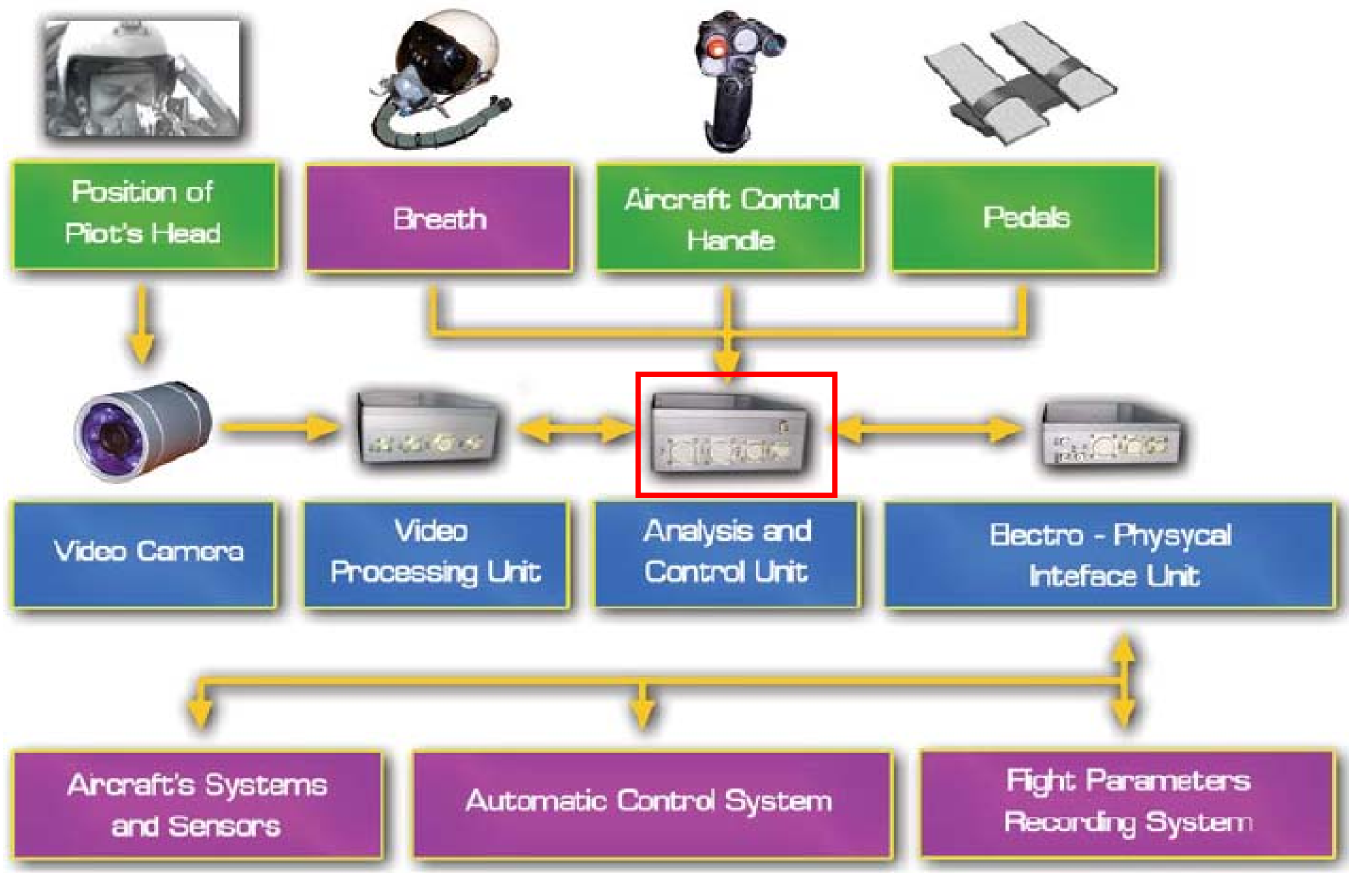
MBT 2012

Model-Based Testing of Safety Critical Real-Time Control Logic Software

Yevgeny Gerlits, Alexey Khoroshilov

Institute for System Programming of RAS

AAFSS – Airborne Active Flight Safety System



- AAFS sensors - AAFS units - Aircraft's embedded systems

1. AAFSS is developed by Russian Systems Corporation. The picture belongs to Russian Systems Corporation. The picture was taken from http://www.rusys.ru

Control Logic Software (CLS) is a subsystem

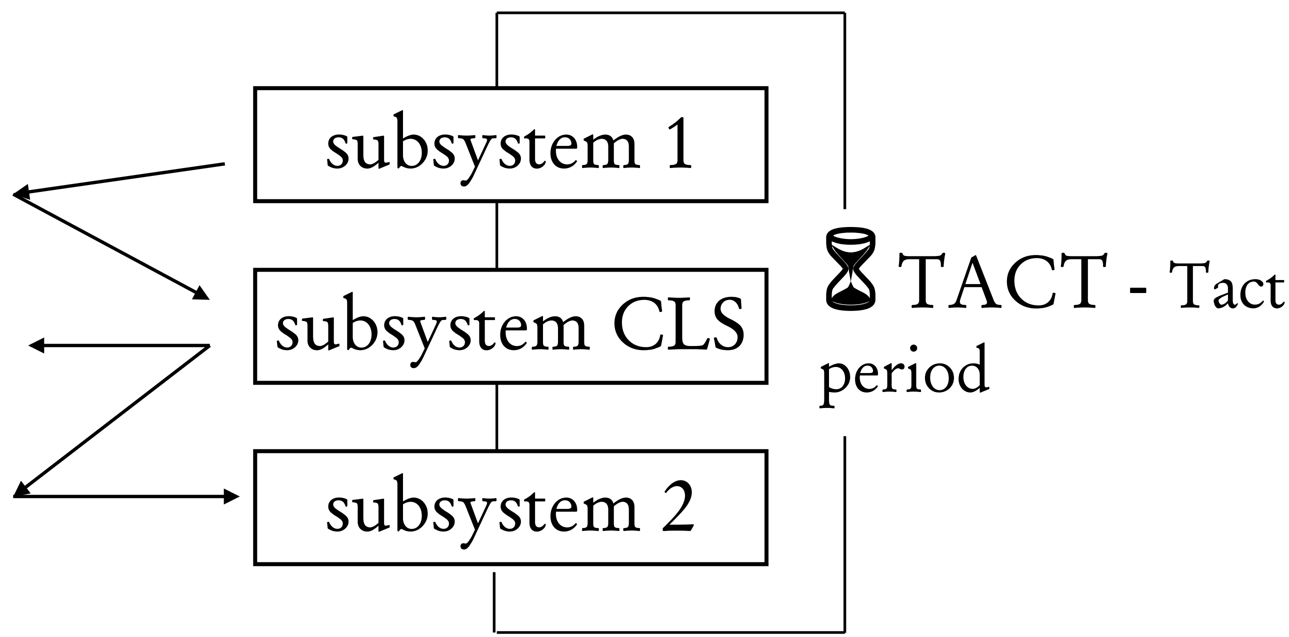
Memory pool:

sys_time

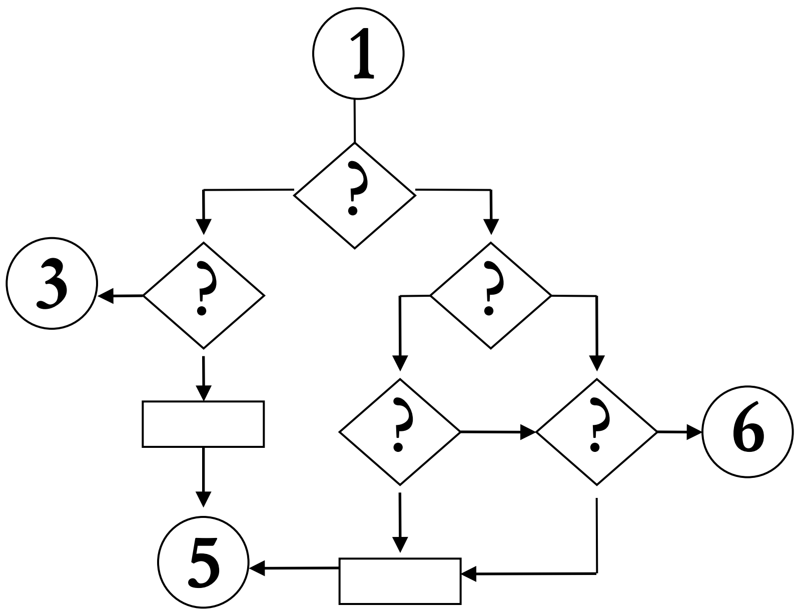
i_1, …, i_n

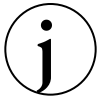
s_1, …, s_m

o_1, …, o_k

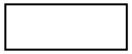


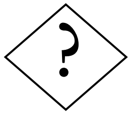
CLS is a number of decision control algorithms





- go to algorithm j;


- set values of output parameters or state variables;



- if condition then… else…

We consider the following conditions:

- Boolean formulas;
- (formula(i_1,..,i_n), T) = true if Boolean formula has been true for T or more.

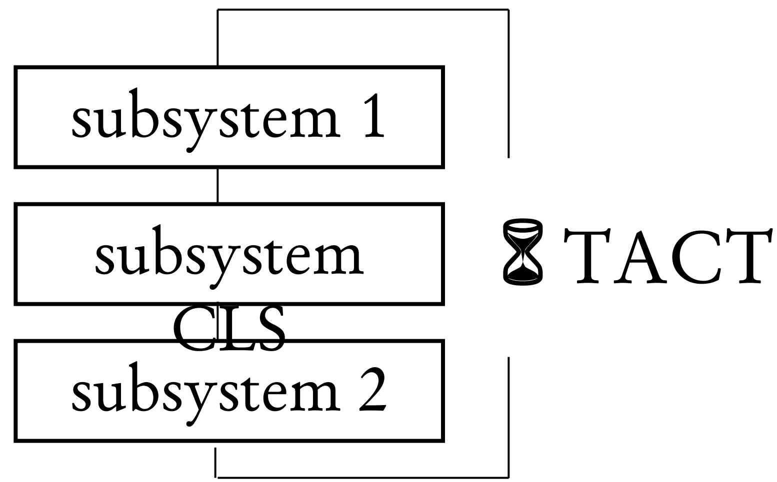
Real Time in CLS

- total time of all subsystems execution < tact period;
- temporal condition (formula(i_1,..,i_n), T) = true if Boolean formula has been true for T or more:

Example: (i_1 < 5, 2*TACT)

Turns of the global control loop:

1. sys_time =0; i_1 = 3;
2. sys_time = TACT; i_1 = 2;
3. **sys_time = 2*TACT; i_1 = 3;**
4. sys_time = 3*TACT; i_1 = 6.



Temporal condition is closer to state than to real time

How does CLS calculate (formula(i_1,..,i_n), T) ?

- Let `sys_time`$_f$ be sys_`time` since when formula(i_1,..,i_n) has been TRUE;
- (formula(i_1,..,i_n), T) = formula(i_1,..,i_n) && (`sys_time - sys_time`$_f$) > = T.

Characteristics of CLS

- ~ 2 000 lines of code;
- Low Level Requirements are 9 flow charts of size A4;
- 32 input parameters of different types;
- 7 state variables of different types;
- 80 temporal conditions in branch instructions;
- 9 output parameters;
- tact period is 60 ms.

Problem Definition

- huge number of input parameters (32);
- huge space of states (7 state variables + 80 temporal conditions);
- CLS is a safety critical software (MC/DC metric)

⇒ Traditional unit testing doesn't work well.

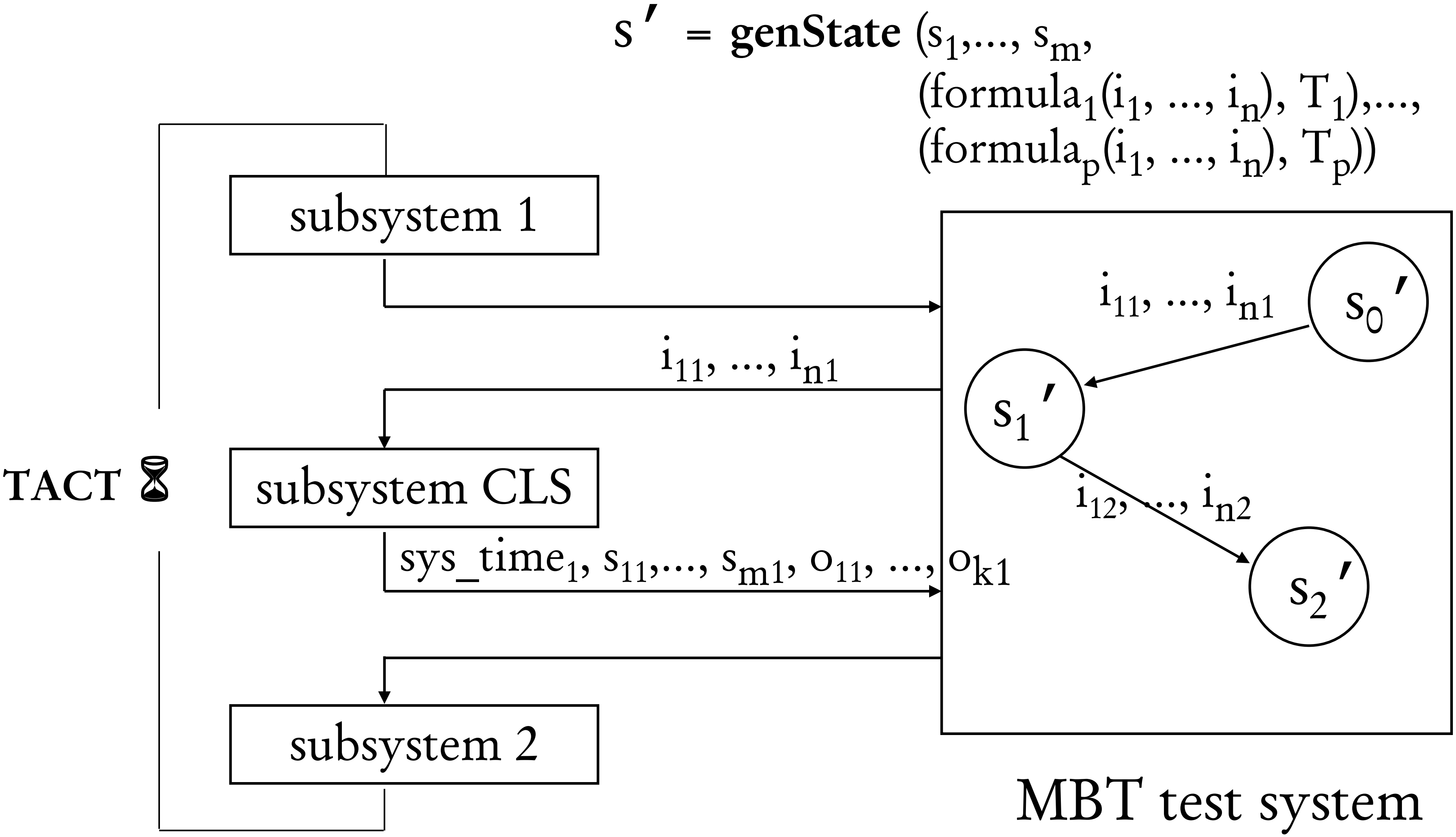
- real time characteristics of CLS are not complicated

⇒ <u>Real Time specific MBT approaches (UPAAL Tron, Timed TorX) are not ultimately required.</u>

- industrial tool is required in a real project

⇒ <u>try a general purpose MBT (SpecExplorer, UniTESK).</u>

On the Fly MBT Approach



Oracle

$i_{11}, \ldots, i_{n1}$

$s_{10}, \ldots, s_{m0}$

$(formula_{10}(i_1, \ldots, i_n), T_{10}), \ldots,$
$(formula_{p0}(i_1, \ldots, i_n), T_{p0}))$

CLS under test

$(formula_{11}(i_1, \ldots, i_n), T_{11}), \ldots,$
$(formula_{p1}(i_1, \ldots, i_n), T_{p1}))$

$s_{11}, \ldots, s_{m1}$

$o_{11}, \ldots, o_{k1}$

$i_{11}, \ldots, i_{n1}$

$s_{11}, \ldots, s_{m1}$

$o_{11}, \ldots, o_{k1}$

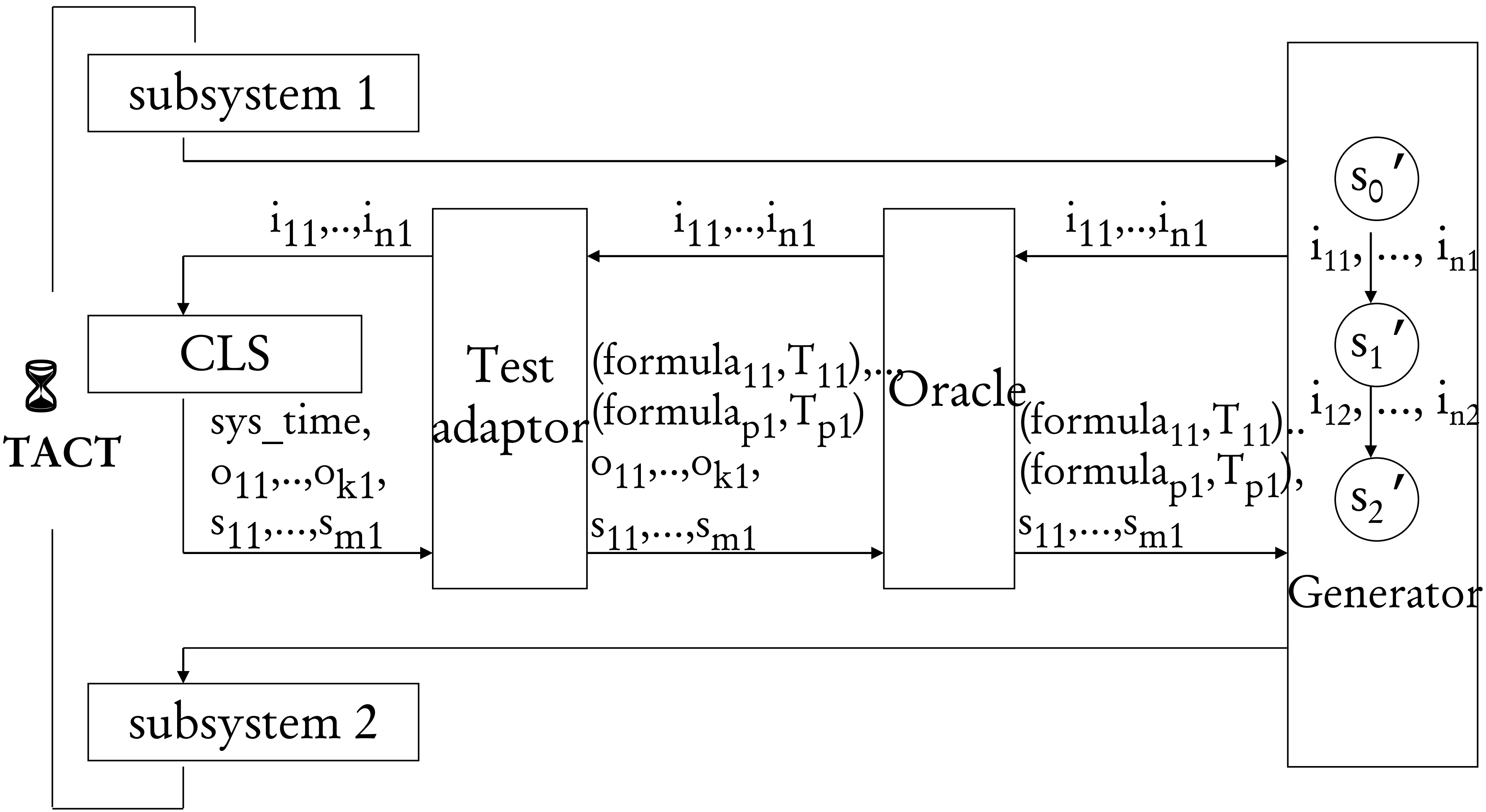
$(formula_{11}(i_1, \ldots, i_n), T_{11}), \ldots,$
$(formula_{p1}(i_1, \ldots, i_n), T_{p1}))$

$s_{10}, \ldots, s_{m0}$

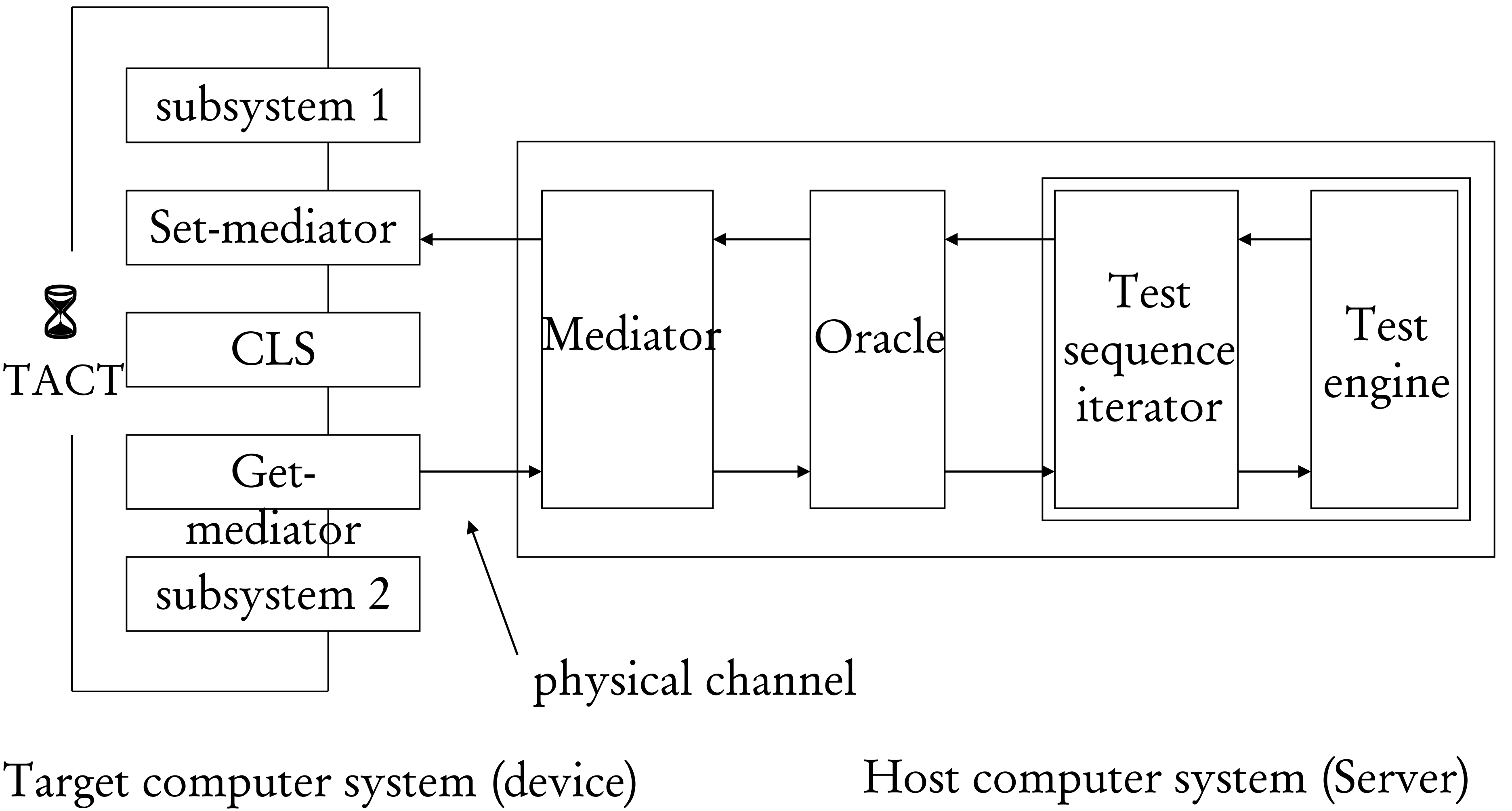
Oracle

{true, false}

MBT Scheme


subsystem 1
$i_{11},..,i_{n1}$
$i_{11},..,i_{n1}$
$i_{11},..,i_{n1}$
s_0'
$i_{11}, \ldots, i_{n1}$
CLS
Test adaptor
$(formula_{11},T_{11}),..,$
$(formula_{p1},T_{p1})$
$o_{11},..,o_{k1},$
$s_{11},\ldots,s_{m1}$
Oracle
$(formula_{11},T_{11})..$
$(formula_{p1},T_{p1}),$
$s_{11},\ldots,s_{m1}$
s_1'
$i_{12}, \ldots, i_{n2}$
s_2'
TACT
sys_time,
$o_{11},..,o_{k1},$
$s_{11},\ldots,s_{m1}$
Generator
subsystem 2

UniTESK MBT Scheme



Conclusion

1. The RTCLS subsystem and the architecture of the whole embedded device were described;
2. An MBT approach to RTCLS was outlined in general terms;
3. The MBT approach was implemented using UniTESK.

⇒ general purpose MBT like UniTESK are applicable to CLS.

Thank you! Questions?