1i}$ OR $a_{1i} < min_{1i}$),...,($a_{6i} > max_{6i}$ OR $a_{6i} < min_{6i}$)
6	Always(s=1 AND j_{1i}=1 AND ... AND j_{6i}=1)	Fault recovery is always executed for joints that reside in their most recent base position	At any execution of the program it is always the case that s=1 and all of the following hold j_{1i}=1,...,j_{6i}=1
7	NeverUntil(r=1 AND s=1, r=0)	When fault recovery is called, the EE can not move to a new position until fault is resolved	It is never the case that r=1 holds when s=1 holds. This condition can be terminated by r=0
8	IfEventuallyAlwaysEventuallyAlways((c_p > L_max OR c_p < L_min), o=1)	If an obstacle is reached by the EE than the obstacle avoidance procedure is performed	It is always the case that if c_p > L_max OR c_p < L_min holds than sometimes in the future o=l holds
9	Deadlock Freedom	The program does not have deadlocks	The model does not have deadlocks

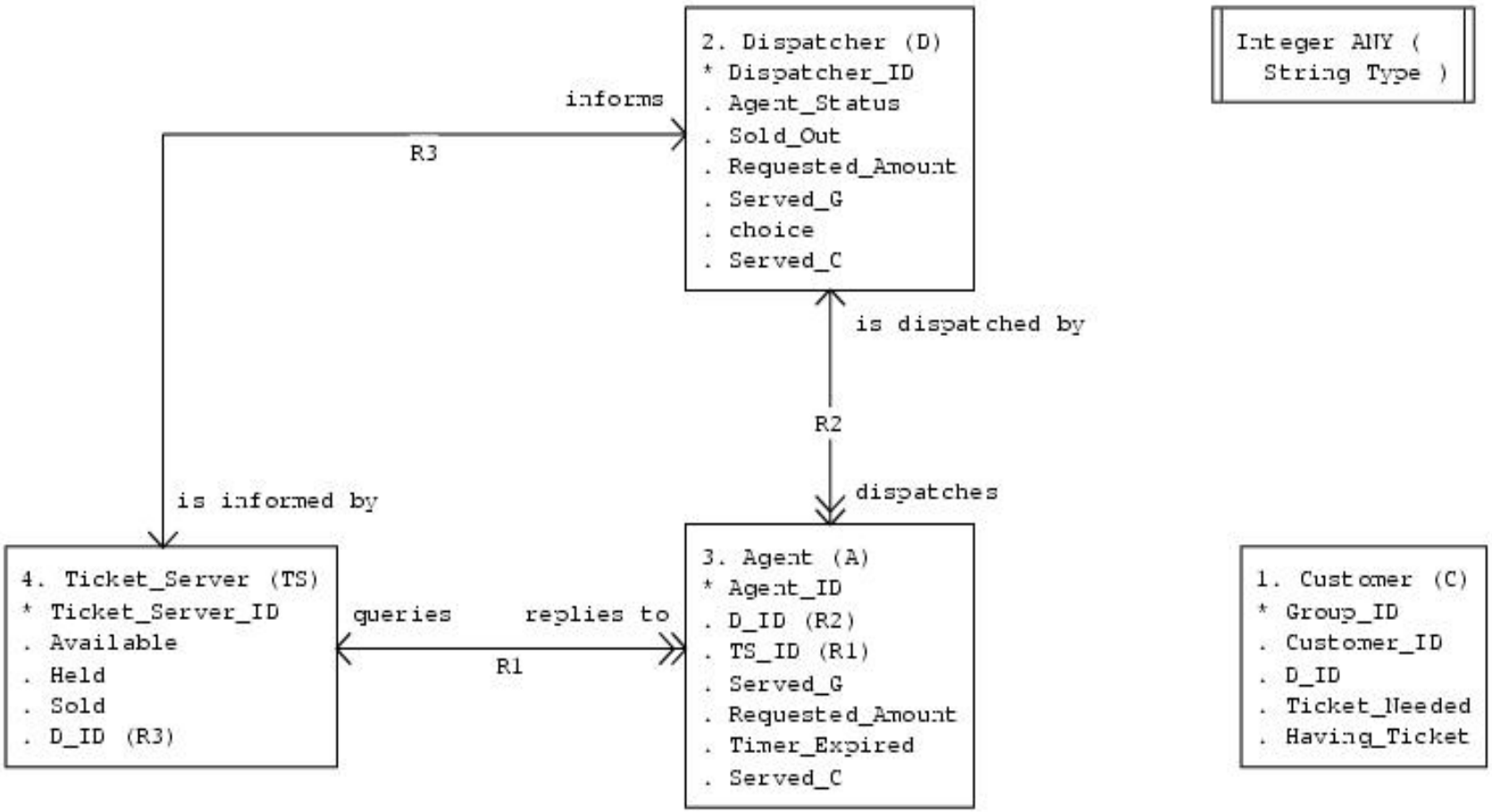
Table 4.5: Verification properties (continued)

N	Property	Robotic Description	Formal Description
10	Never((c_p > L_max OR c_p < L_min) AND r=1)	The robot never operates outside of allowed workspace	It is never the case that c_p < L_max OR c_p > L_min holds and r=l holds
11	Never($ch > v$)	The program always performs computations for an actual robot: the general description of a robot is always reduced to that of an actual robot	At any execution of the program the current value of ch never exceeds the value of the dof parameter
12	NeverUntil(r=l,fk=l)	No command to move the EE is scheduled before an initial EE position is computed	It is never the case that r=l holds until fk=l holds
13	NeverUntil(r=l,(ch=v AND $status$=l) OR $status$=0)	The EE does not proceed to a new position until an acknowledgement from the JCH is received	At any execution r=l does not hold until either ch=dof and $status$=l hold or $status$=0 holds
14	IfEventuallyAlwaysEventually((ch=v AND $status$=l),r=l)	The robot arm always follows the specified trajectory	If during the execution of the program ch=dof AND $status$=l holds than sometime in the future r=l holds
15	AfterEventually(NOT rec = 0, p=l)	Chained fault recovery is not permitted (if the fault recovery did not complete for an instance of the robot configuration, the fault recovery for a different robot configuration instance is not allowed)	At any execution if rec=0 holds than sometime in the future p=l holds
16	Never(r=l AND c_p = $obstacle_{1i}$ AND ... AND c_p = $obstacle_{ni}$)	EE is never located at some undesired locations, $obstacle_{ni}$, where n is the number of pre-defined EE positions	At any execution r=l and c_p = $obstacle_{1i}$ AND ... AND c_p = $obstacle_{ni}$ are never possible at the same time
17	Always((sp != 0) → fac=l)	Only validated solutions of TrailConfigurations are used for the optimization of the robot control	At any time when $optimal\_solution$ is not 0, fac=l holds

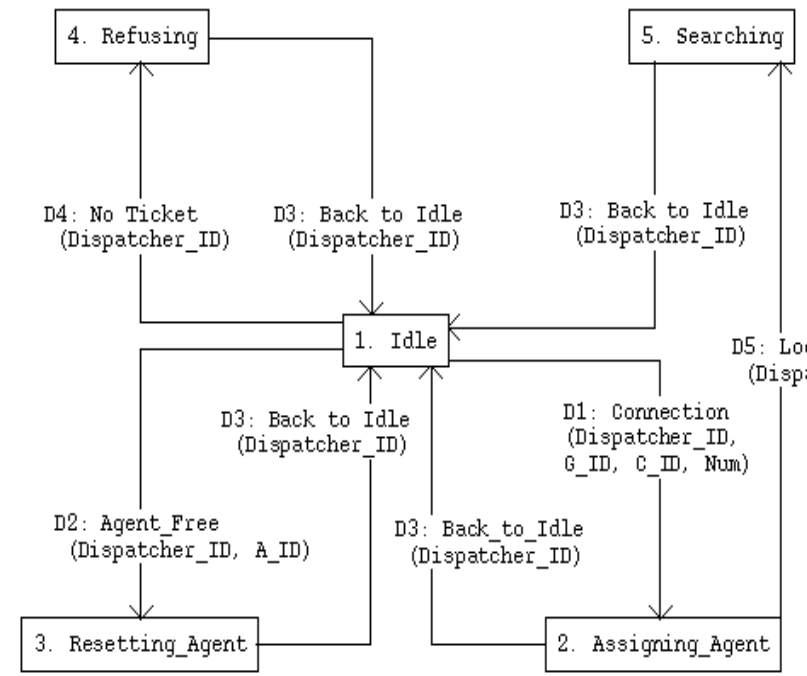
Model Checking of Online Ticket Sale System

- A typical commercial transaction system;
- Presented at FASE 2002;
- Focus: Integrated state space reduction.

An Online Ticket Sale System (Class Diagram)



An Online Ticket Sale System (A State Model)




```
Sold_Out = TRUE;

Generate D3: Back_to_Idle (Dispatcher_ID);
```

```
Agent_Status[A_ID - 1] = TRUE;

Generate D3: Back_to_Idle (Dispatcher_ID);
```

```
if ((Agent_Status[0] == TRUE)&&(Agent_Status[1] == TRUE)) {
  choice = ANY("A_ID_TYPE");
  Agent_Status[choice - 1] = FALSE;
  Generate A1: Assignment (choice , Served_G, Served_C, Requested_Amount);
}
else if ((Agent_Status[0] == TRUE)&&(Agent_Status[1] == FALSE)){
  Agent_Status[0] = FALSE;
  Generate A1: Assignment (1, Served_G, Served_C, Requested_Amount);
}
else if ((Agent_Status[0] == FALSE)&&(Agent_Status[1] == TRUE)){
  Agent_Status[1] = FALSE;
  Generate A1: Assignment (2, Served_G, Served_C, Requested_Amount);
}
else
  Generate C5: Try_Later (Served_G, Served_C);

Generate D3: Back_to_Idle(Dispatcher_ID);
```

```
if (Sold_Out == TRUE) {
  Generate C4: No_Ticket (G_ID, C_ID);
  Generate D3: Back_to_Idle (Dispatcher_ID);
 }
else {
  Served_G = G_ID; Served_C = C_ID;

  Requested_Amount = Num;
  Generate D5: Loop (Dispatcher_ID);
 }
```

Some Verification Statistics of Online Ticket Sale System

- Verification of a liveness property
  - After an agent is assigned to a customer, eventually the agent will be released.
- Statistics related to state space reductions

SPOR	SMC	Memory Usage	Time Usage
Off	Off	Out of Memory	-
Off	On	113.73M	44736.S
On	Off	17.3M	6668.3S
On	On	74.0M	1450.3S

Presentation Outline

- Overview and Architecture of ObjectCheck
- Modeling and Verification of a TinyOS Run-time Image with ObjectCheck
- More Case Studies
- Summary and Future Work
- Work Built on ObjectCheck

Summary and Future Work

- ObjectCheck
  - Integrates industrial software development environments and model checkers with research tools;
  - Provides comprehensive automation for development, validation, and model checking of xUML models;
  - Has enabled verification of non-trivial software system designs modeled in xUML.
- Future work is focused on
  - State space reduction capability of ObjectCheck;
  - Hardware/software co-design and co-verification.

Related Work

- Most closely related work
  - UML Model Checking toolset from *University of Michigan;*
  - vUML tool from *Åbo Akademi University;*
- There is also related work on model checking of statecharts with different semantics.

Additional Information

- http: //www.cs.utexas.edu/users/ObjectCheck
- Selected publications:
  - Fei Xie and James C. Browne. Verified Systems by Composition from Verified Components. Submitted for review.
  - Fei Xie, James C. Browne, and Robert P. Kurshan. Translation-based Compositional Reasoning for Software Systems. Submitted for review.
  - Fei Xie and James C. Browne. Integrated State Space Reduction for Model Checking Executable Object-oriented Software System Designs. In Proc. of FASE 2002.
  - Fei Xie, Vladimir Levin, and James C. Browne. ObjectCheck: A Model Checking Tool for Executable Object-oriented Software System Designs. In Proc. of FASE 2002.
  - Fei Xie, Vladimir Levin, and James C. Browne. Model Checking for an Executable Subset of UML. In Proc. of ASE, 2001.

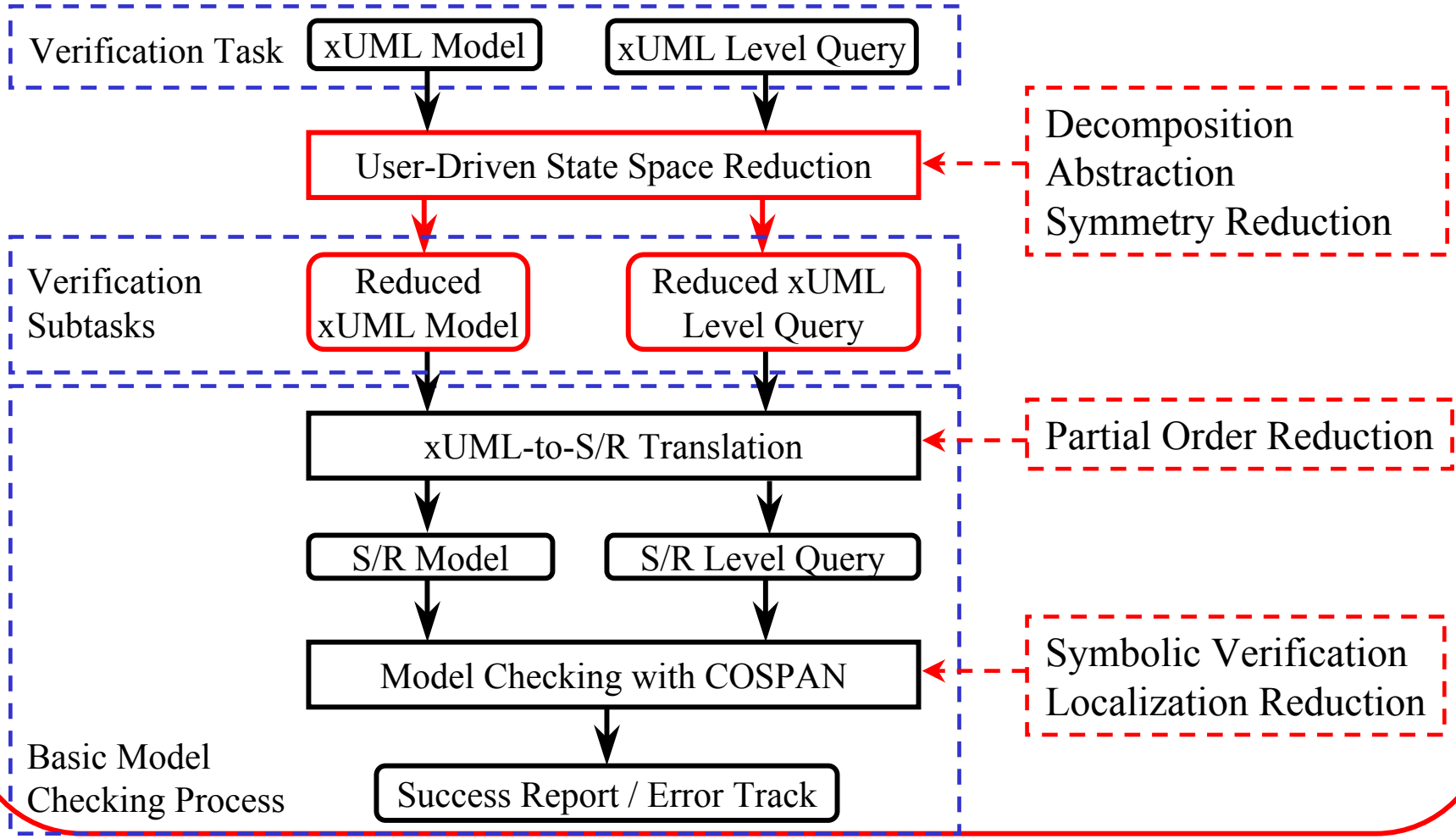
Presentation Outline

- Overview and Architecture of ObjectCheck
- Modeling and Verification of a TinyOS Run-time Image with ObjectCheck
- More Case Studies
- Summary and Future Work
- Work Built on ObjectCheck

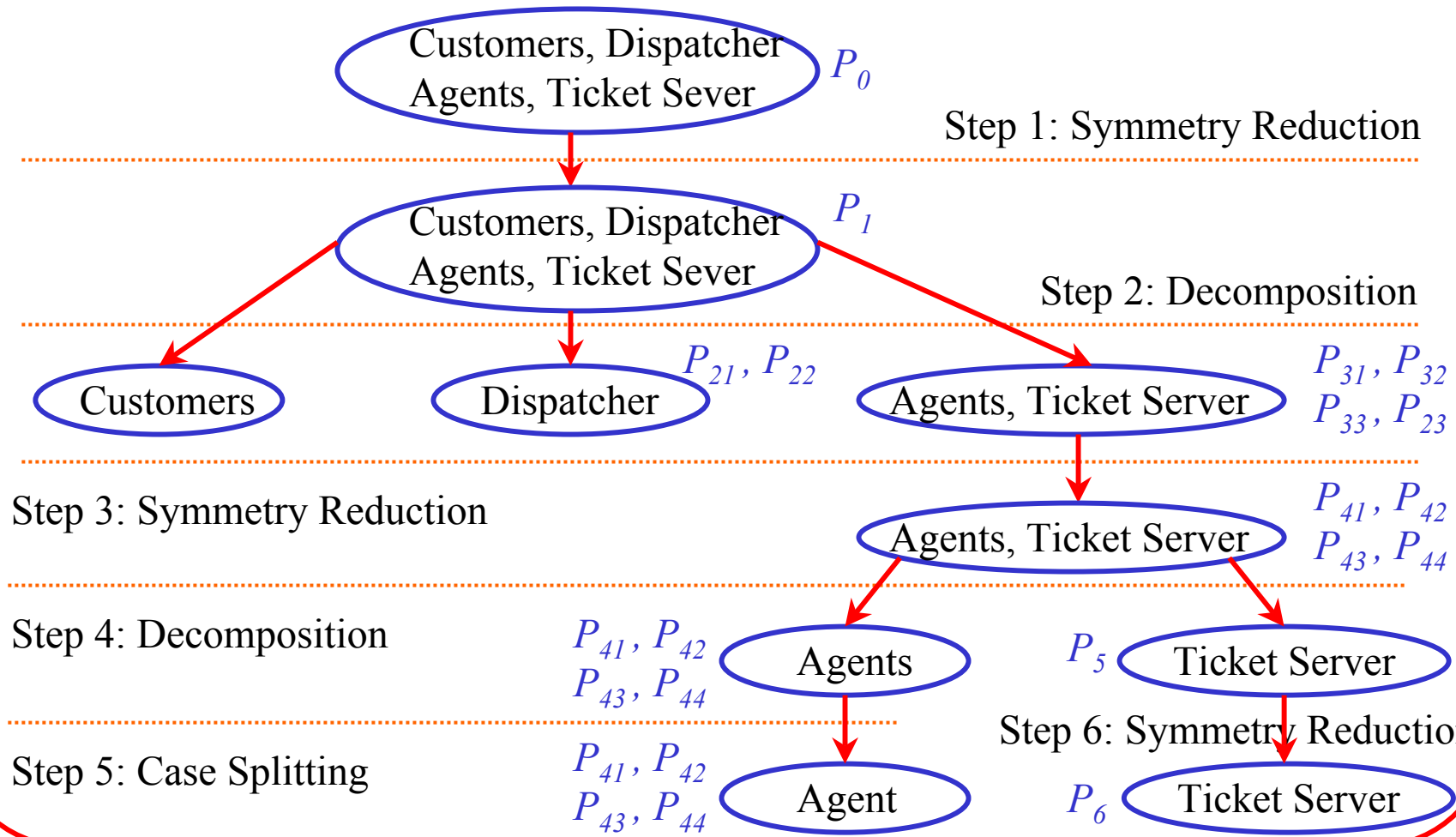
Work Built on ObjectCheck

- An integrated state space reduction framework;
- Integration of model checking into component-based development of software.

Integrated State Space Reduction Framework



Reduction Steps for Checking P_0



Evaluation of User-driven State Space Reduction

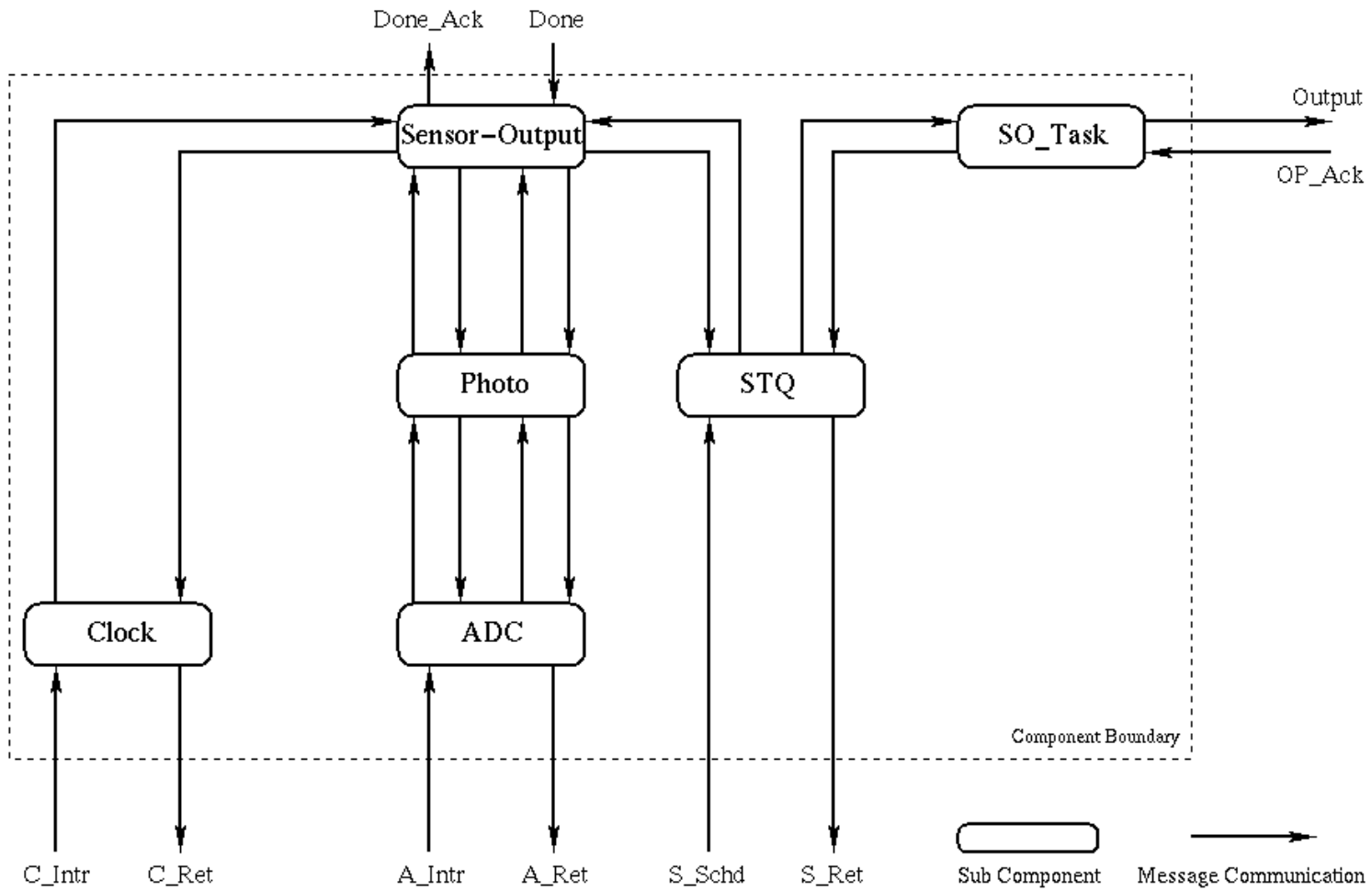
- Directly model checking P_0 on OTSS
  - Two customer instances and two agent instances;
  - SPOR and SMC are both applied.
  - Memory usage: 152.79M
  - Time usage: 16273.7S
- Memory and time usages for discharging subtasks at the leaf nodes of the reduction tree.

	P_{21}	P_{22}	P_{41}	P_{42}	P_{43}	P_{44}	P_6
Memory	0.30M	0.95M	0.28M	0.29M	0.28M	0.29M	0.35M
Time	0.02S	1.81S	0.01S	0.04S	0.01S	0.04S	0.63S

Integration of Model Checking into Component-based Development

- Temporal properties of a component are
  - Established with assumptions on the environment of the component;
  - Verified under these assumptions and then packaged with the component.
- Selecting a component for reuse considers not only its functionality but also its temporal properties.
- Properties of a composed component are verified by reusing verified properties of its sub-components and applying compositional reasoning.

Sensor Component



Properties of Sensor Component

Properties:

Repeatedly (Output);
After (Output) Never (Output) UntilAfter (OP_Ack);
After (Done) Eventually (Done_Ack);
Never (Done_Ack) UntilAfter (Done);
After (Done_Ack) Never (Done_Ack) UntilAfter(Done);

Assumptions:

After (Output) Eventually (OP_Ack);
Never (OP_Ack) UntilAfter (Output);
After (OP_Ack) Never (OP_Ack) UntilAfter (Output);
After (Done) Never (Done) UntilAfter (Done_Ack);
Repeatedly (C_Intr);
After (C_Intr) Never (C_Intr + A_Intr + S_Schd) UntilAfter (C_Ret);
After (ADC.Pending) Eventually (A_Intr);
After (A_Intr) Never (C_Intr + A_Intr + S_Schd) UntilAfter (A_Ret);
After (STQ.Empty = FALSE) Eventually (S_Schd);
After (S_Schd) Never (C_Intr + A_Intr + S_Schd) UntilAfter (S_Ret);