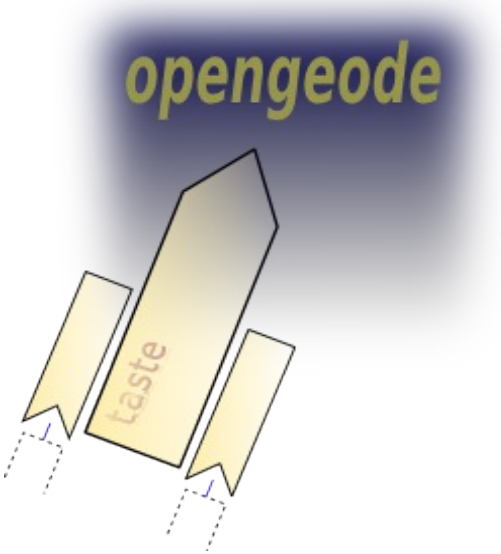
# opengeode

A tiny free, open-source state-machine editor and code generator based on the SDL and ASN.1 languages

Maxime.Perrotin@esa.int
Software Engineering Division - TEC-SWE

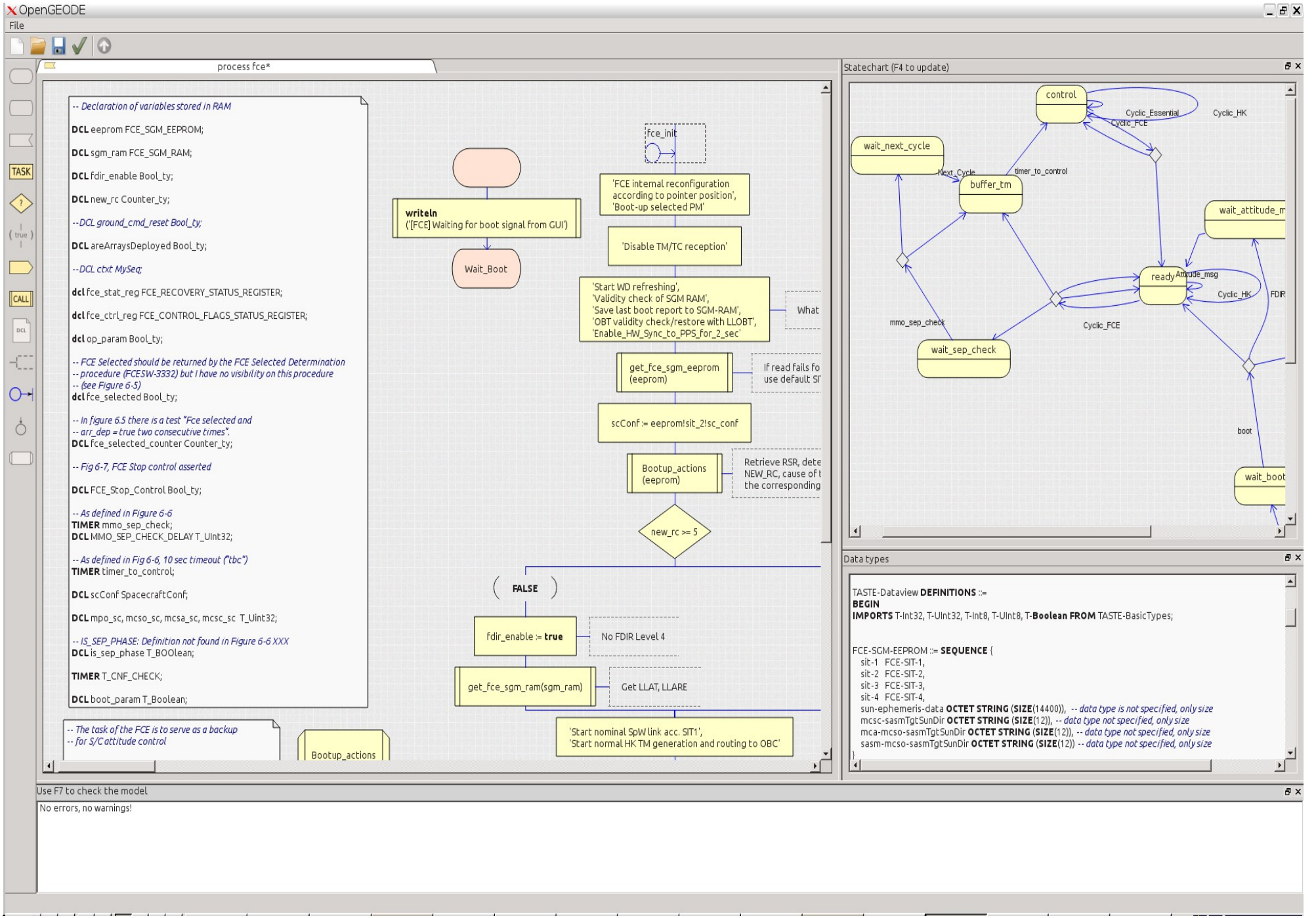
- Opengeode is an SDL editor that was created to support the design and quick prototyping of state machines in the scope of the TASTE project.




- TASTE is a modelling environment that targets embedded, real-time systems ; it is based on formal languages.

  Check it ! http://taste.tuxfamily.org

taste
The Assert Set of Tools for Engineering


OpenGEODE
File
process fce*
-- Declaration of variables stored in RAM
DCL eeprom FCE_SGM_EEPROM;
DCL sgm_ram FCE_SGM_RAM;
DCL fdir_enable Bool_ty;
DCL new_rc Counter_ty;
--DCL ground_cmd_reset Bool_ty;
DCL areArraysDeployed Bool_ty;
--DCL ctxt MySeq;
dcl fce_stat_reg FCE_RECOVERY_STATUS_REGISTER;
dcl fce_ctrl_reg FCE_CONTROL_FLAGS_STATUS_REGISTER;
dcl op_param Bool_ty;
-- FCE Selected should be returned by the FCE Selected Determination
-- procedure (FCESW-3332) but I have no visibility on this procedure
-- (see Figure 6-5)
dcl fce_selected Bool_ty;
-- In figure 6.5 there is a test "Fce selected and
-- arr_dep = true two consecutive times".
DCL fce_selected_counter Counter_ty;
-- Fig 6-7, FCE Stop control asserted
DCL FCE_Stop_Control Bool_ty;
-- As defined in Figure 6-6
TIMER mmo_sep_check;
DCL MMO_SEP_CHECK_DELAY T_UInt32;
-- As defined in Fig 6-6, 10 sec timeout ("tbc")
TIMER timer_to_control;
DCL scConf SpacecraftConf;
DCL mpo_sc, mcso_sc, mcsa_sc, mcsc_sc T_UInt32;
-- IS_SEP_PHASE: Definition not found in Figure 6-6 XXX
DCL is_sep_phase T_BOOlean;
TIMER T_CNF_CHECK;
DCL boot_param T_Boolean;
-- The task of the FCE is to serve as a backup
-- for S/C attitude control
Bootup_actions
fce_init
writeln
('[FCE] Waiting for boot signal from GUI')
Wait_Boot
'FCE internal reconfiguration
according to pointer position',
'Boot-up selected PM'
'Disable TM/TC reception'
'Start WD refreshing',
'Validity check of SGM RAM',
'Save last boot report to SGM-RAM',
'OBT validity check/restore with LLOBT',
'Enable_HW_Sync_to_PPS_for_2_sec'
What
get_fce_sgm_eeprom
(eeprom)
If read fails fo
use default SI
scConf := eeprom!sit_2!sc_conf
Bootup_actions
(eeprom)
Retrieve RSR, dete
NEW_RC, cause of t
the corresponding
new_rc >= 5
FALSE
fdir_enable := true
No FDIR Level 4
get_fce_sgm_ram(sgm_ram)
Get LLAT, LLARE
'Start nominal SpW link acc. SIT1',
'Start normal HK TM generation and routing to OBC'
Statechart (F4 to update)
control
Cyclic_Essential
Cyclic_HK
Cyclic_FCE
wait_next_cycle
Next_Cycle
timer_to_control
buffer_tm
wait_attitude_m
ready
Attitude_msg
Cyclic_HK
FDIR
mmo_sep_check
Cyclic_FCE
wait_sep_check
boot
wait_boot
Data types
TASTE-Dataview DEFINITIONS ::=
BEGIN
IMPORTS T-Int32, T-UInt32, T-Int8, T-UInt8, T-Boolean FROM TASTE-BasicTypes;
FCE-SGM-EEPROM ::= SEQUENCE {
sit-1 FCE-SIT-1,
sit-2 FCE-SIT-2,
sit-3 FCE-SIT-3,
sit-4 FCE-SIT-4,
sun-ephemeris-data OCTET STRING (SIZE(14400)), -- data type is not specified, only size
mcsc-sasmTgtSunDir OCTET STRING (SIZE(12)), -- data type not specified, only size
mca-mcso-sasmTgtSunDir OCTET STRING (SIZE(12)), -- data type not specified, only size
sasm-mcso-sasmTgtSunDir OCTET STRING (SIZE(12)) -- data type not specified, only size
}
Use F7 to check the model
No errors, no warnings!

# Download and installation

- Opengeode can run on many platforms (tested on Linux, Windows, FreeBSD and Mac)
- It is pre-installed in the TASTE virtual machine
- Manual download and installation instructions on www.opengeode.net or https://github.com/maxime-esa/opengeode or https://pypi.python.org/pypi/opengeode

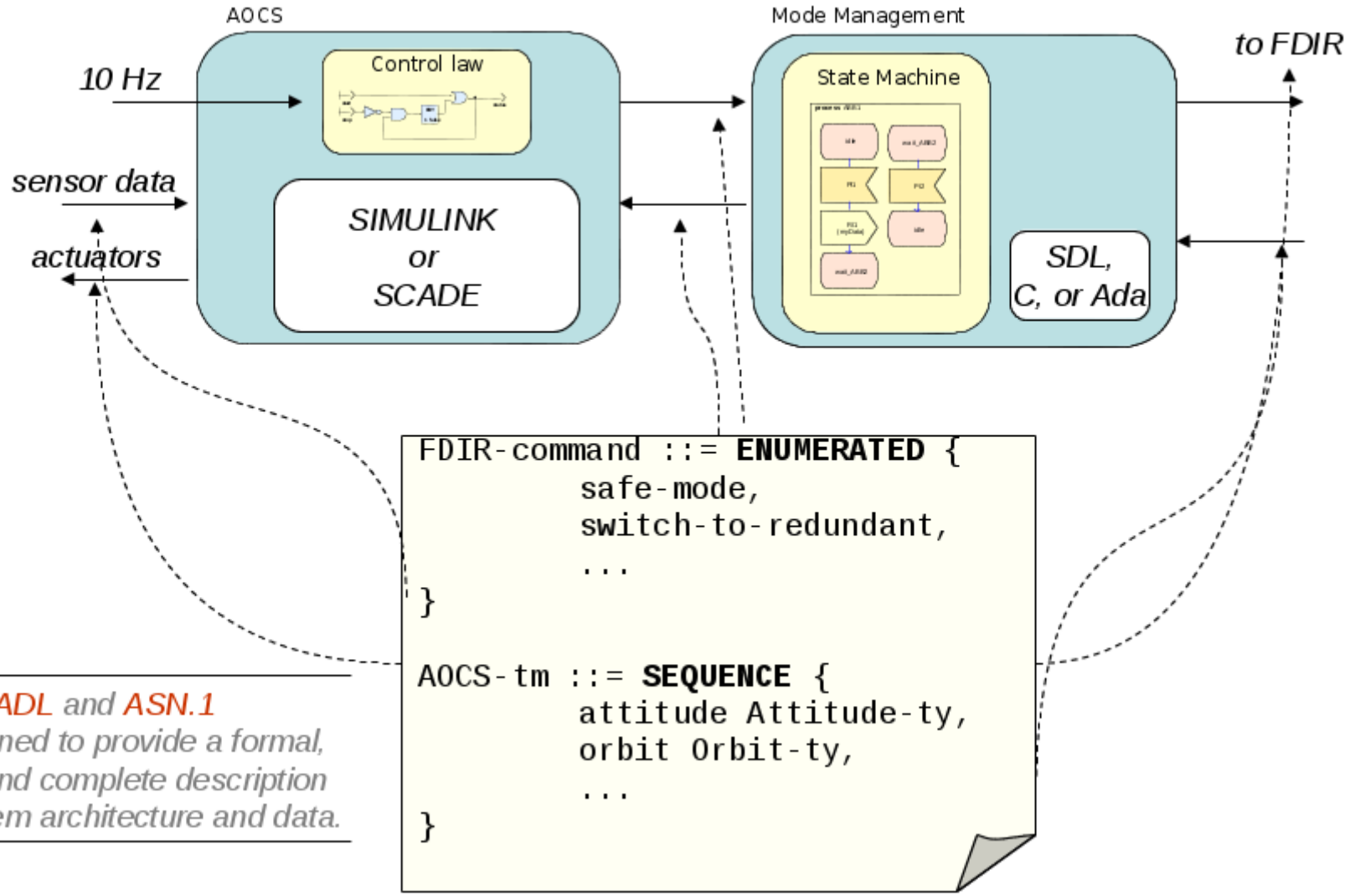
# What is SDL good for


- Specify or design the behaviour of communicating software, such as
  - Embedded software,
  - Communication protocols
  - State machines
- Reformulate and understand existing software specifications, design, or code

- Simulate, verify and generate low-level code (C, Ada) from a formal language that is simple, complete, unambiguous, easy to learn and clear to read.

- Check http://www.sdl-forum.org

# Architecture and data → use TASTE




*AADL* and *ASN.1*
are combined to provide a formal, precise, and complete description of the system architecture and data.

# ASN.1

- International standard (ISO and ITU-T)
- Simple text notation for precise and complete **data type description**
- Real added values compared to other notations
  - Value notation : the syntax allows to specify types but also the value of an instance of a type – however complex it is ;
  - the physical encoding rules (compact binary encoding, endianness-neutral, but also XML encoding, legacy encoding specifications).
  - Separate the encoding rules from the types specification
  - Not proprietary – many implementations, massive use in existing applications (air trafic control, cryptography, bank transfers...)
  - ASN.1 is fully integrated in the SDL language and can be used natively

- Opengeode uses ESA ASN.1 compiler :

  https://github.com/ttsiodras/asn1scc
  - Generates C and SPARK/Ada code with identical memory mappings

# ASN.1 – basic types

**INTEGER**

→ My-int ::= INTEGER (0..7)

**Value notation : value My-int ::= 5**

**REAL**

→ My-real ::= REAL (10.0 .. 42.0)

**BOOLEAN**

**ENUMERATED**

→ My-enum ::= ENUMERATED { hello, world }

**Value notation : value My-enum ::= hello**

**OCTET STRING**

→ My-string ::= OCTET STRING (SIZE (0..255))

**Value notation : value My-string::= 'DEADBEEF'H**

**BIT STRING**

→ My-bitstring ::= BIT STRING (SIZE (10..12))

**Value notation : value My-bitstring ::= '00111000110'B**

# ASN.1 – complex types

- SEQUENCE
  - → My-seq ::= **SEQUENCE** {
    x My-int,
    y My-enum OPTIONAL
    }

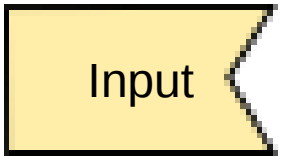
    **Value notation : value My-seq::= { x 5 }**
- CHOICE
  - → My-choice ::= CHOICE {
    choiceA My-real,
    choiceB My-bitstring
    }

    **Value notation : value My-choice ::= choiceA : 42.0**
- SEQUENCE OF
  - → My-seq ::= SEQUENCE (SIZE (0..5)) OF BOOLEAN

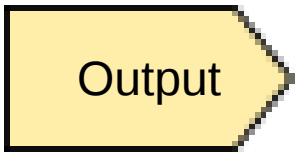
    **Value notation : value My-seq:= { 1, 2 ,3 }**
- SET / SET OF

# Major SDL elements for behavioural design


A state




Input (triggers a transition)




Output (sends a message)

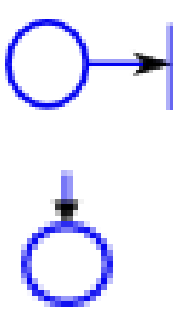

Decision


Action


Procedure call

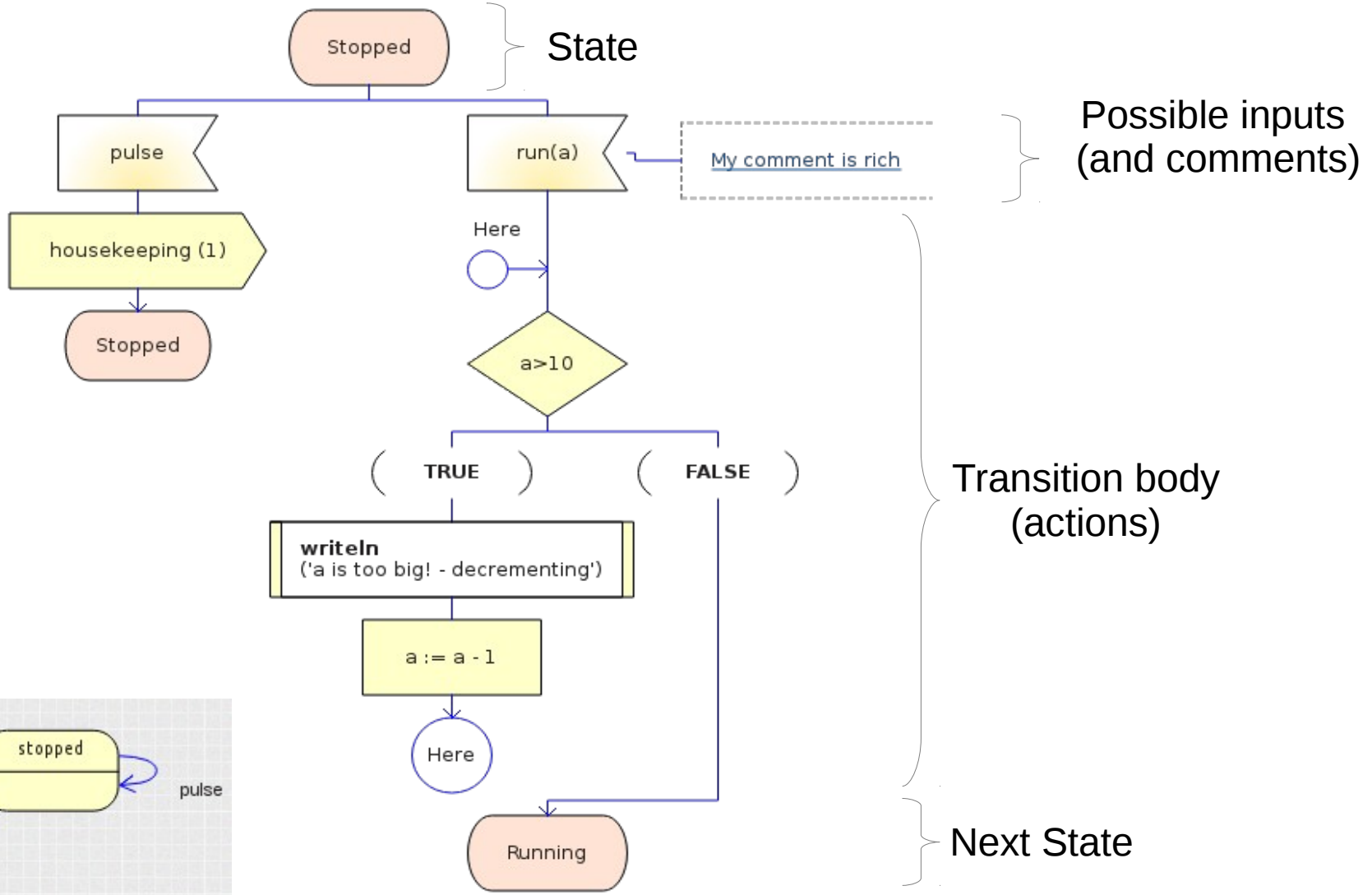
Procedure definition

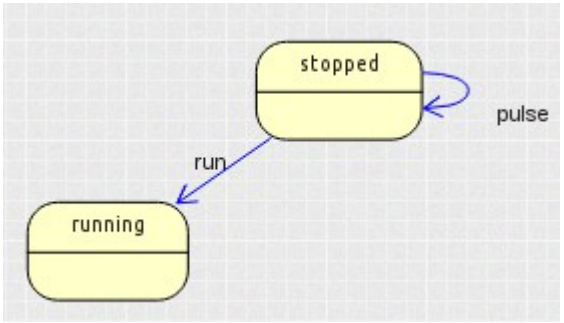

Variables declaration



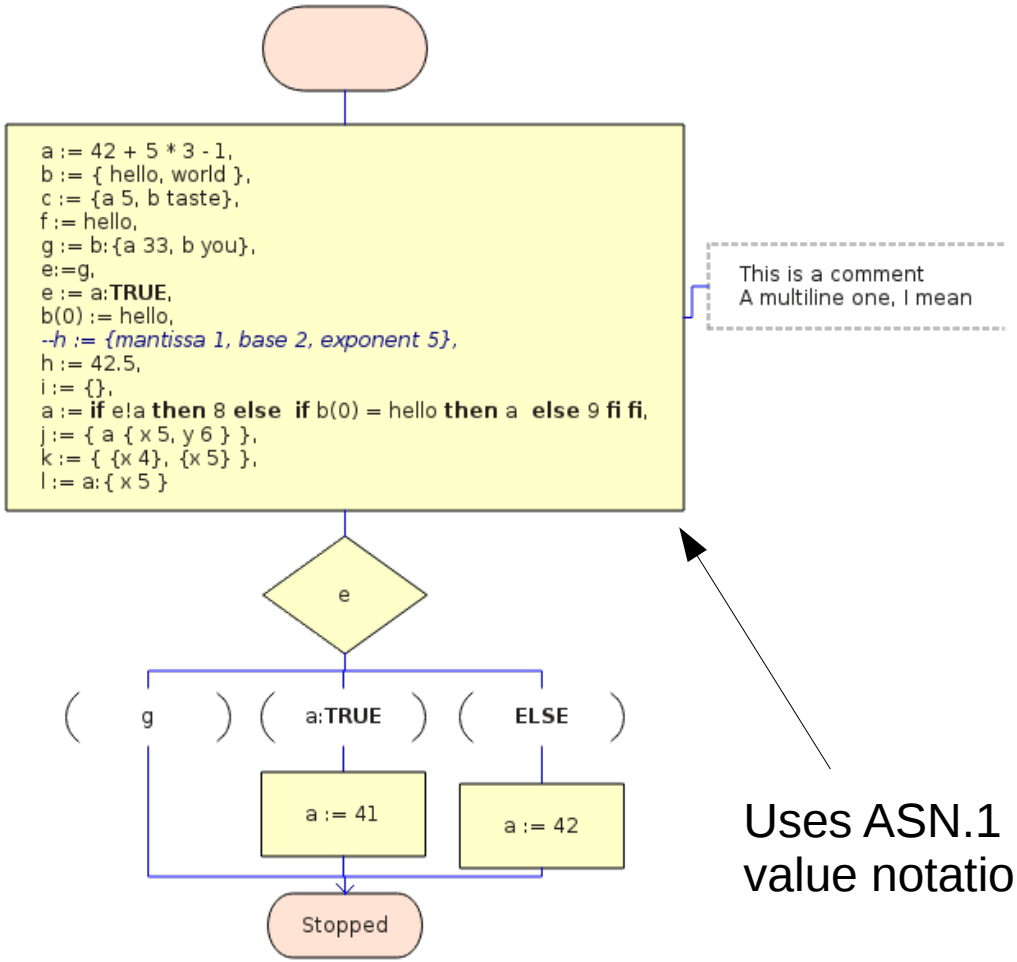
Label/Join

# Typical transition diagram

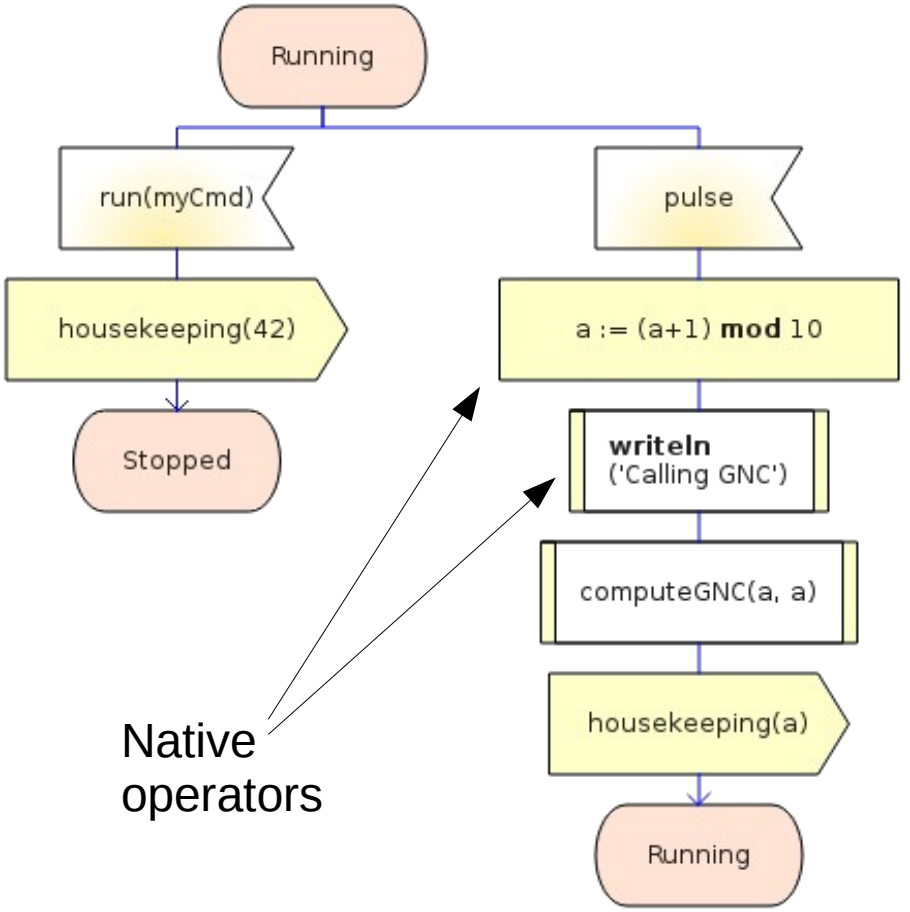


# Data manipulation (overview)

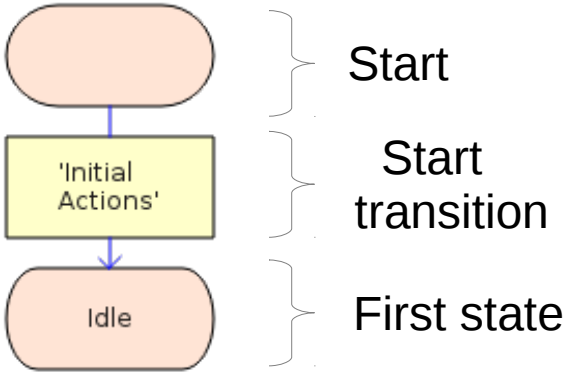



Uses ASN.1 value notation



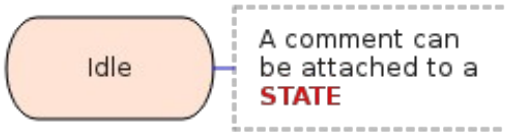

Native operators

# Start : initialization transition



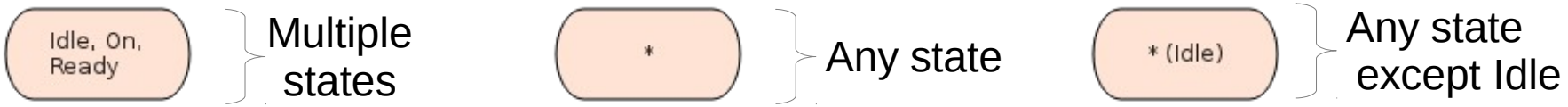

- A state machine has exactly one start transition
- The start transition is executed at process creation (do not call required interfaces there)
- The start transition
  - Sets the initial state
  - May execute initial actions (initialization of variables)

# State / Nextstate

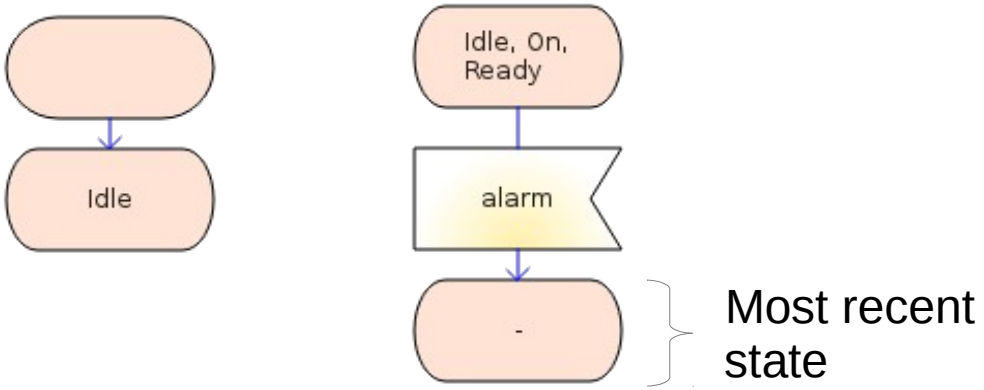



- Each state has a name
- In a given state, the process is expecting to receive messages
- A state can be **composite**

*Shortcuts*

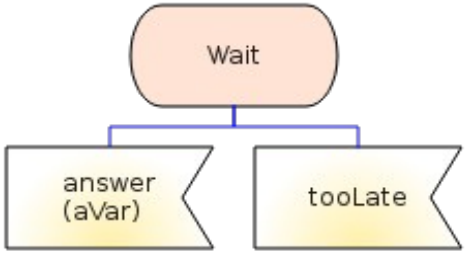



*Shortcut*



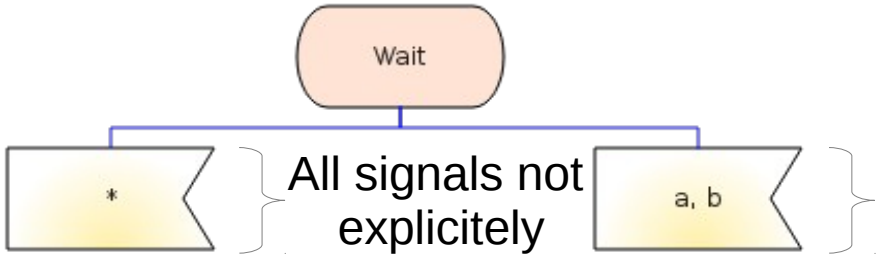

- Arrival state
- Unique
- Is the initial state of other transitions

# Input




- Fires a transition : the transition is executed when the process consumes the signal
- In a given state, the process can expect several signals
- May have parameters (use variables to store their values)

*Shortcuts*

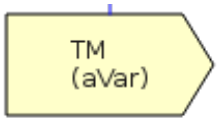



All signals not explicitely mentioned

a or b

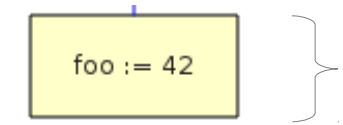
- Inputs at level N have priority over inputs at level N-1 (composite states)
- *As a consequence, be careful with « asterisk » inputs : if the state is composite, all inner inputs are ignored*.

# Output

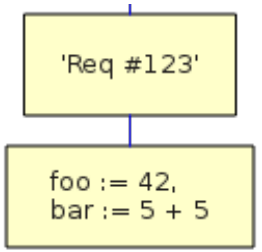



- Transmission of a signal
  *in TASTE terms : invocation of a sporadic required interface*
- May have a parameter

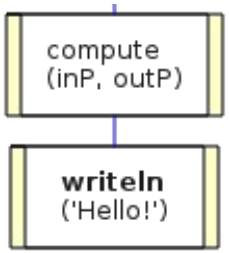
# Task, Procedure call




formal



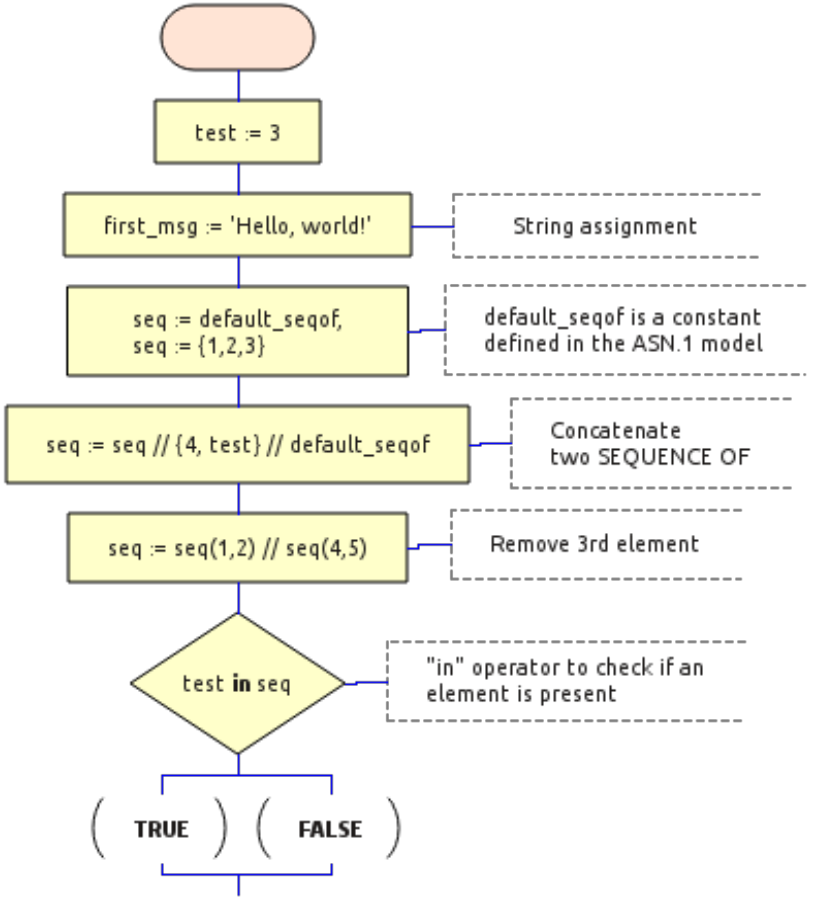

informal

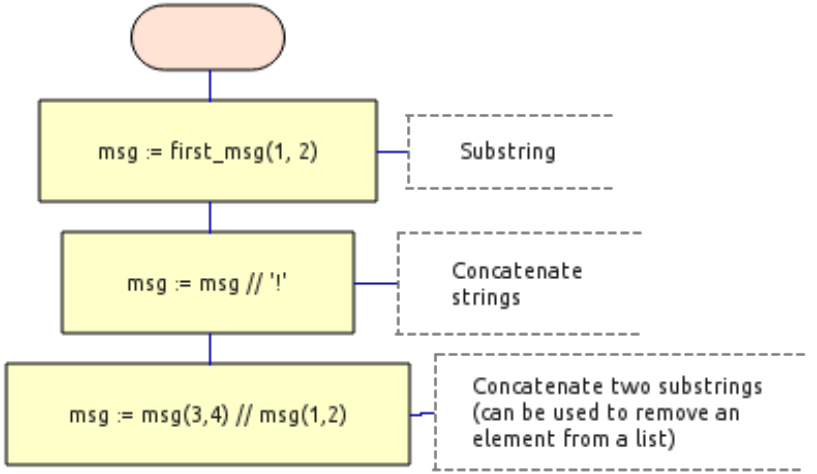



- Elementary action of process transition
- Informal task
- Task setting a variable to a given value
- For loops

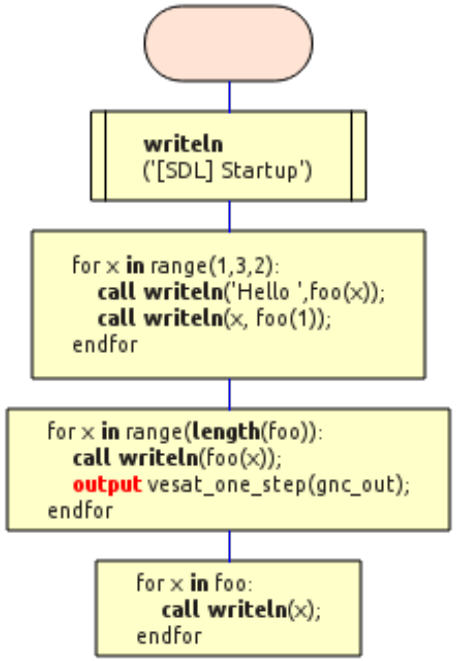
- Call an external procedure
  *In TASTE terms, call a synchronous required interface (protected or unprotected)*
- Can have input and output parameters
- Writeln : built-in print function

# Advanced data manipulation (1)



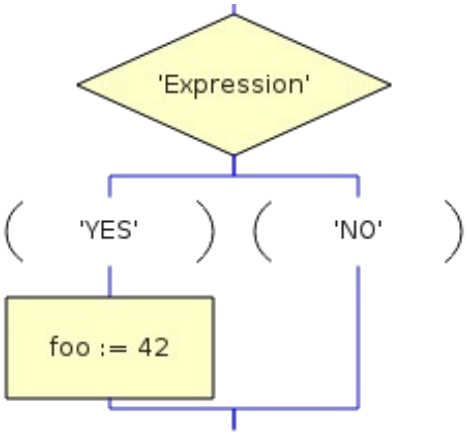
# Advanced data manipulation (2)

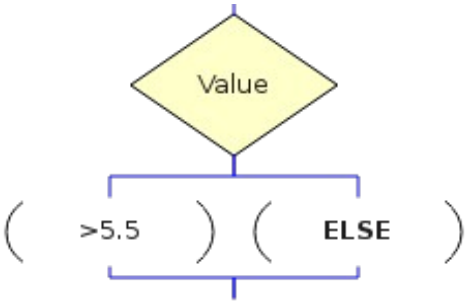



- FOR loop
- Range([start], stop, [step])
  or iterator (SEQUENCE OF)

- Transition can use any SDL construct

# Decision



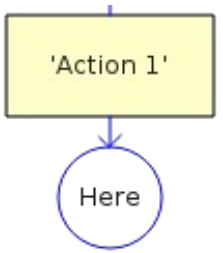



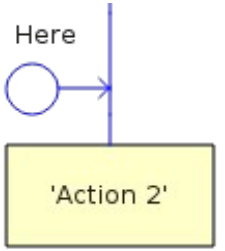

- Control structure
  To represent conditional action sequences
- A decision can have more than two answers

  - Multiple answers must be mutually exclusive

  -The last answer can be ELSE
- Useful to build loops

# Labels and branches

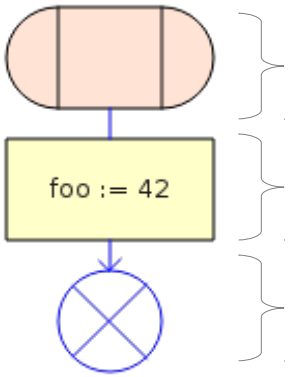



- Allow rerouting
- Loop description
- « Don't repeat yourself » (DRY)

  But do not use to describe complex algorithms..

# Procedures




Start

Transition

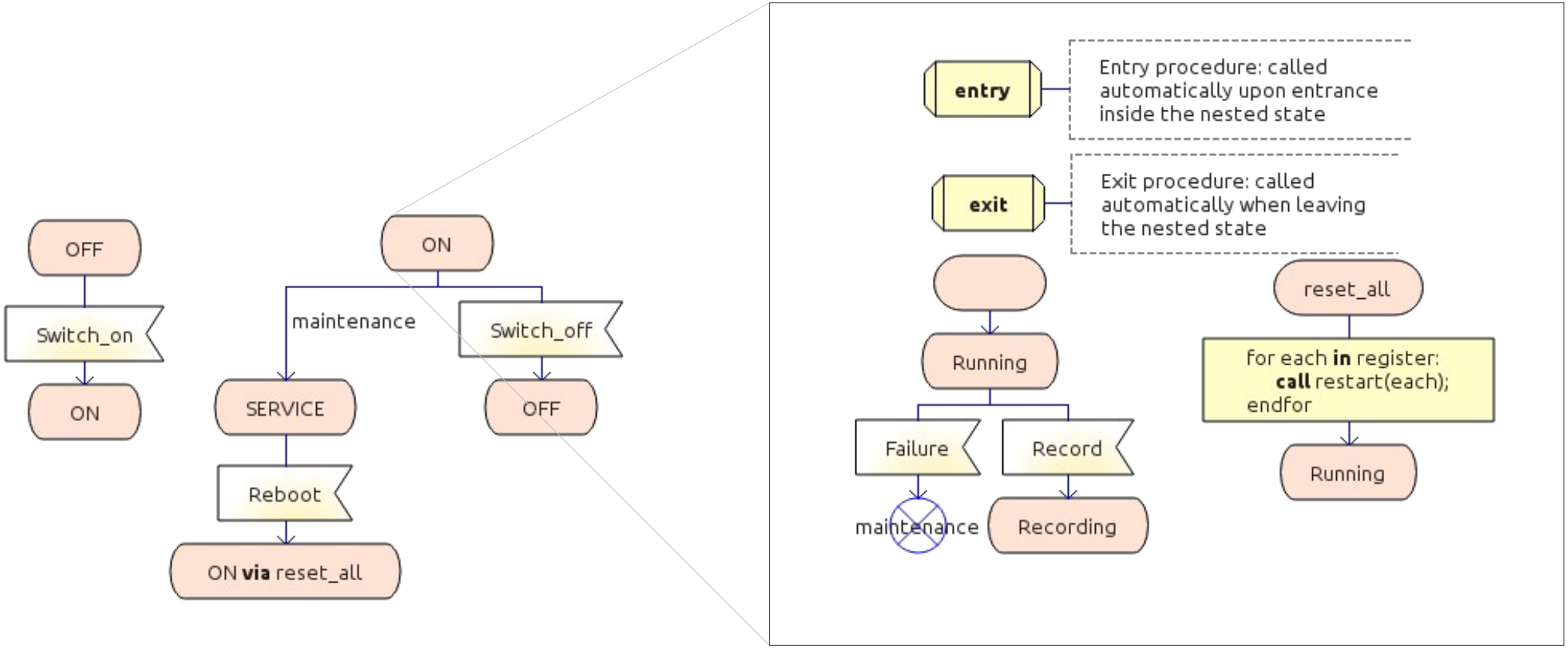
Return

- Sequential sub-functions
- Can have parameters (in and in/out)
- Have visibility on the parent variables
- Same constructs as a process
- Local variables
- But no internal states

```
-- A Local variable
DCL foo MyInteger;

-- Procedure interface
fpar
   in toto MyInteger,
   in/out tutu MyInteger;
```

# Composite states




- Hierarchical state machines
- Entry and exit procedures
- Multiple entry and exit points

# Quality criteria for state machines

- State oriented
  - Use variables for storing data, not object states
- Complexity
  - Number of states
  - Number of transitions per state
  - Avoid decisions in waterfall wat
  - Minimum of data
- Graphical justification comments
- Use hyperlinks for better traceability

# Summary

- SDL includes a complete data model
  - Declare and use variables within transition symbols
- Design is complete
  - Designers without expertise in programming languages can build complete executable models
  - TASTE allows communication with external code
- Best approach : model behaviour with SDL, algorithms with Simulink, and drivers with Ada or C
- Opengeode can generate Ada code for a complete state machine
  - For full SDL support, C code and model simulation use Pragmadev RTDS tool :www.pragmadev.com