



taste
The Assert Set of Tools for Engineering

**Real Time Developer Studio
(RTDS)
tool for Verification of properties
and
automatic code generation**



19/04/2010

TASTE an assert-project / ESA / Confidential

PragmaDev – Company



Dedicated to the development of a

modelling tool

for the development of

event driven software.



19/04/2010

TASTE an assert-project / ESA / Confidential

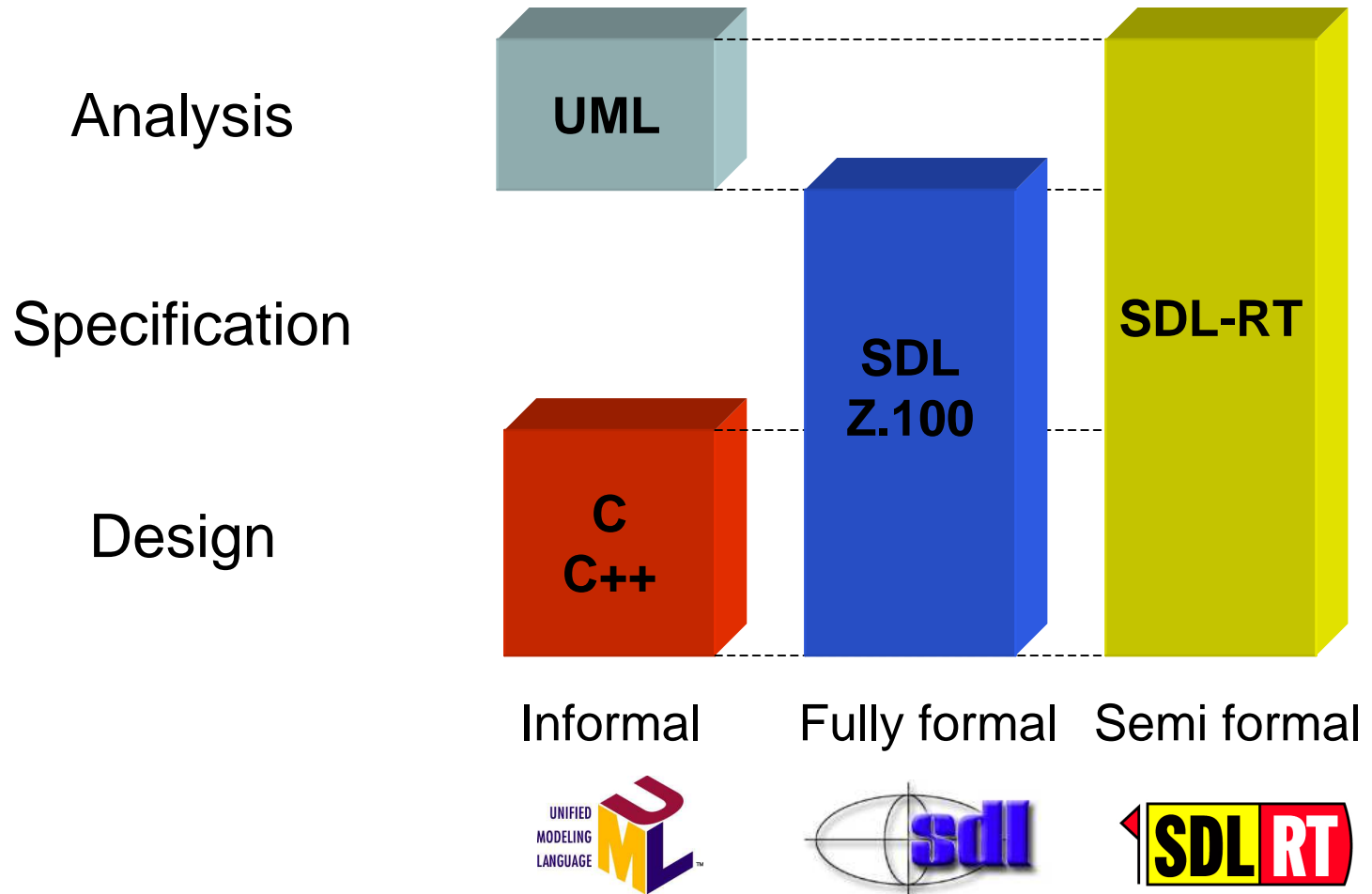
Specification and Description Language

SDL main characteristics:

- ITU-T standard (Z.100),
- Best modeling language for event driven systems (Telecom standards),
- Object oriented (since '92),
- Formal (complete and non ambiguous),
- A UML profile based on SDL has been standardized,
- Increase quality by a ratio of 5,
- Reduces development time by 30%.



PragmaDev – Supported languages



19/04/2010

TASTE an assert-project / ESA / Confidential

PragmaDev – Modelling levels

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

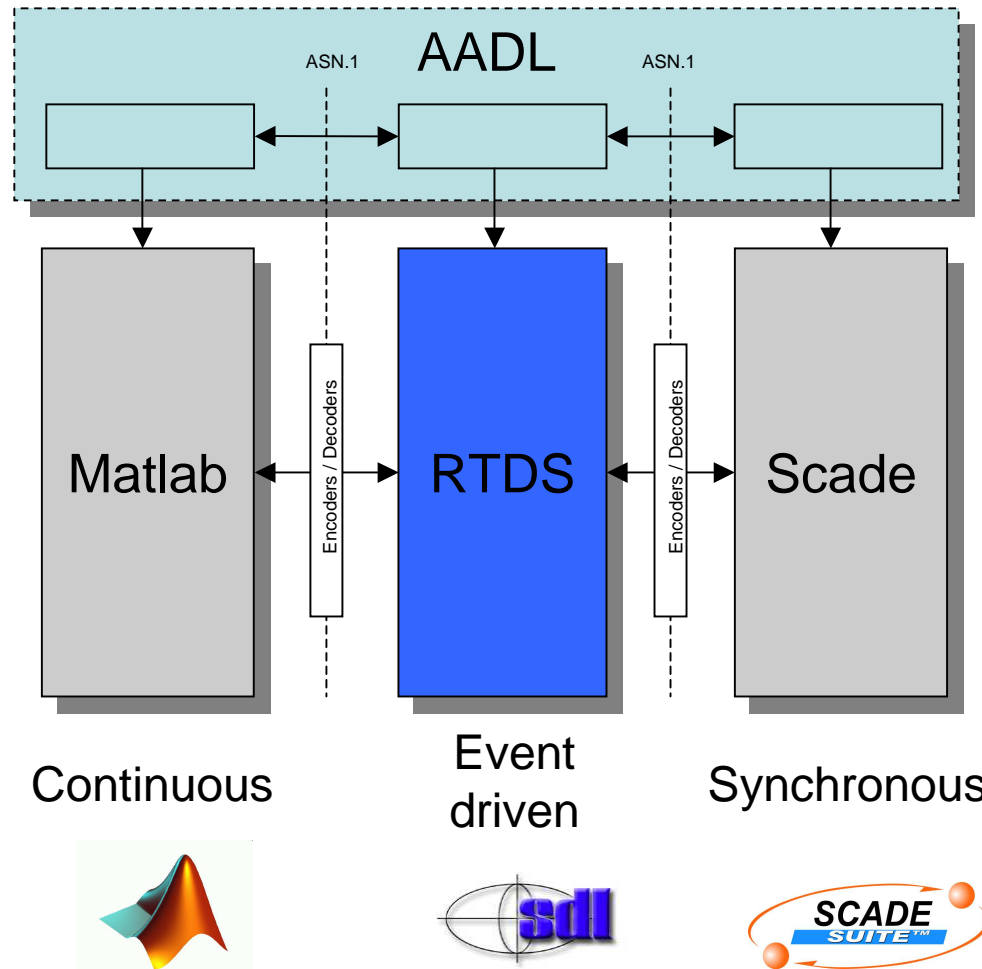
- Edition
- Syntactic et semantics checking
- Simulation
- Verification
- Code generation
- Graphical debugging
- Test



19/04/2010

TASTE an assert-project / ESA / Confidential

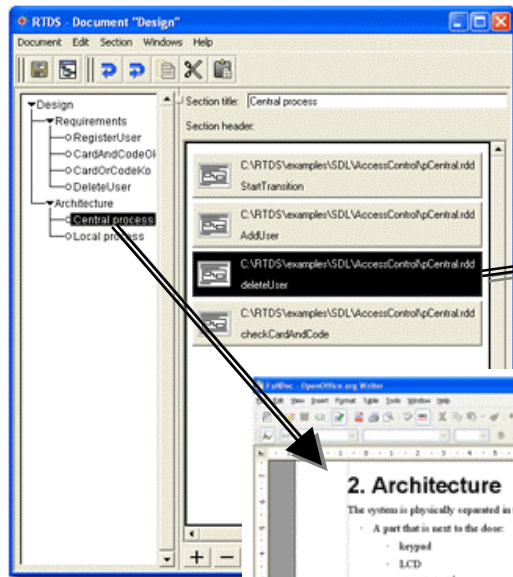
PragmaDev – Positioning in Taste framework



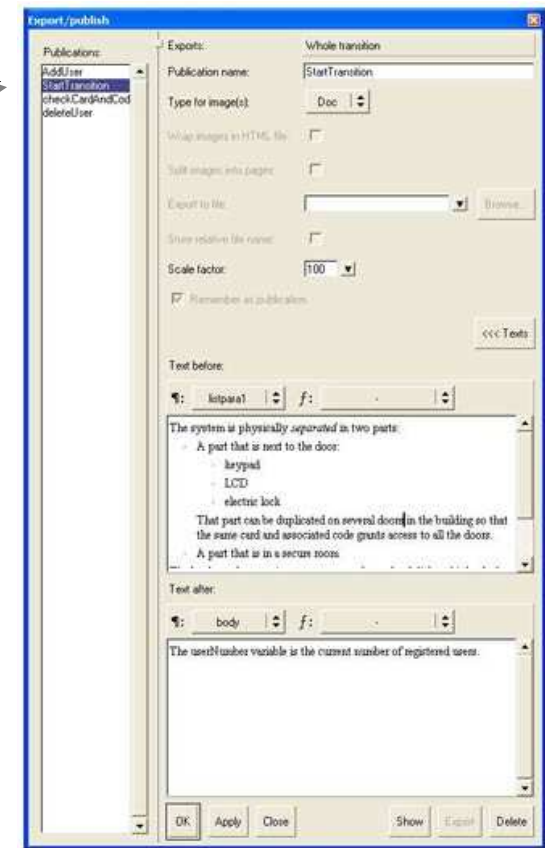
19/04/2010

TASTE an assert-project / ESA / Confidential

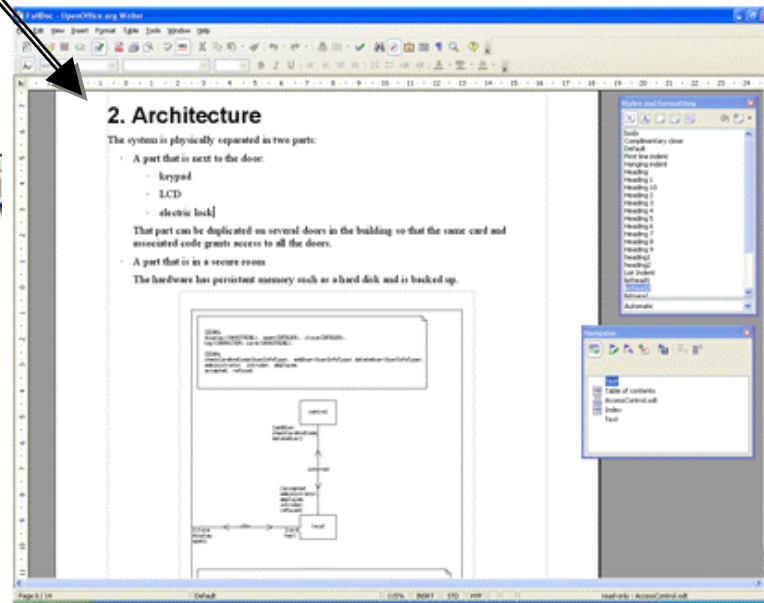
PragmaDev – Documentation generation



A document is made of logical publications



A publication



*The generated documentation
RTF, OpenDocument, HTML, SGML
including table of content and index entries*



19/04/2010

TASTE an assert-project / ESA / Confidential

PragmaDev – Model simulation

The screenshot displays the SDL simulator interface with several panels:

- Process information:** A table listing processes like Layer3, L3Dispatcher, Layer2, RTDS_Env, L3API, and Application with their Pids, Sig, and SDL states.
- Timer info:** A table showing system time (0) and timer events (tAck, tDisplay) with their respective times left.
- Watch window:** A table for monitoring variable values.
- Local variables:** A list of variables such as SELF, senderPid, i, st, s, and z.
- SDL system queue:** A table showing the queue of signals (info, Ack, key) and their receivers (Layer2, Layer3, Application).
- RTDS - Diagram "CoDec" (modified):** A state transition diagram with states like idle, Connecting, and mFAL, and transitions triggered by signals like conReq, disReq, and tAck.
- Debugger console:** A log of system events and messages, including signal reception and state changes.
- Debugger state:** Shows the current state as STOPPED and the active thread as Layer3.

A graphical *debugger* for fully formal models based on the model semantic

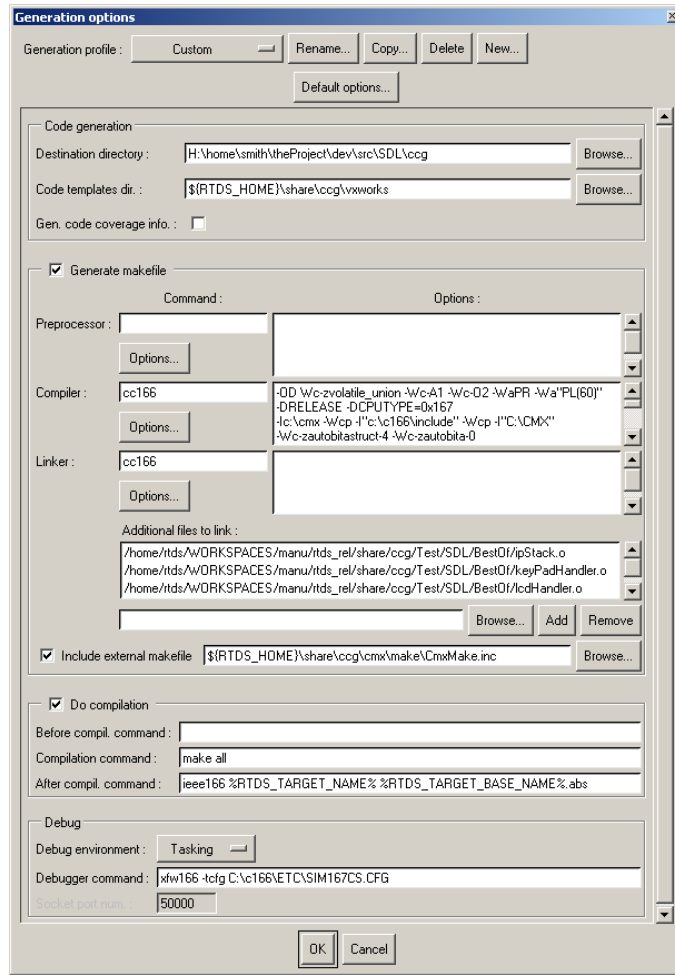
- Set breakpoints and step in the diagrams,
- Externally defined or interactive operator calls,
- Dynamic traces,
- Connecting an external tool is possible through a socket.



19/04/2010

TASTE an assert-project / ESA / Confidential

PragmaDev – Code generation options



- Control of memory allocation,
- C scheduler provided,
- Generated code is legible,
- Generation profile wizard,
- Royalty free,
- Documented for customization.

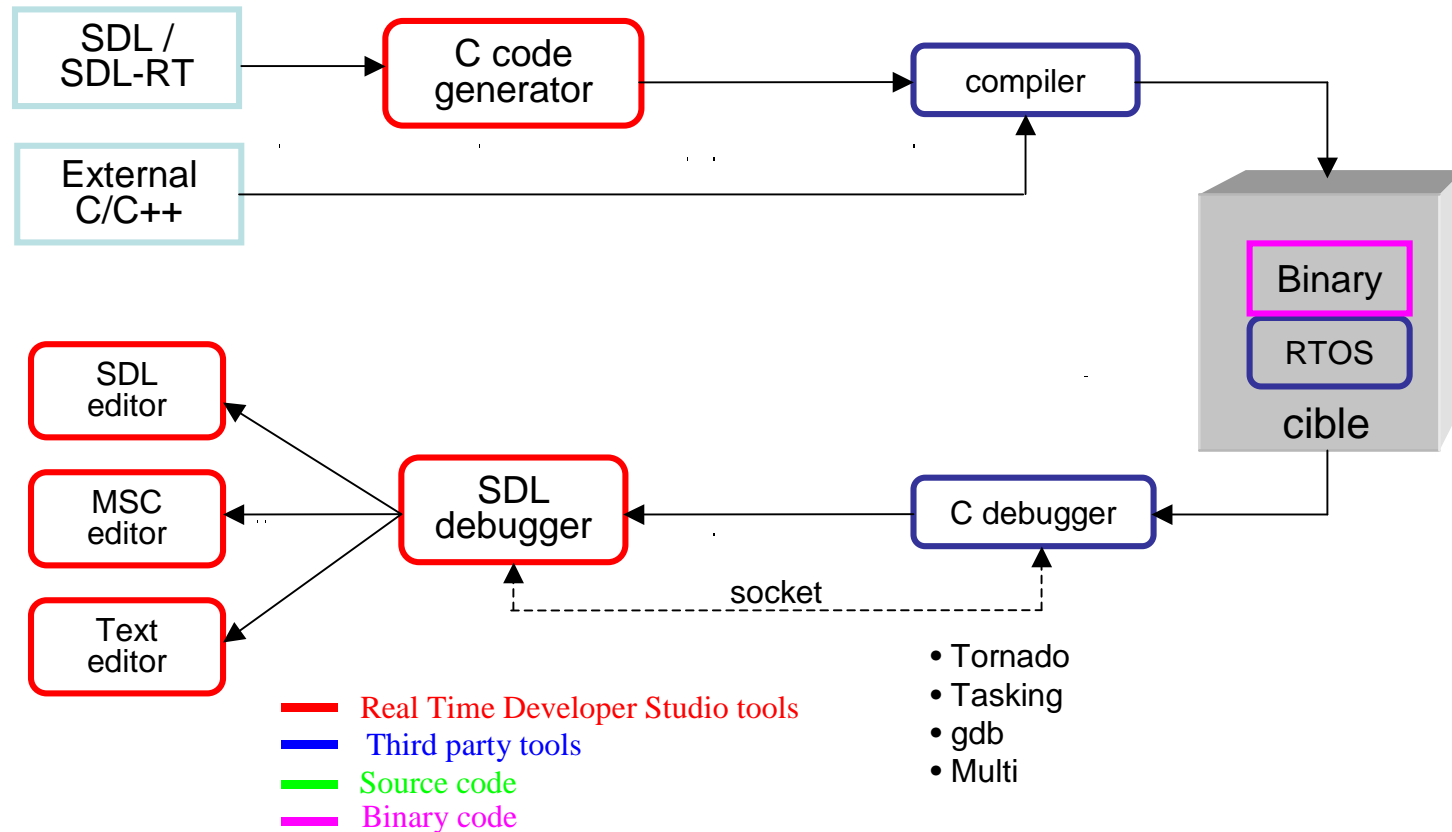


19/04/2010

TASTE an assert-project / ESA / Confidential

PragmaDev – Code generation and debug

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



19/04/2010

TASTE an assert-project / ESA / Confidential

PragmaDev – Model debugger

Depending on the integration model:

- Relies on the target semantic: processor and RTOS,
- Relies on the SDL semantic.



Debug in the model:

- Breakpoints, stepping, in the 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 (empty) with SemId 0x4b3ef08, binary - PPRID, held by pRemote (prio:15) and pLocal (prio:175).
- Watch window:** myGlobalVariable ((\$SharedData *) 0x4b2cc88) with fields: id (2), name (0x495cd68 "supervisor"), level (4), next (struct tSharedData *) 0x0.
- Local variables:** 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: 0x1392 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\ccg\SDL\BestOf\CCG_S0\pSupervisor.c, at address: 0x495ce17,
>step
>Break at line: 46, in file: H:\WORKSPACES\manu\rtdev\Test\ccg\SDL\BestOf\CCG_S0\pSupervisor.c, at address: 0x495ce22
>step
>Break at line: 47, in file: H:\WORKSPACES\manu\rtdev\Test\ccg\SDL\BestOf\CCG_S0\pSupervisor.c, at address: 0x495ce2e
>step
>Break at line: 48, in file: H:\WORKSPACES\manu\rtdev\Test\ccg\SDL\BestOf\CCG_S0\pSupervisor.c, at address: 0x495ce3a
>myGlobalVariable watch added.
>
```
- Debugger state:** STOPPED, Active thread: 0x4b22268->pSupervisor



19/04/2010

TASTE an assert-project / ESA / Confidential

PragmaDev – Message Sequence Chart

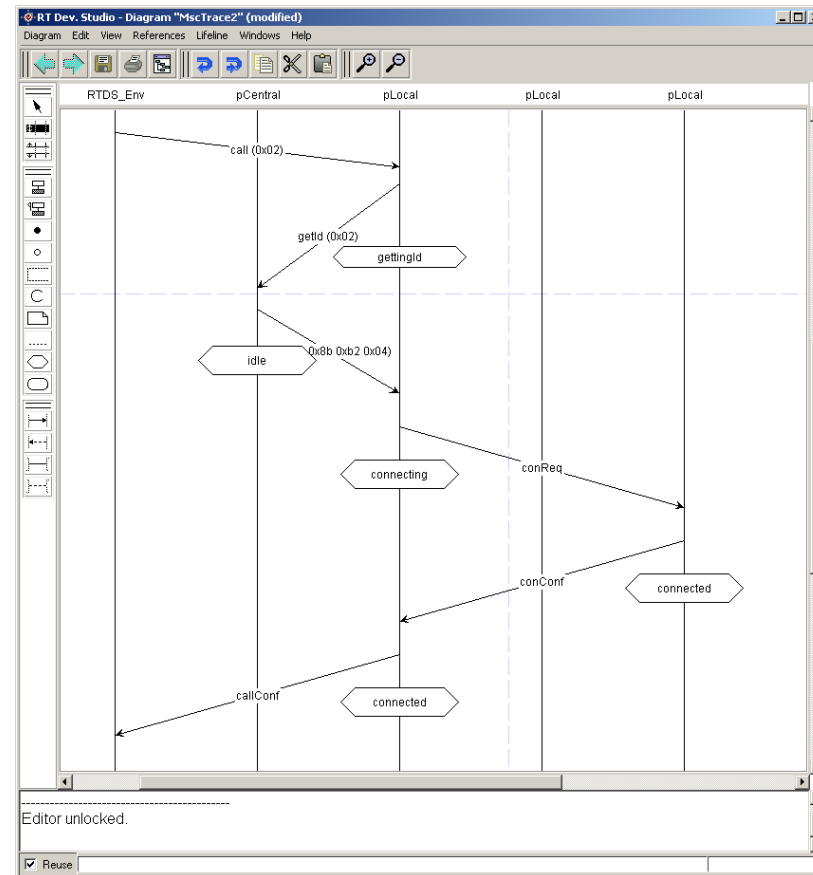
Execution traces:

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

Trace level configuration
Display of system time

MSC Diff allows to check:

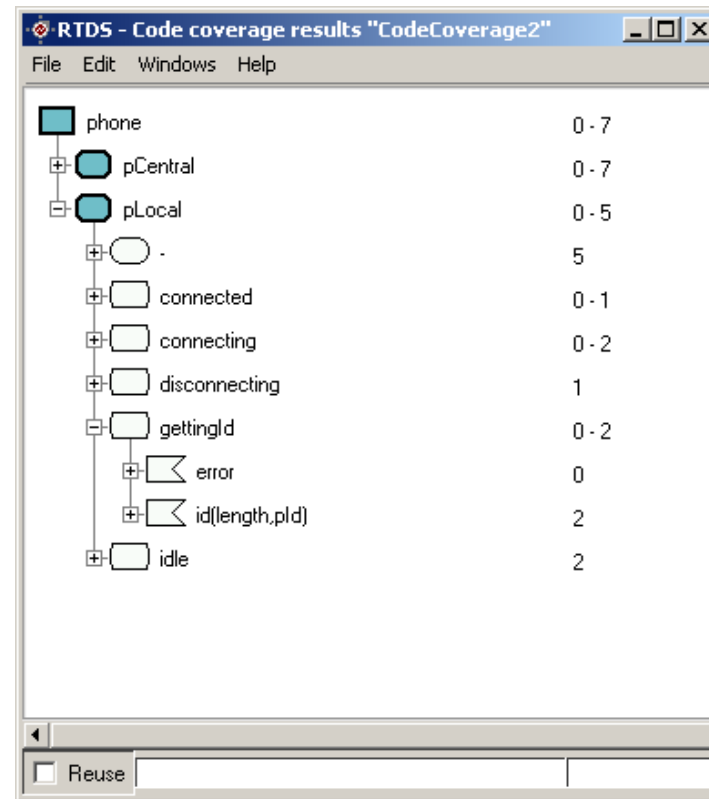
- Conformity,
- Non-regression.



PragmaDev – Model coverage



- Graphical model coverage analysis
- Merge feature



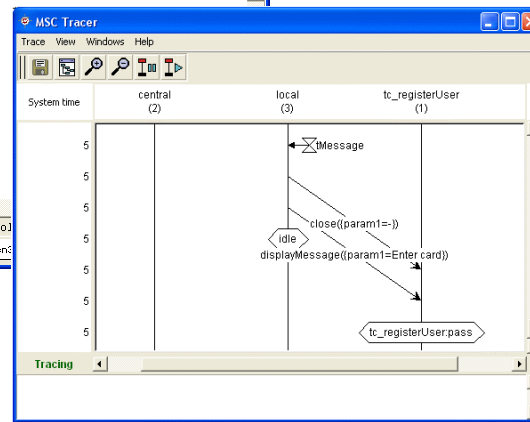
19/04/2010

TASTE an assert-project / ESA / Confidential

PragmaDev – Model based testing



```
RTDS AccessControlTest.ttcn3
File Edit Search Preferences Windows Help
type port cEnv_type message {
  out card;
  out key;
  in displayMessage;
  in close;
  in open;
}
type component AccessControl {
  port cEnv_type cEnv;
}
template displayMessage EnterCardMessage := { param1 := "Enter card";
template displayMessage EnterCodeMessage := { param1 := "Enter code";
template displayMessage EnterNewCodeMessage := { param1 := "Enter new code";
template displayMessage OneStar := { param1 := "1";
template displayMessage TwoStar := { param1 := "11";
template displayMessage ThreeStar := { param1 := "111";
template displayMessage Whatever := { };
template card AdministratorCard := { param1 := "MasterCard";
template card UserCardOne := { param1 := "UserCard1";
template card UserCardTwo := { param1 := "UserCard2";
template key Key0 := { param1 := "0";
template key Key1 := { param1 := "1";
template key Key2 := { param1 := "2";
template key Key3 := { param1 := "3";
template key Key4 := { param1 := "4";
template key Key5 := { param1 := "5";
template key Key6 := { param1 := "6";
template key Key7 := { param1 := "7";
template key Key8 := { param1 := "8";
template key Key9 := { param1 := "9";
template key KeyStar := { param1 := "+";
template key KeyHash := { param1 := "#";
testcase tc_registerUser() runs on AccessControl {
  cEnv.receive(EnterCardMessage);
  cEnv.send(AdministratorCard);
  cEnv.receive(EnterCodeMessage);
  cEnv.send(Key0);
  cEnv.receive(OneStar);
  cEnv.send(Key0);
  cEnv.receive(TwoStar);
}
```



- Based on TTCN-3 international standard:
 - Data types definitions,
 - Templates definitions,
 - Test cases,
 - Execution control.
- Connects automatically to the Simulator:
 - Breakpoints in the model or in the test suite,
 - Verdict displayed in the trace.

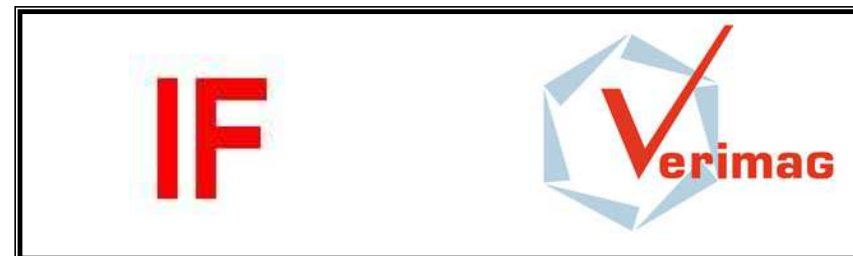


19/04/2010

PragmaDev – Model checking

Exoticus

- Partnership with Verimag on IF technology.
 - Exhaustive simulation,
 - Observers,
 - Test generation.
- RTDS feature
 - Export to IF,
 - Execute a script
 - Generate an MSC feedback.



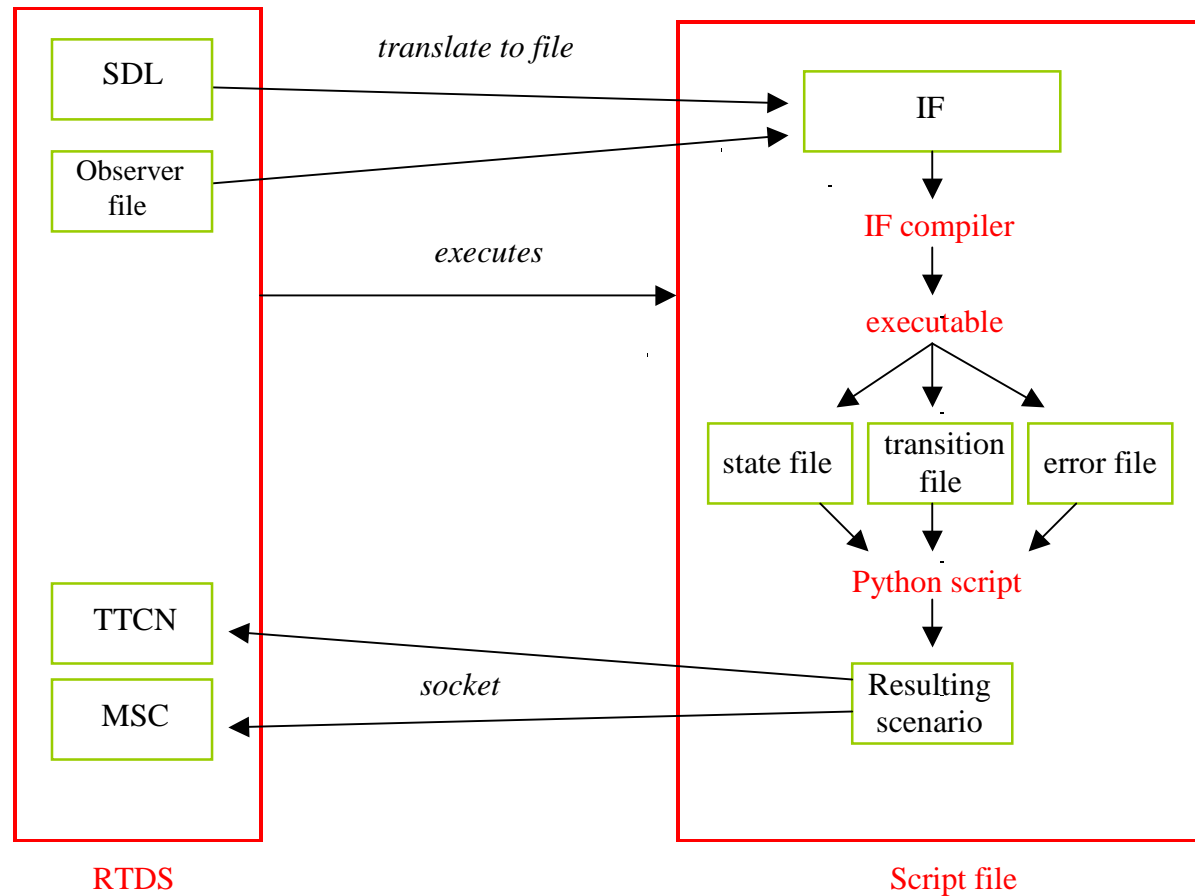
taste
The Assert Set of Tools for Engineering



19/04/2010

TASTE an assert-project / ESA / Confidential

PragmaDev – External tool integration



19/04/2010

TASTE an assert-project / ESA / Confidential

PragmaDev – Conclusion

- Three levels of modelling:
 - Informal,
 - Semi-formal,
 - Formal.
- Tools to:
 - Document,
 - Simulate,
 - Validate,
 - Test.
- Based on international standards.
- Integrated in Taste framework.

taste
The Assert Set of Tools for Engineering



19/04/2010

TASTE an assert-project / ESA / Confidential