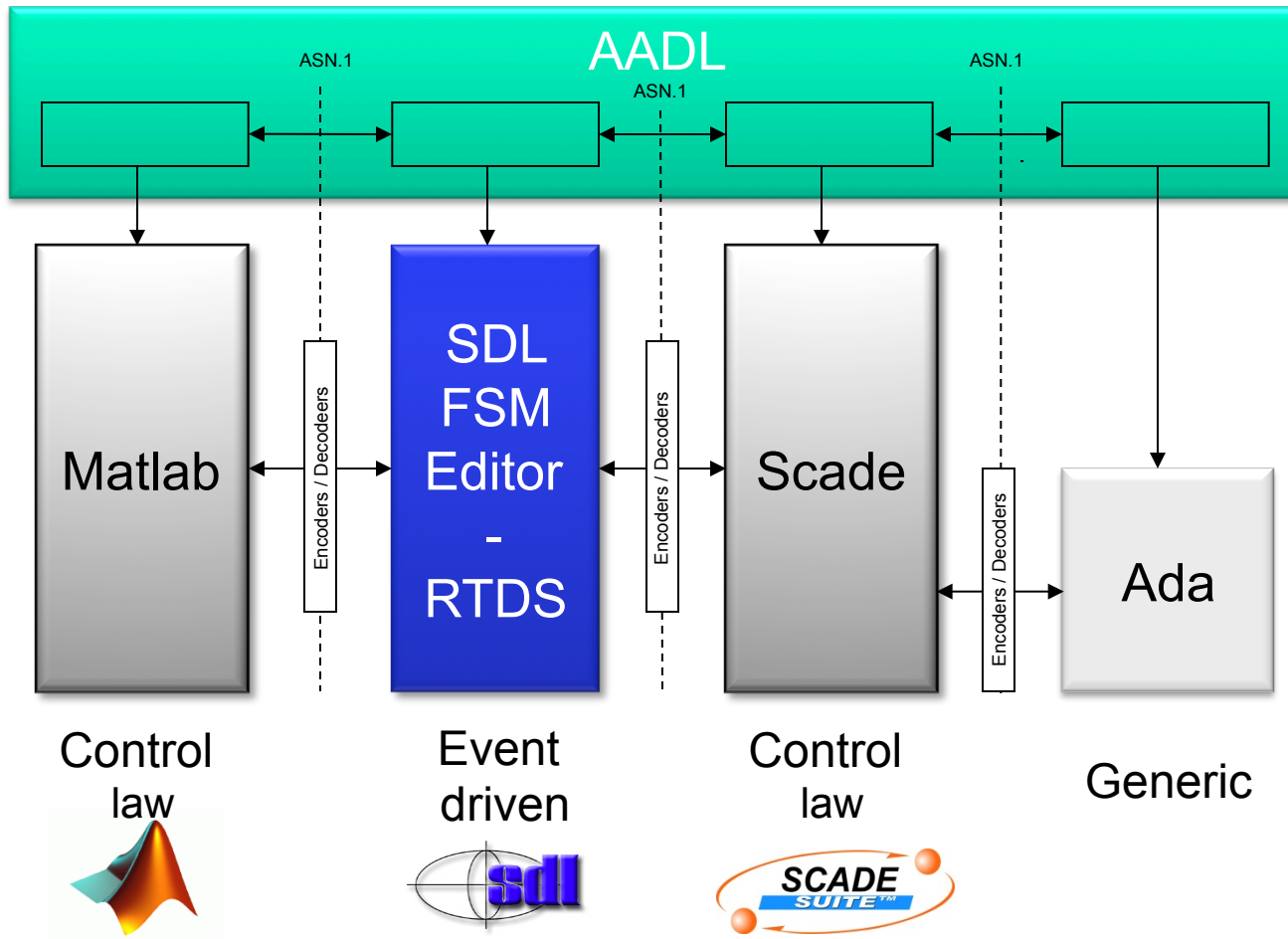


PragmaDev SDL FSM Editor & Real Time Developer Studio within the TASTE framework

ESA – May 22th, 2014

Eric Brunel
eric.brunel@pragmadev.com

ESA Taste framework integration





- Specification and Description Language, ITU-T standard intended to write the detailed specifications of the protocols in order to make sure they will interoperate.
- Major updaters every 4 years since 1976.
- Major releases:
 - SDL 1988: First mature version
 - SDL 1992: Object orientation
 - SDL 2000: UML alignment
 - SDL 2010: C types support in Z.104
- Annual conference
 - SDL Forum (<http://www.sdl-forum.org/>)
 - SAM workshop (satellite event of Models)
- 11 commercial tools, 10 public domain tools.
- Integrated technology in TASTE ESA framework.

SDL: features

- SDL **graphical abstraction** (architecture, communication, behavior, services) fits the needs.
- SDL being formal, it is possible to **simulate** the model.
- SDL being formal, partial or full **code generation** is possible.
- SDL being **object oriented**, software components are reusable.
- System are globally asynchronous (GALS) so SDL can be used at **system level**.
- SDL has the characteristics to describe **a good PIM**.
- SDL is recognized by certification authorities (European Aviation Safety Agency Certification Memorandum, ETSI, ESA)

SDL: the figures

Years of experience allows to quantify gains of SDL usage.

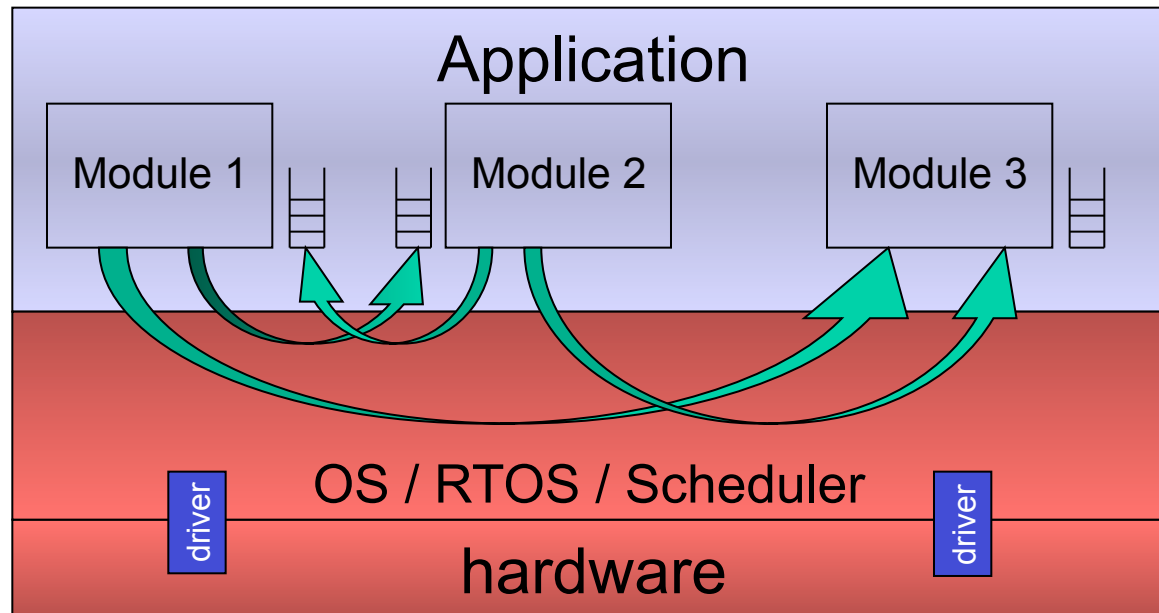
- C code: 35 to 50 mistakes per 1000 lines
- **SDL** code: **8** mistakes per 1000 lines
- Development time is globally reduced by **35%**
 - Reduced up to 50% in the left branch of the V cycle
 - Less gain on the right side of the V because of the gap with technical reality

Target segment

Event driven systems

Embedded & Real time

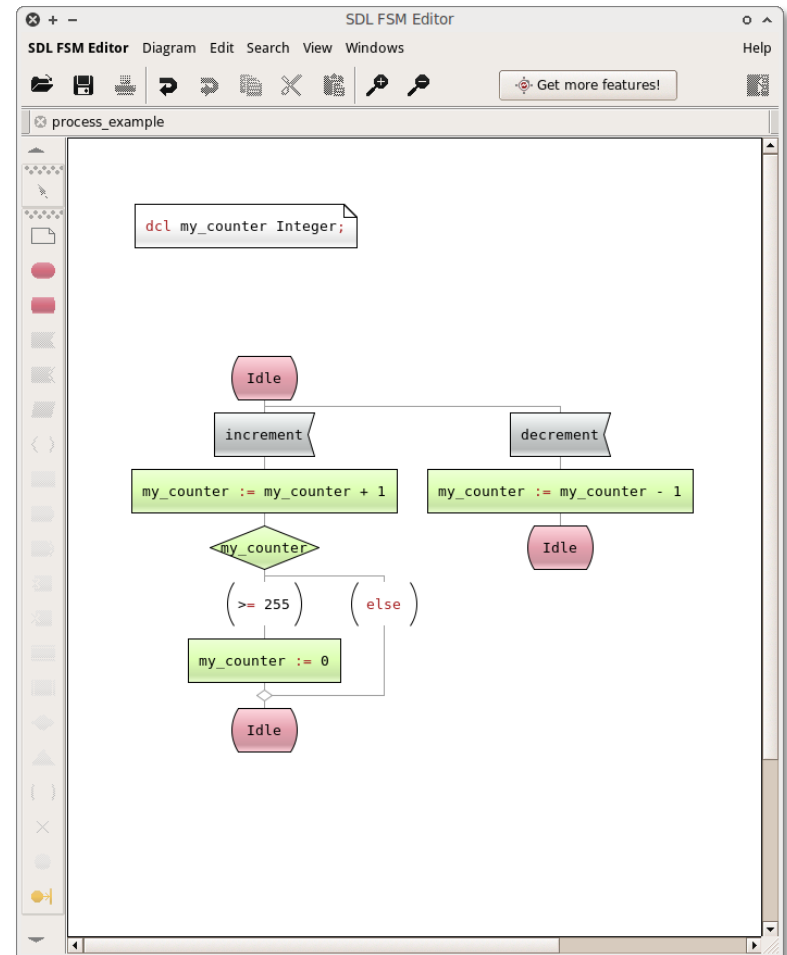
- Decomposed in tasks running concurrently
- Communicating via messages (sporadic or cycling in TASTE)



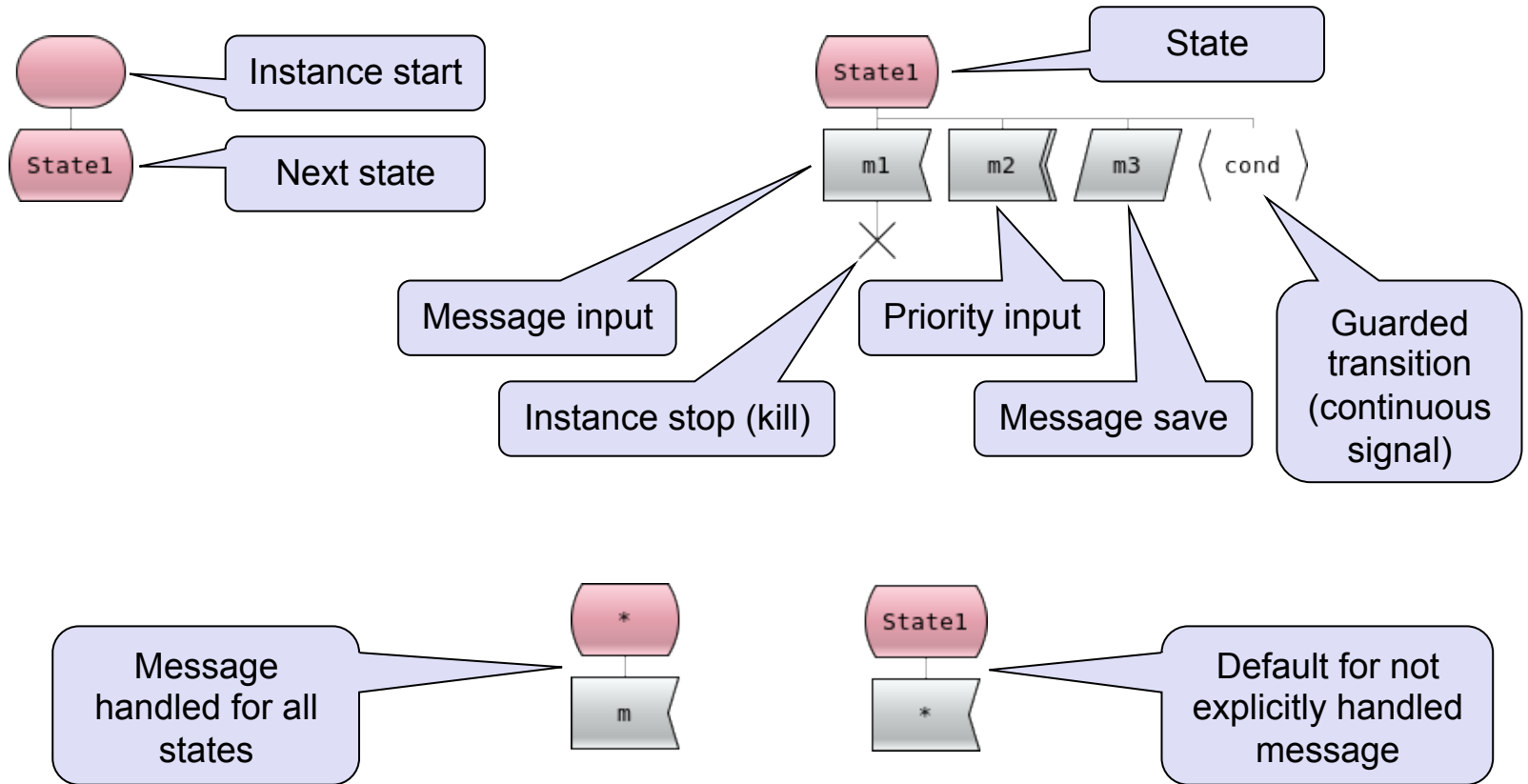
real time developer studio

SDL FSM Editor

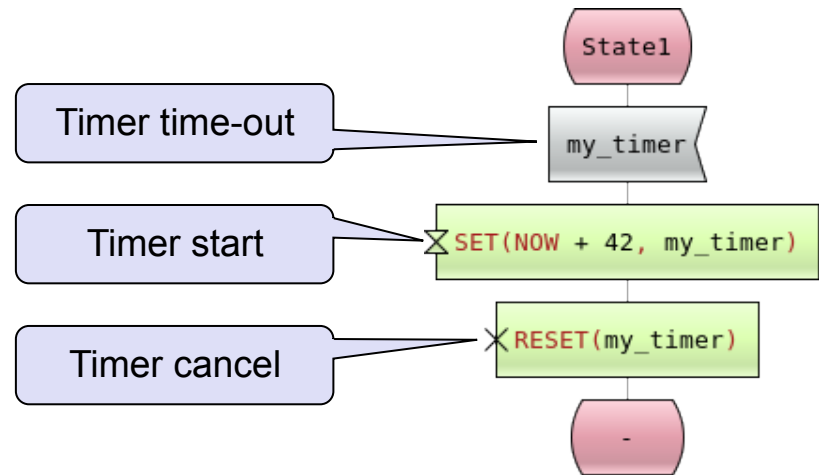
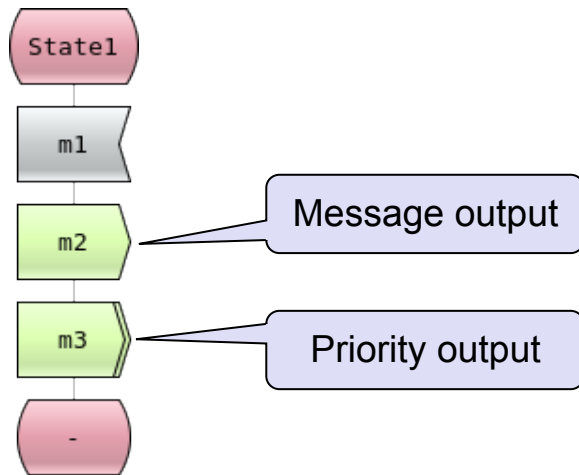
- ASN.1 or SDL abstract data types.
- Syntax highlighting
- Code completion (limited)
- Context-sensitive action availability
- Logical-unit based editing
- Operation preview



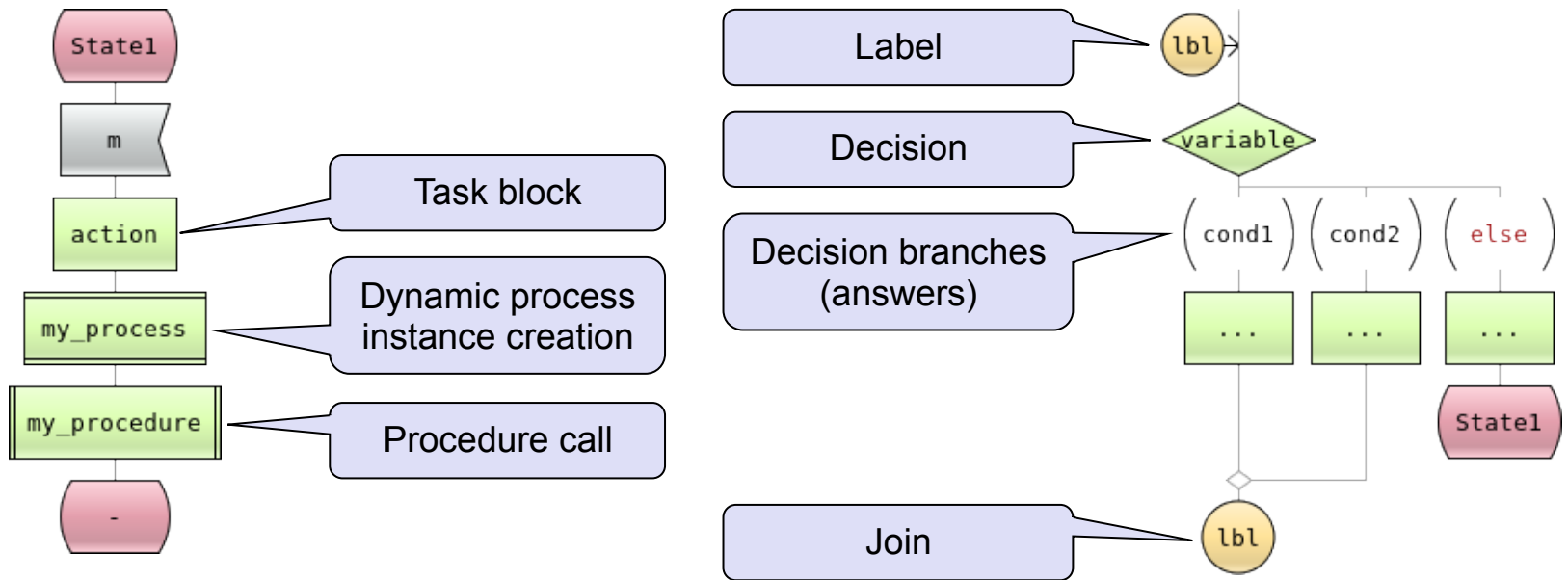
SDL concepts in processes (1)



SDL concepts in processes (2)

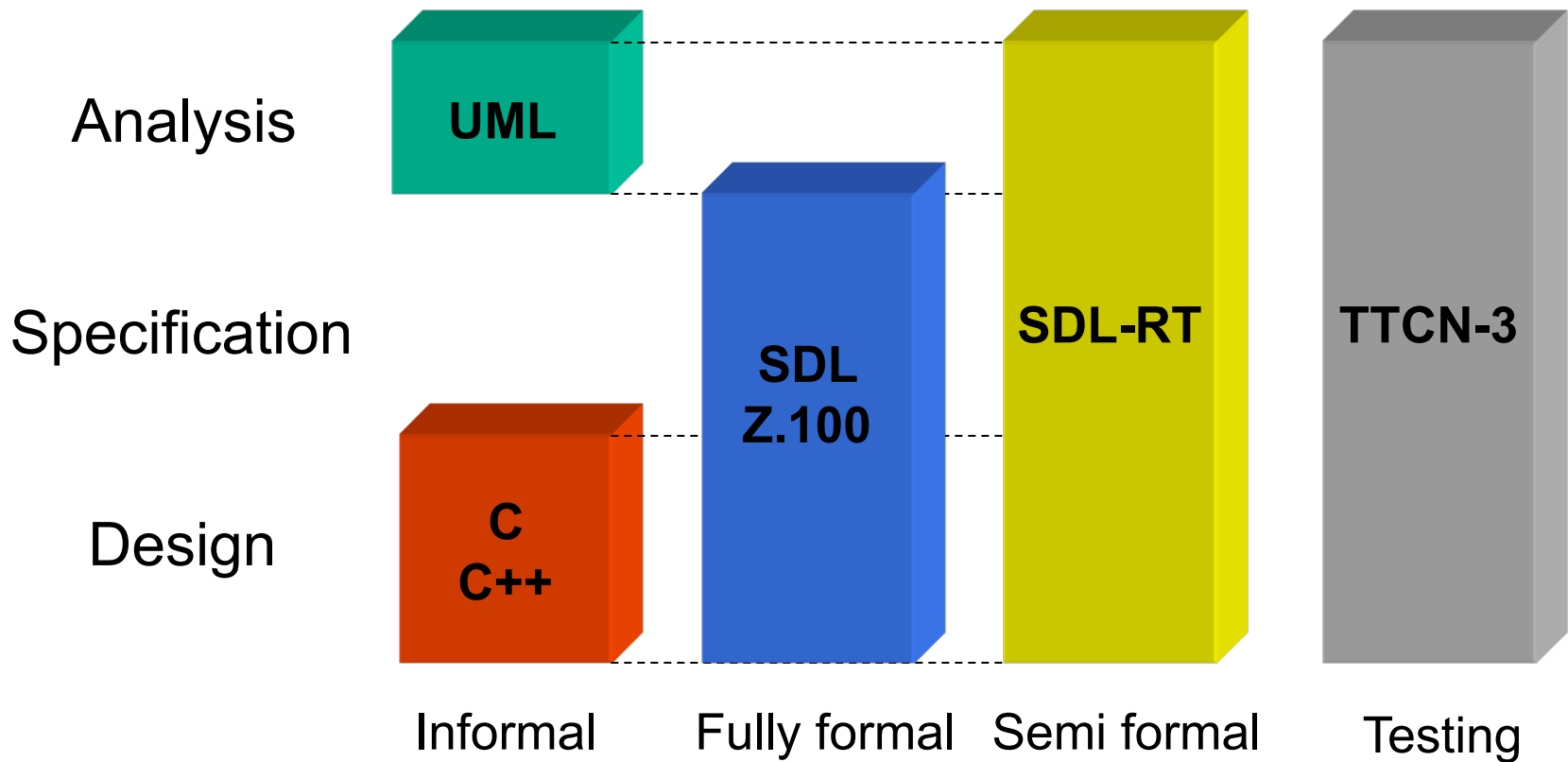


SDL concepts in processes (3)



More “standard” control flow statements available in task blocks (`for`, `break / continue`, `if`, ...).

RTDS: supported languages



RTDS: supported languages

Informal modelling for requirements: UML

- Edition
- C++ stubs generation



Semi-formal modelling for design: SDL-RT

- Edition
- Syntactic et semantics checking
- Code generation
- Graphical debugging

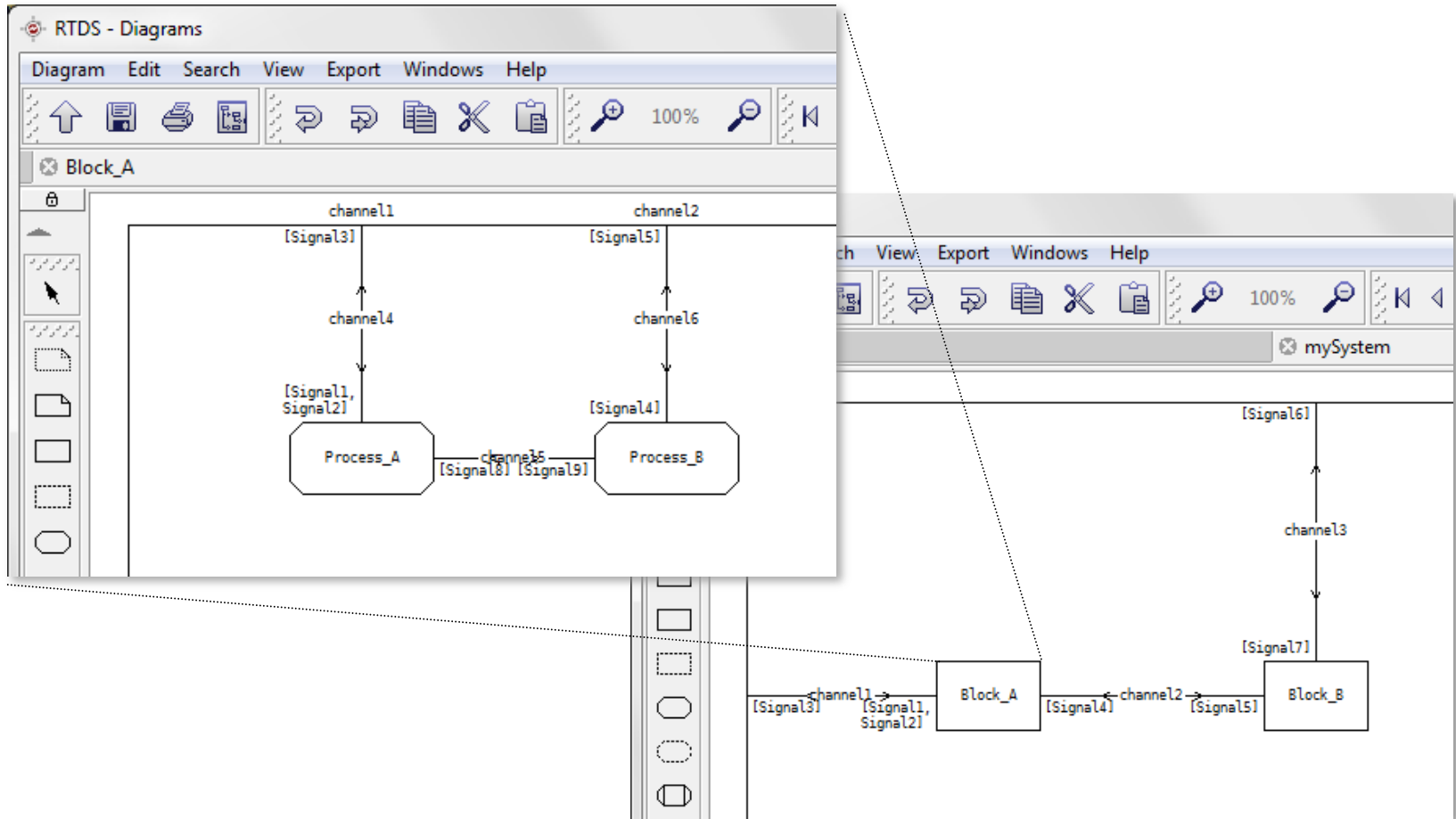


Fully formal modelling for specification: SDL Z.100

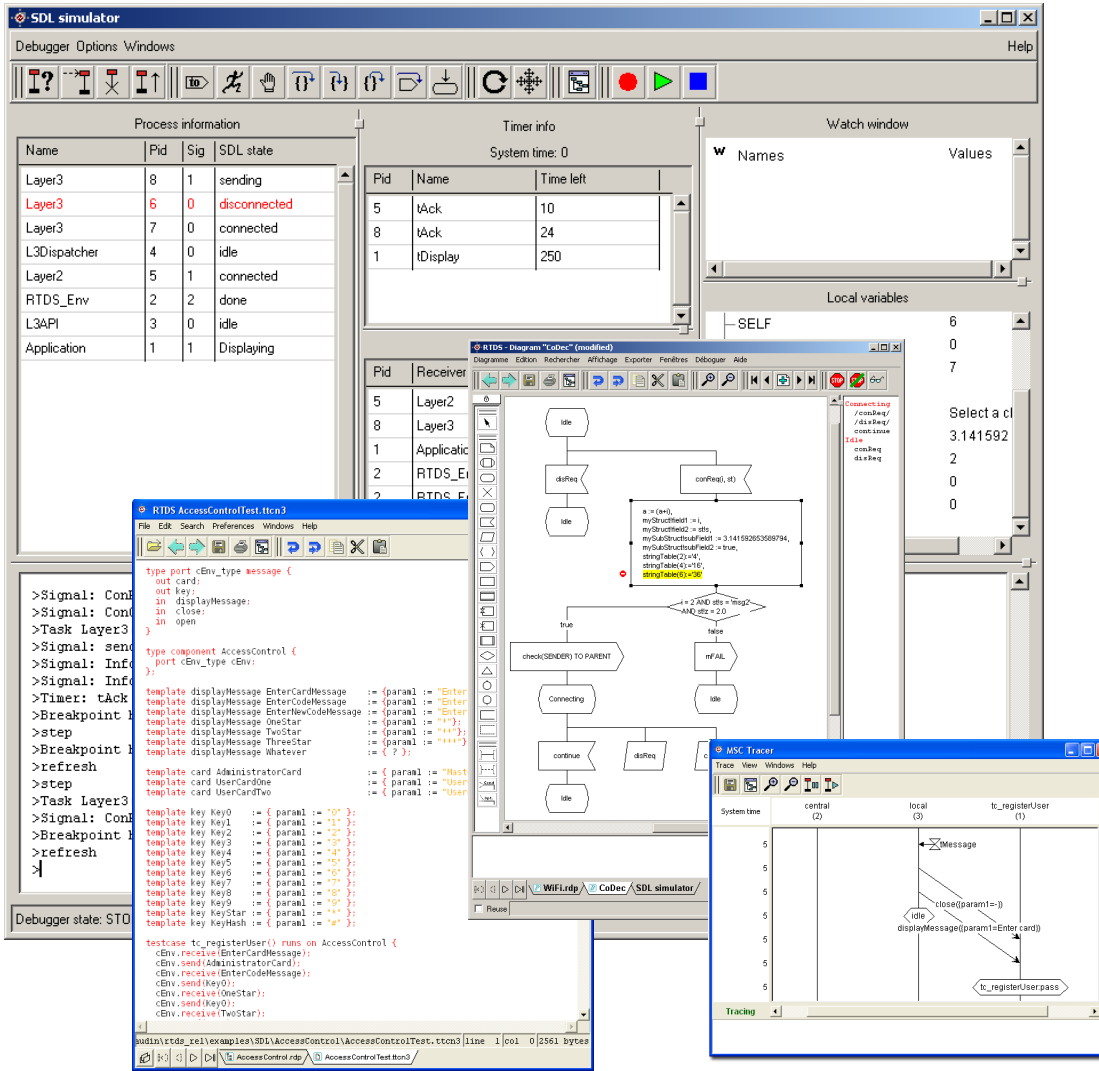
- Edition
- Syntactic et semantics checking
- Simulation
- Verification
- Code generation
- Graphical debugging
- Test



Architecture and communication



real time developer studio



The screenshot displays the SDL simulator interface with several panels:

- Process information:** A table showing the state of various processes.

Name	Pid	Sig	SDL state
Layer3	8	1	sending
Layer3	6	0	disconnected
Layer3	7	0	connected
L3Dispatcher	4	0	idle
Layer2	5	1	connected
RTDS_Env	2	2	done
L3API	3	0	idle
Application	1	1	Displaying
- Timer info:** Shows system time as 0 and a list of timers with their names and time left.

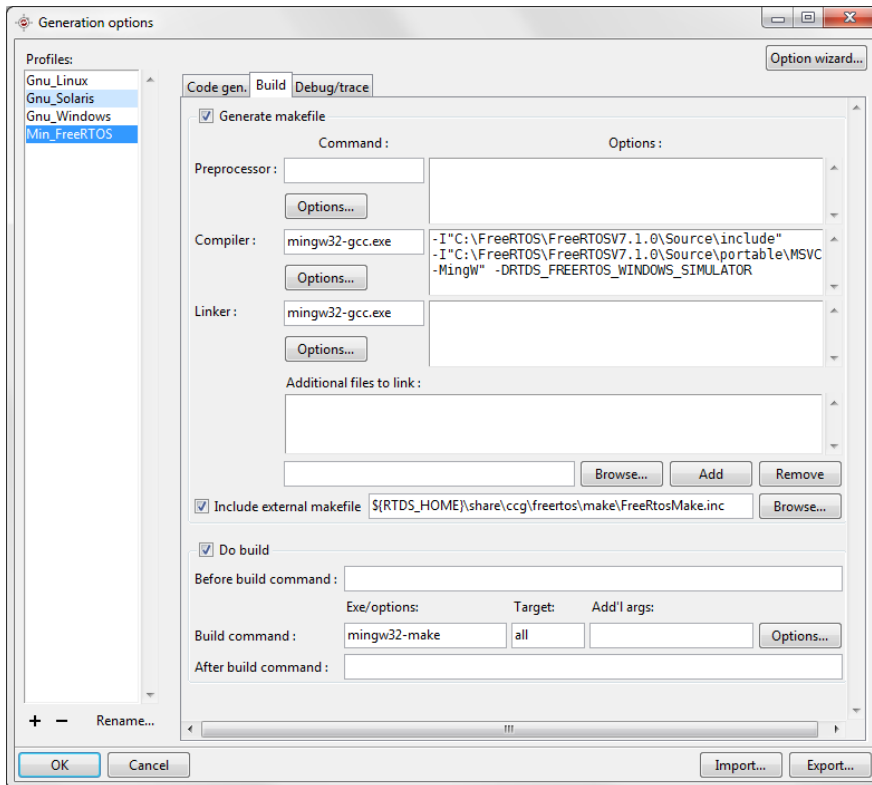
Pid	Name	Time left
5	tAck	10
8	tAck	24
1	tDisplay	250
- Watch window:** Shows a variable named 'Names' with values 6, 0, 7.
- Local variables:** Shows a variable named '-SELF' with values 6, 0, 7.
- Diagram:** A state machine diagram for 'RIDS - Diagram "CoDec" (modified)'. It shows states like 'idle', 'connecting', and 'displaying', with transitions based on events like 'check(SENDER) TO PARENT' and 'displayMessage'. A decision diamond is visible with conditions like 'AND: str = "reg2"'.
- MSC Tracer:** A sequence diagram showing interactions between 'central (C)', 'local (L)', and 'tc_registerUser (U)'. It includes messages like 'Message', 'close(param1=)', and 'displayMessage(param=Enter card)'. The time axis is labeled 'System time'.
- Debugger Console:** Shows a list of commands and their outputs, such as '>Signal: Con', '>Task Layer3', and '>step'. It also displays the current debugger state as 'SDT'.

Model simulator

A graphical *debugger* for fully formal models and TTCN-3 test cases

- Set breakpoints and step in the model,
- Dynamic traces.

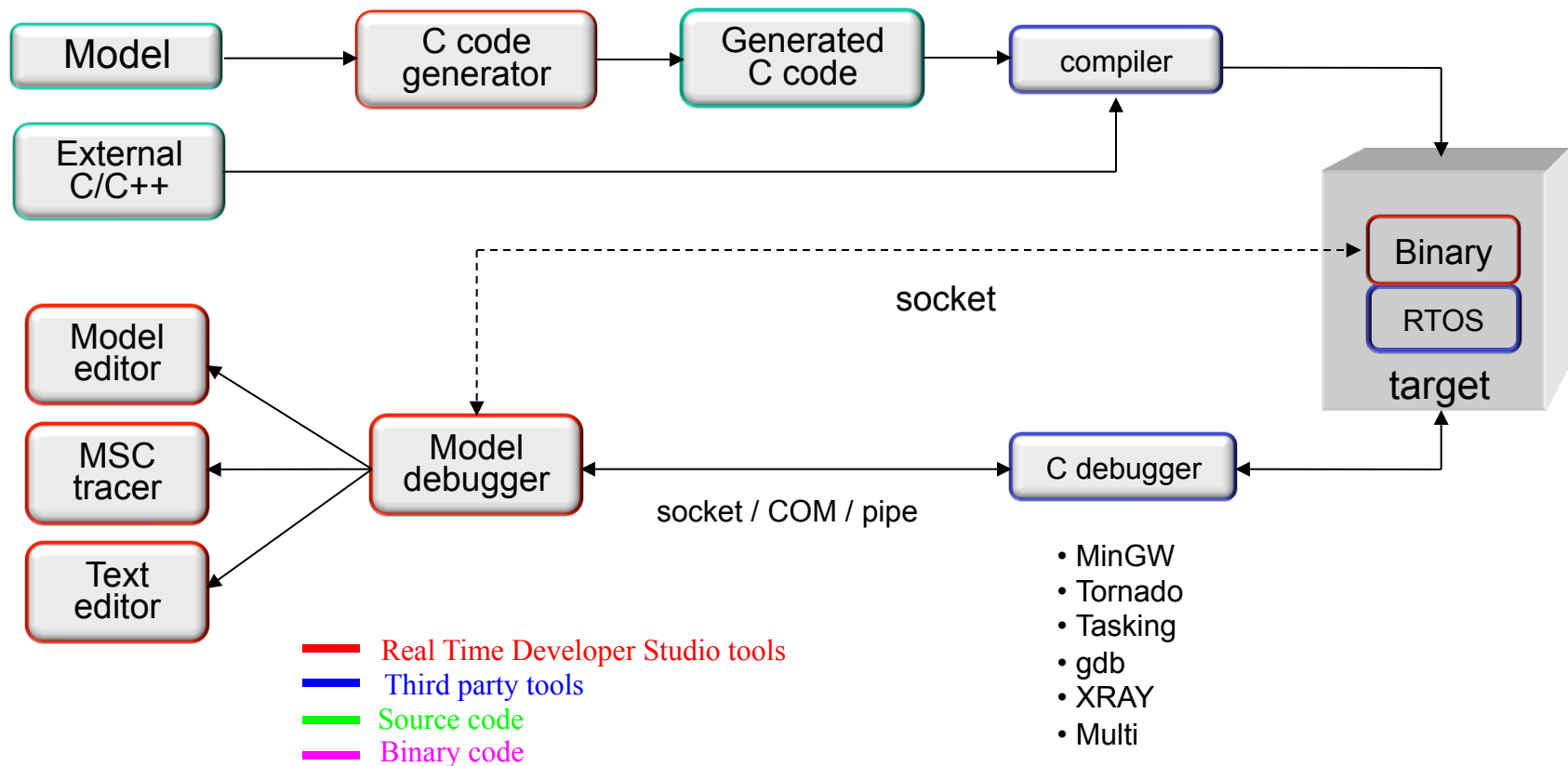
Code generation



- C++ skeleton for static classes
- C or C++ for dynamic classes
- Generated code is legible
- Generation profile wizard
- The code is:
 - Integrated with: FreeRTOS, VxWorks, OSE, OSE Epsilon, CMX RTX, Nucleus, uiTRON, Posix, ThreadX, and Win32,
 - Provided with an scheduler,
 - Royalty free,
 - Documented for customization.

Debugging architecture

The Model debugger relies on a traditional C debugger or cross debugger to provide graphical debugging.

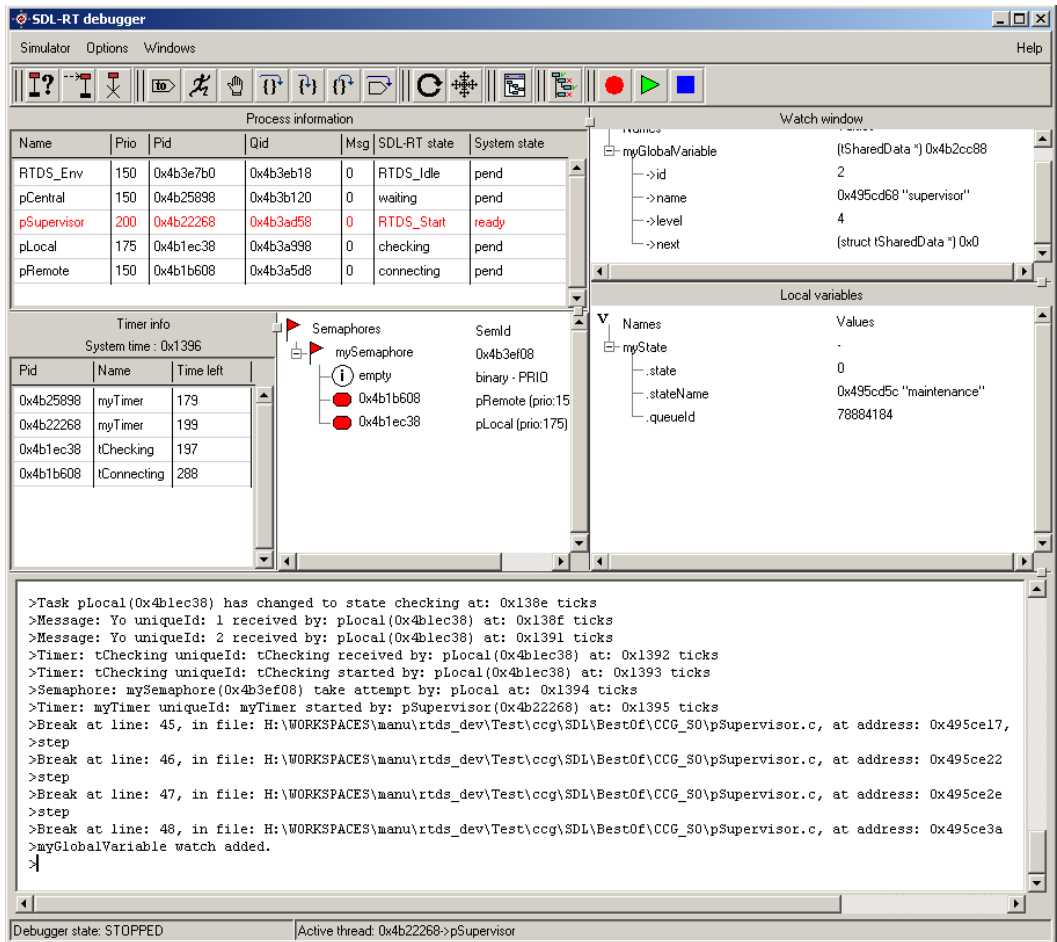


Model debugger

Relies on the target semantic: processor and RTOS.

Debug in the model:

- Breakpoints, stepping, in the SDL/RT diagrams or in the generated C files,
- Dynamic MSC traces,
- Connecting an external tool is possible through a socket.



The screenshot shows the SDL-RT debugger interface with the following components:

- Process information table:**

Name	Prio	Pid	Qid	Msg	SDL-RT state	System state
RTDS_Env	150	0x4b3e7b0	0x4b3eb18	0	RTDS_Idle	pend
pCentral	150	0x4b25898	0x4b3b120	0	waiting	pend
pSupervisor	200	0x4b22268	0x4b3ad58	0	RTDS_Start	ready
pLocal	175	0x4b1ec38	0x4b3a998	0	checking	pend
pRemote	150	0x4b1b608	0x4b3a5d8	0	connecting	pend
- Timer info table:**

Pid	Name	Time left
0x4b25898	myTimer	179
0x4b22268	myTimer	199
0x4b1ec38	tChecking	197
0x4b1b608	tConnecting	288
- Semaphores:**
 - mySemaphore (SemId: 0x4b3ef08, binary - PRIO)
 - empty
 - 0x4b1b608 (pRemote (prio:15))
 - 0x4b1ec38 (pLocal (prio:175))
- Watch window:** myGlobalVariable (ISharedData *) 0x4b2cc88
 - >id: 2
 - >name: 0x495cd68 "supervisor"
 - >level: 4
 - >next: (struct ISharedData *) 0x0
- Local variables:**

Names	Values
myState	.
.state	0
.stateName	0x495cd5c "maintenance"
.queueId	78884184
- Log window:**

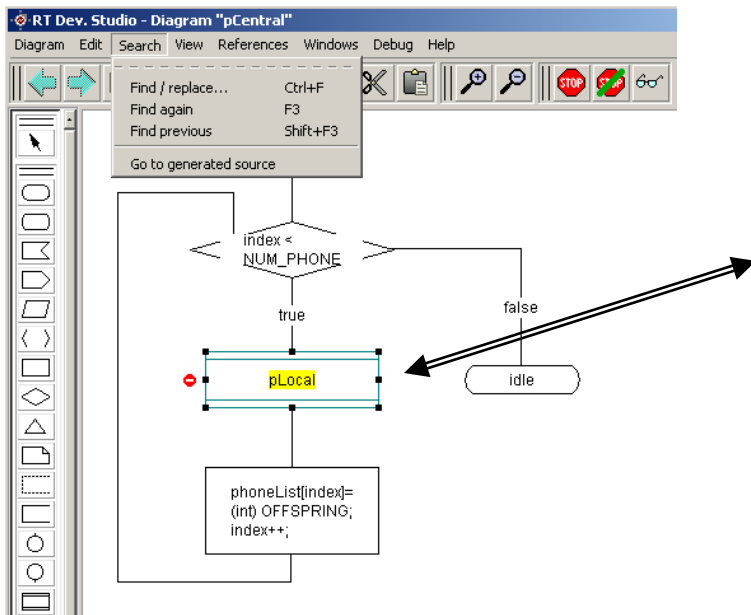
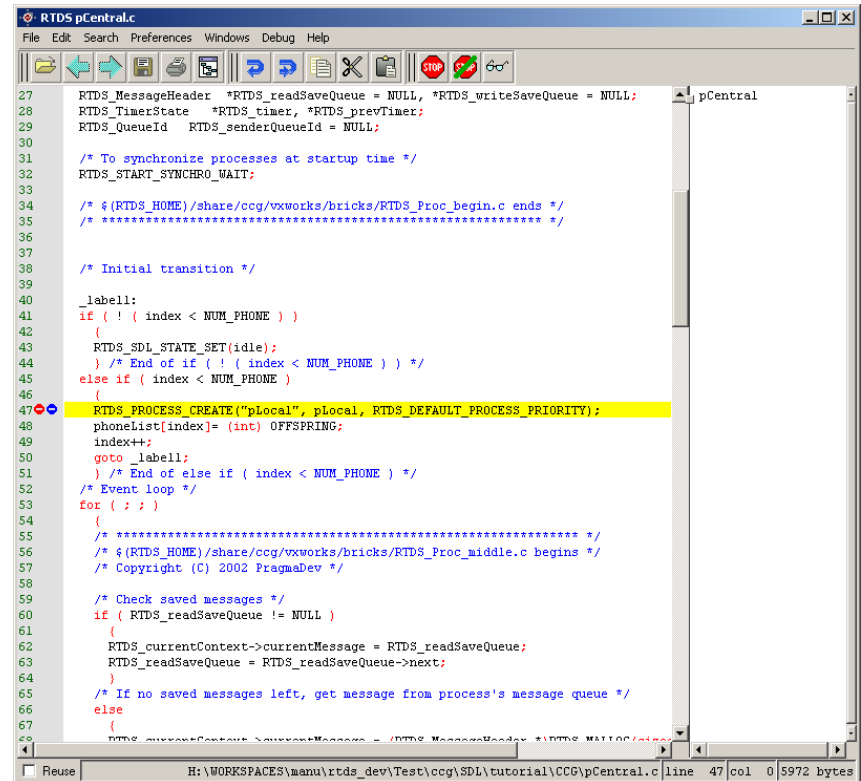
```

>Task pLocal(0x4b1ec38) has changed to state checking at: 0x138e ticks
>Message: Yo uniqueId: 1 received by: pLocal(0x4b1ec38) at: 0x138f ticks
>Message: Yo uniqueId: 2 received by: pLocal(0x4b1ec38) at: 0x1391 ticks
>Timer: tChecking uniqueId: tChecking received by: pLocal(0x4b1ec38) at: 0x1392 ticks
>Timer: tChecking uniqueId: tChecking started by: pLocal(0x4b1ec38) at: 0x1393 ticks
>Semaphore: mySemaphore(0x4b3ef08) take attempt by: pLocal at: 0x1394 ticks
>Timer: myTimer uniqueId: myTimer started by: pSupervisor(0x4b22268) at: 0x1395 ticks
>Break at line: 45, in file: H:\WORKSPACES\manu\rtdev\Test\cog\SDL\BestOf\CCG_S0\pSupervisor.c, at address: 0x495ce17,
>step
>Break at line: 46, in file: H:\WORKSPACES\manu\rtdev\Test\cog\SDL\BestOf\CCG_S0\pSupervisor.c, at address: 0x495ce22
>step
>Break at line: 47, in file: H:\WORKSPACES\manu\rtdev\Test\cog\SDL\BestOf\CCG_S0\pSupervisor.c, at address: 0x495ce2e
>step
>Break at line: 48, in file: H:\WORKSPACES\manu\rtdev\Test\cog\SDL\BestOf\CCG_S0\pSupervisor.c, at address: 0x495ce3a
>myGlobalVariable watch added.
>

```
- Debugger state:** STOPPED | Active thread: 0x4b22268->pSupervisor

Debug features

- Switch between
 - Model
 - Generated C/C++ code

```

27 RTDS_MessageHeader *RTDS_readSaveQueue = NULL, *RTDS_writeSaveQueue = NULL;
28 RTDS_TimerState *RTDS_timer, *RTDS_prevTimer;
29 RTDS_QueueId RTDS_senderQueueId = NULL;
30
31 /* To synchronize processes at startup time */
32 RTDS_START_SYNCHRO_WAIT;
33
34 /* $(RTDS_HOME)/share/ccg/vxworks/bricks/RTDS_Proc_begin.c ends */
35 /* ***** */
36
37 /* Initial transition */
38
39
40 _labell:
41 if ( ! ( index < NUM_PHONE ) )
42 {
43 RTDS_SDL_STATE_SET(idle);
44 /* End of if ( ! ( index < NUM_PHONE ) ) */
45 else if ( index < NUM_PHONE )
46 {
47 RTDS_PROCESS_CREATE("pLocal", pLocal, RTDS_DEFAULT_PROCESS_PRIORITY);
48 phoneList[index]= (int) OFFSPRING;
49 index++;
50 goto _labell;
51 /* End of else if ( index < NUM_PHONE ) */
52 /* Event loop */
53 for ( ; ; )
54 {
55 /* ***** */
56 /* $(RTDS_HOME)/share/ccg/vxworks/bricks/RTDS_Proc_middle.c begins */
57 /* Copyright (C) 2002 PragmaDev */
58
59 /* Check saved messages */
60 if ( RTDS_readSaveQueue != NULL )
61 {
62 RTDS_currentContext->currentMessage = RTDS_readSaveQueue;
63 RTDS_readSaveQueue = RTDS_readSaveQueue->next;
64 }
65 /* If no saved messages left, get message from process's message queue */
66 else
67 {
68 RTDS_currentContext->currentMessage = RTDS_MessageHeader *RTDS_MessageHeader;
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
  
```

Graphical traces

Execution traces:

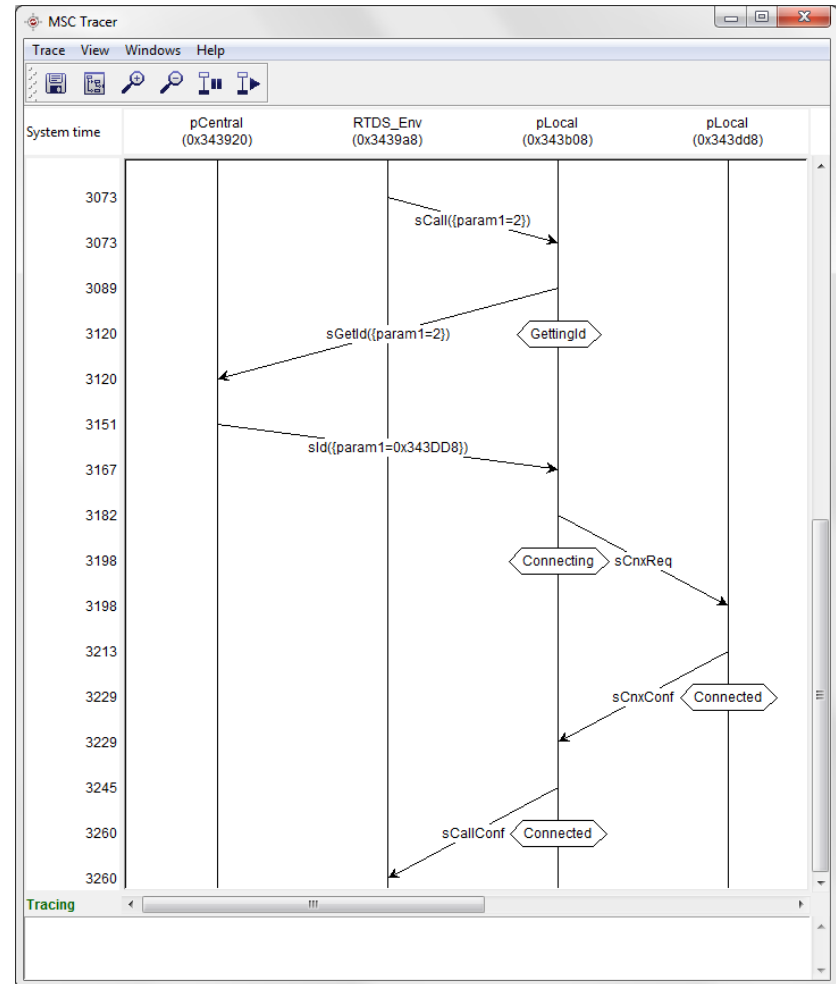
- States,
- Events,
- Semaphores,
- Timers.

Trace level configuration

Display of system time

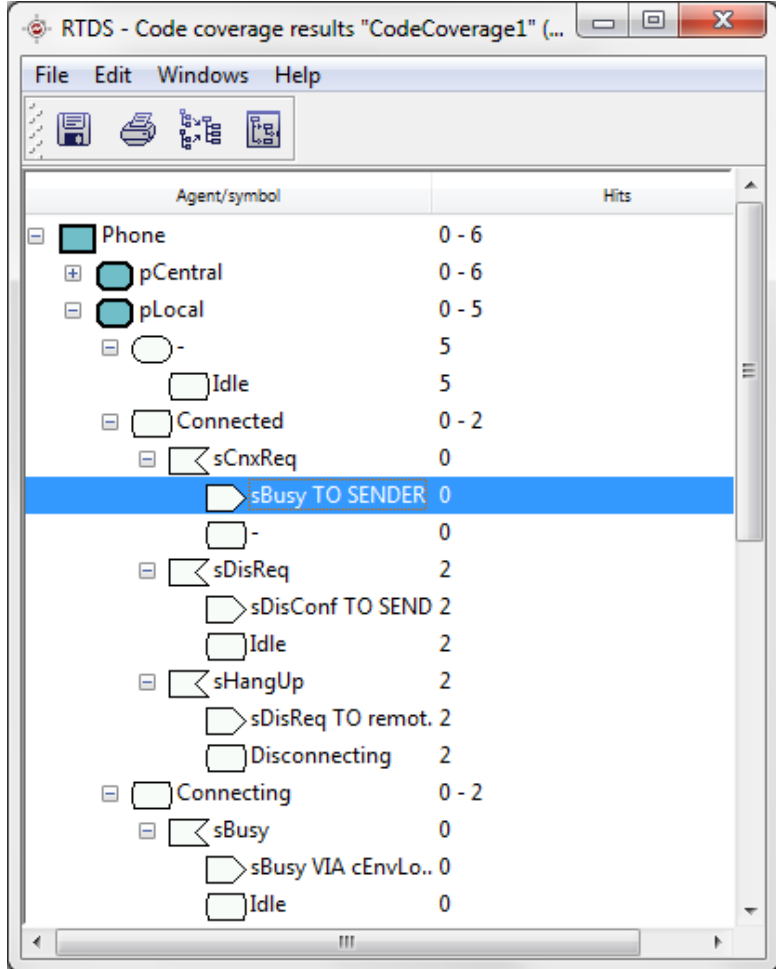
MSC Diff allows to check:

- Conformity,
- Non-regression.



Model coverage

- Graphical model coverage analysis
- Merge feature



The screenshot shows a window titled "RTDS - Code coverage results 'CodeCoverage1' (...)" with a menu bar (File, Edit, Windows, Help) and a toolbar. The main area displays a tree view of model elements with their corresponding hit counts. The element "sBusy TO SENDER" is highlighted in blue.

Agent/symbol	Hits
Phone	0 - 6
pCentral	0 - 6
pLocal	0 - 5
-	5
Idle	5
Connected	0 - 2
sCnxReq	0
sBusy TO SENDER	0
-	0
sDisReq	2
sDisConf TO SEND	2
Idle	2
sHangUp	2
sDisReq TO remot.	2
Disconnecting	2
Connecting	0 - 2
sBusy	0
sBusy VIA cEnvLo..	0
Idle	0

Prototyping interface

SDL Platform

Debugger

Timer info
System time: 0

Owner Name Time left

SDL system queue

Watch variables Values

Local variables Values

Strobe_Switch StrobeSwitch

Simulation trace Env.

System time

AB (1) EL_CO (2) E_SB (3)

running running running

RTDS - Diagrams

Diagram ESR Search View Export Windows Debug Help

EL_CO

allGearAndGround * True

running

running

SetSTROBE(Strobe_Switch)

Strobe_Switch

StrobeSwitch ON StrobeSwitch OFF StrobeSwitch

SDL model tracking

testGUI

EXT LT

STROBE ON REACTOR ON NAV ON LOGO ON

WING ON LDG ON NOISE ON

WOW_True WOW_False

Simulator GUI

➤ Knows about the model inputs and outputs.

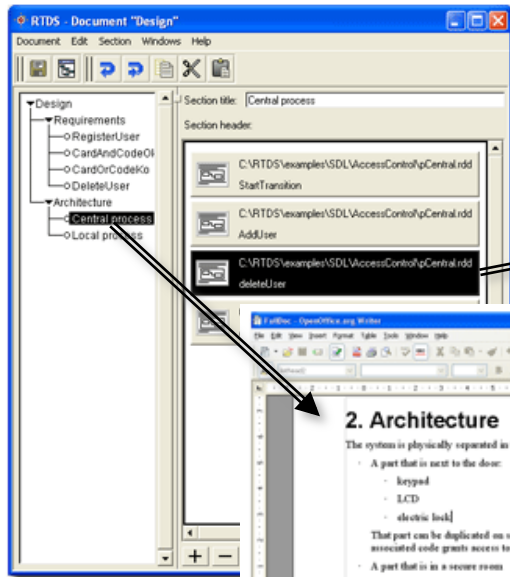
➤ Connects automatically to the simulator or the debugger.

real time developer studio

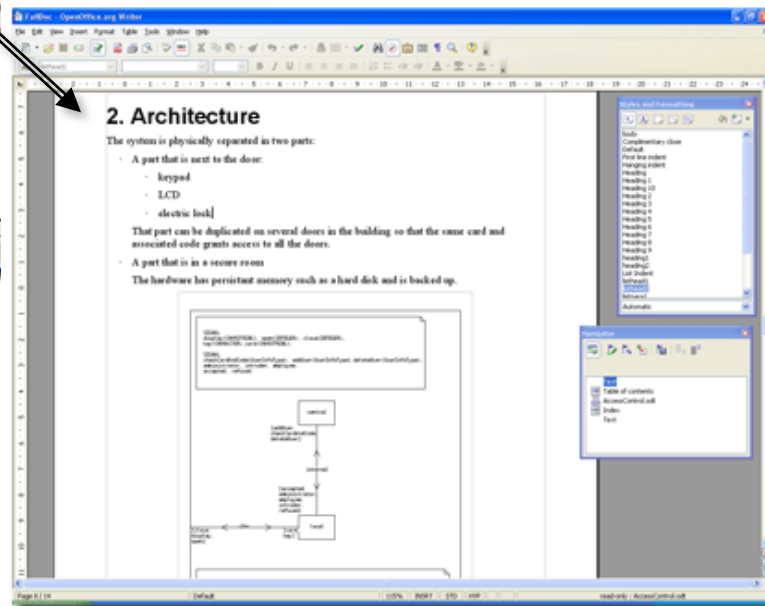
Documentation generation

- Logical publications (state, transition, partition, diagram)
- Comments preceding or following the publication
 - Styles for paragraphs
 - Styles for characters
- Export format
 - RTF
 - OpenDocument
 - HTML
 - SGML
- Exported elements
 - Texts with publications
 - Index entries
 - Table of contents entries

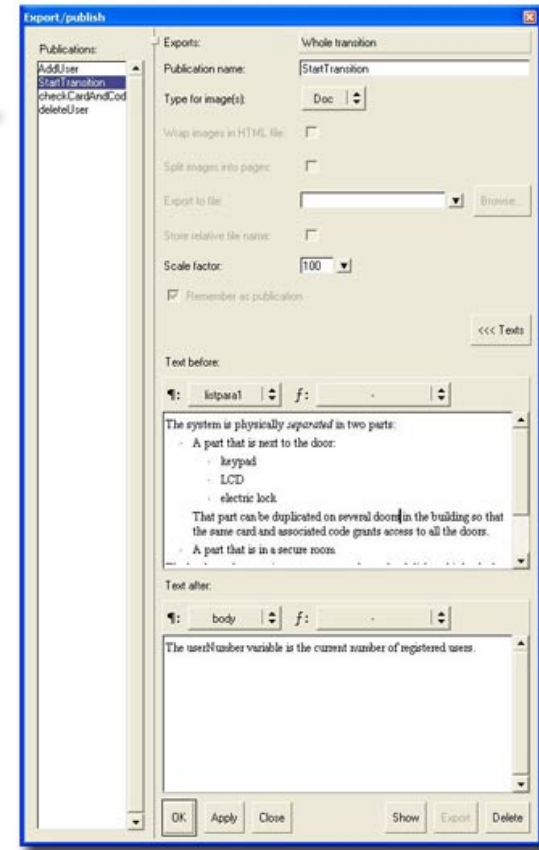
Documentation generation



A document



The generated documentation



A publication

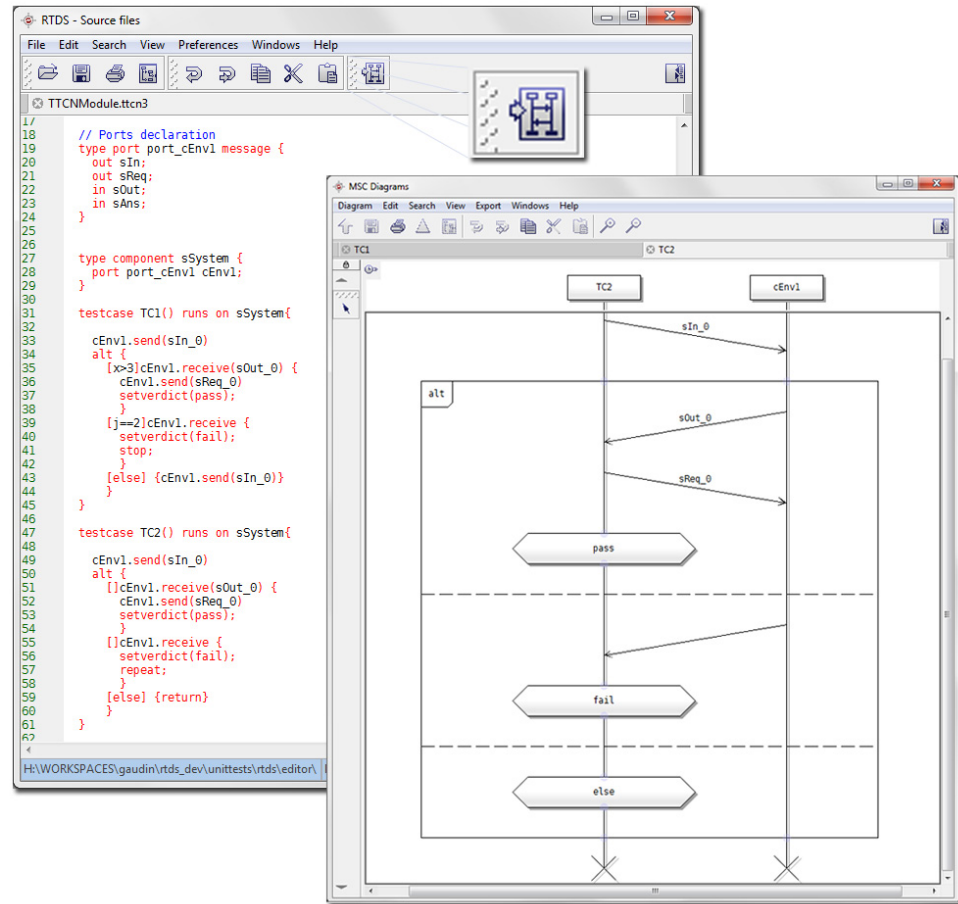
Standard testing language

- Relies on basic services
 - Messages
 - Procedures
 - Timers
 - Parallel execution
- Based on TTCN-3 international standard:
 - Data types definitions or ASN.1,
 - Templates definitions,
 - Test cases,
 - Verdict,
 - Execution control.

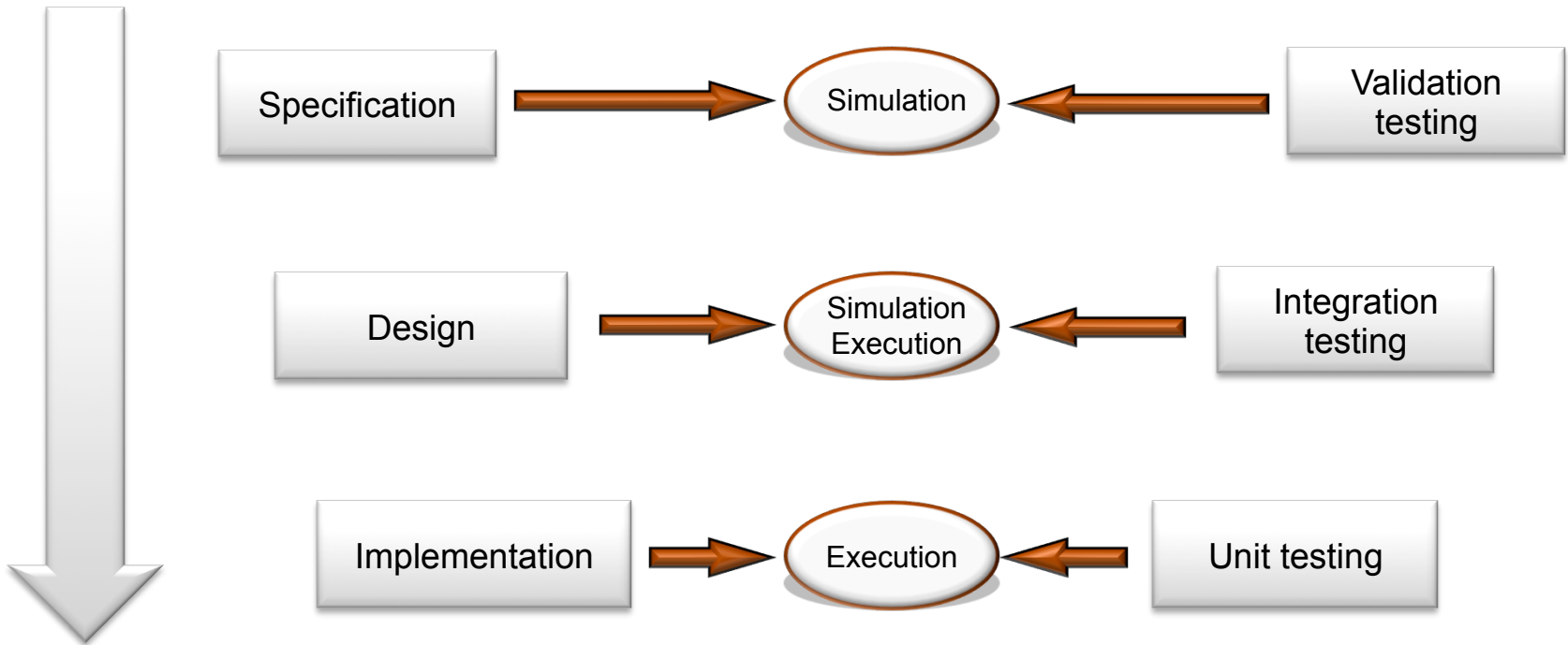


TTCN-3 support

- Textual language
- Simulator with Test manager
- C++ code generator
- TTCN-3 to MSC generation
- MSC to TTCN-3 generation
- TTCN-3 generation from a property on the model (Verimag)
- TTCN-3 generation based on model coverage (to come)



Continuous integration

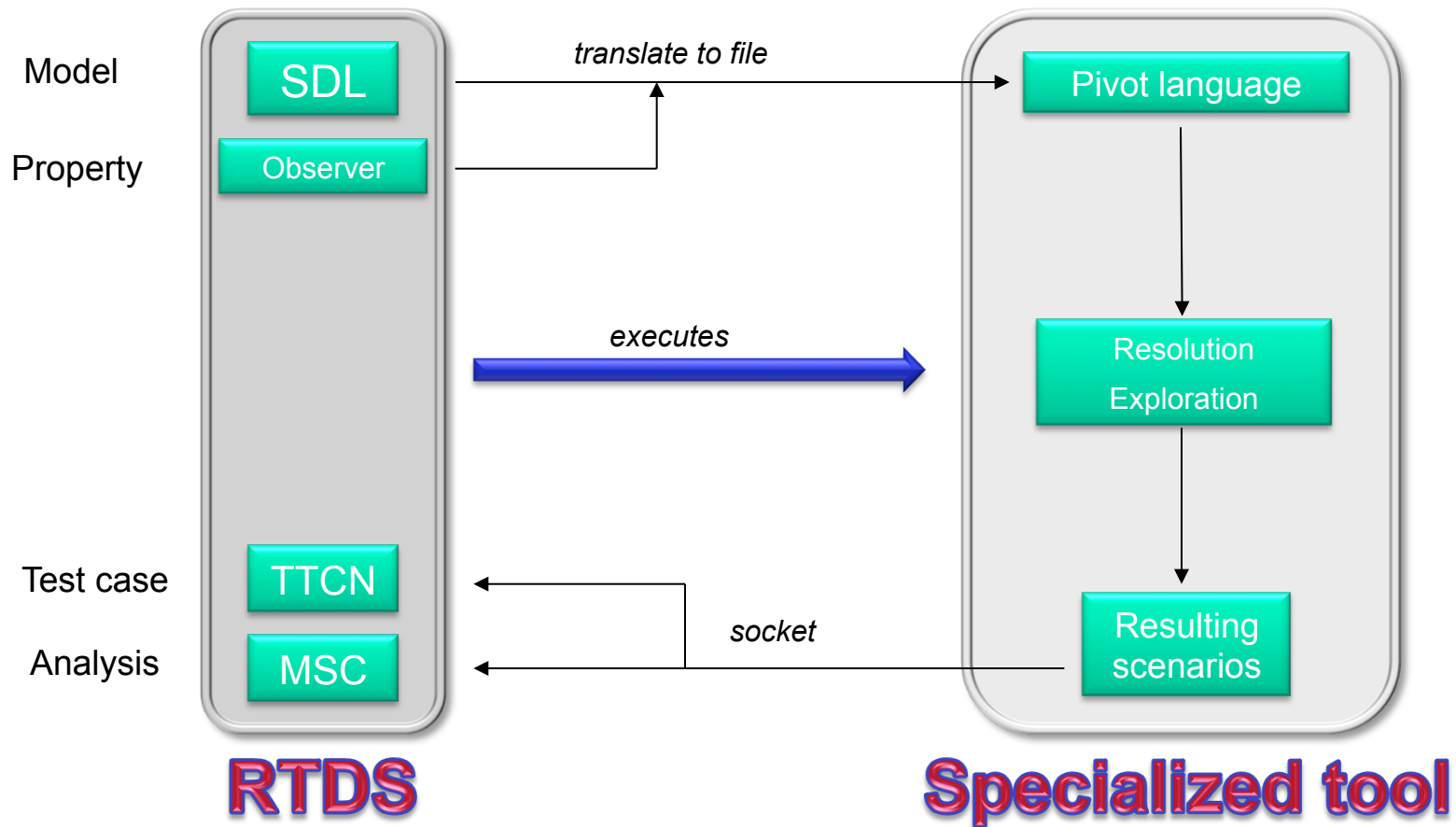


Model checking

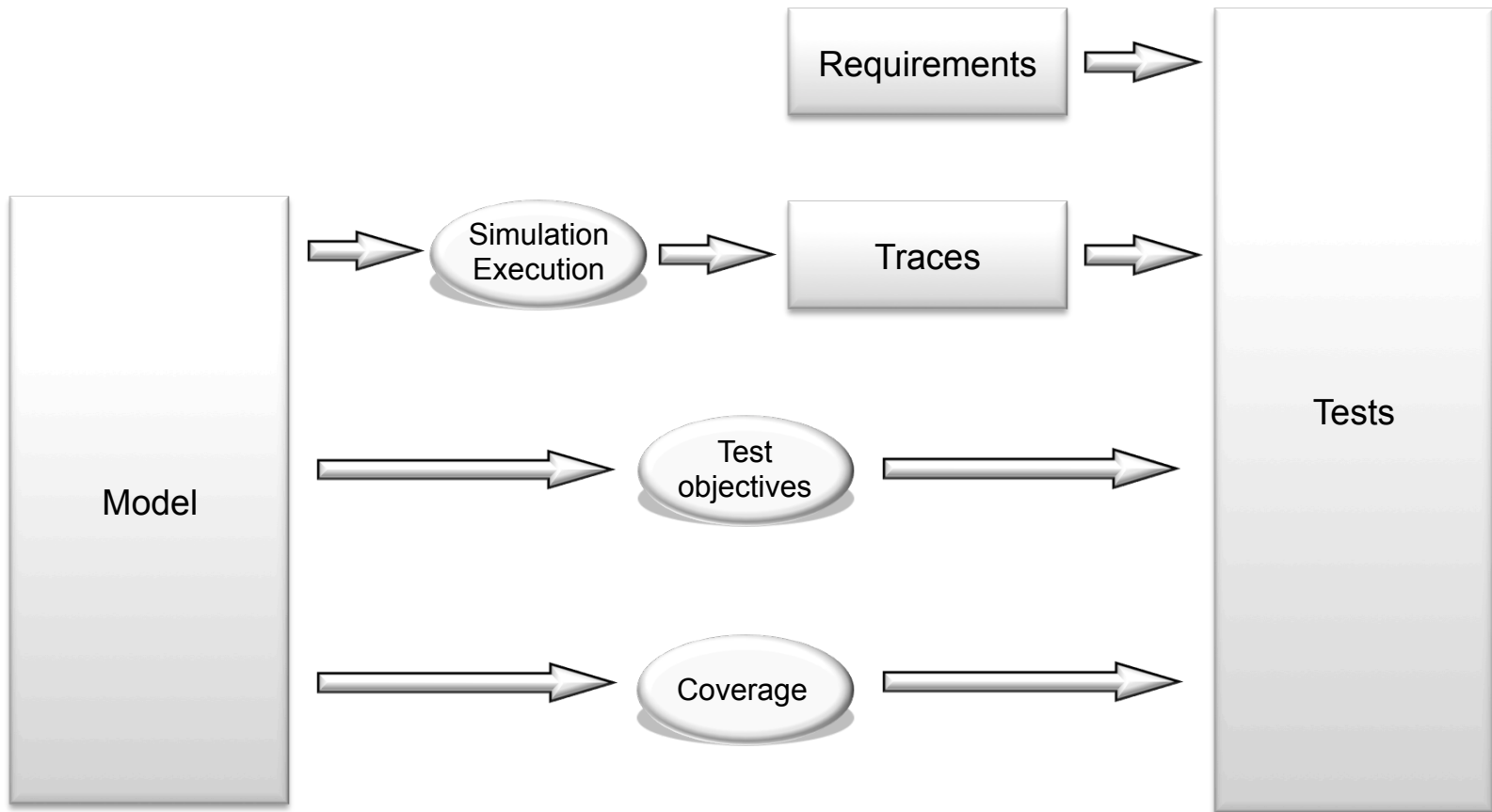
- Partnership with specialized labs:
 - Exhaustive simulation,
 - Symbolic resolution.
- Properties:
 - Model coverage,
 - Static or dynamic property:
 - Property verification,
 - Test objectives.
- RTDS feature:
 - Export,
 - Execute a script,
 - Get the results back.



Implementation



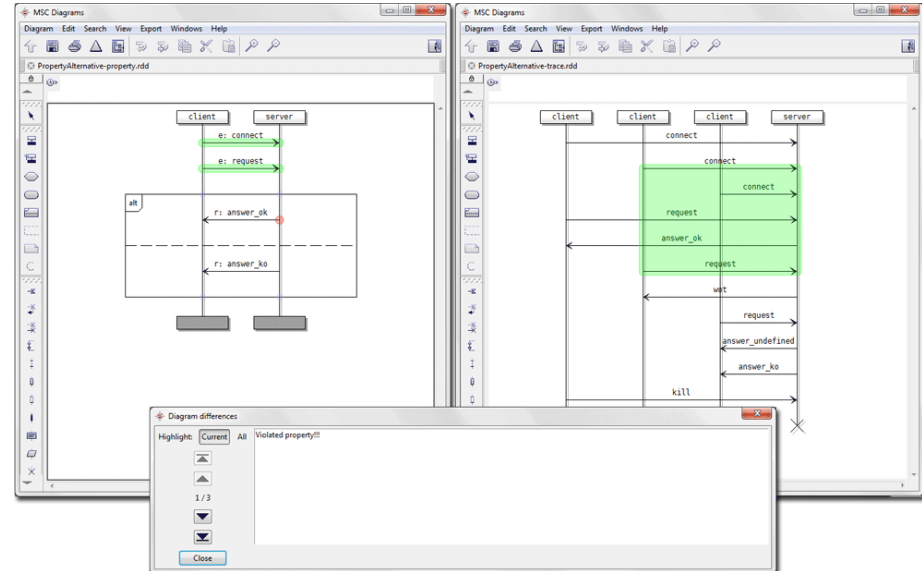
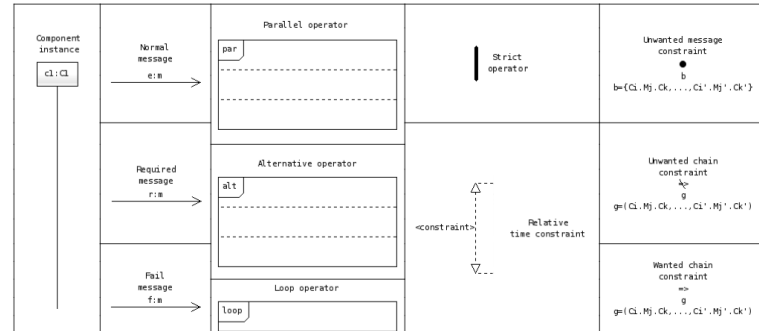
Reference testing



Property Sequence Chart



- PRESTO european project:
 - Functional property verification.
 - Non functional property verification.
- Free tool: PragmaDev Tracer



Conclusion / Future

- SDL FSM Editor:
 - Editor only.
 - More user-friendly than RTDS, but limited.
 - Evolutions planned, but will remain limited.
- RTDS:
 - Much more features: debug, documentation, test, validation, ...
 - V5 will integrate the SDL FSM Editor, and open the same diagrams.