

Petit bogue, Grand boum!

jean-jacques.levy@inria.fr

Inria, Irif

Journées du patrimoine

Rocquencourt

15 Septembre 2023

<http://jeanjacqueslevy.net/talks/23patrimoine/bogue.pdf>



JUIN 1996



Après l'explosion

dans les marécages près de Kourou



Commission d'enquête

J.-L. Lions, Gilles Kahn



- à la 36ème seconde, le système de guidage par inertie bogue
- alors virage soudain de la fusée
- et explosion

Commission d'enquête

J.-L. Lions, Gilles Kahn

- erreur due à un bogue logiciel dans le SRI2:
 - l'accélération horizontale d'Ariane 5 est 5 fois plus grande que celle d'Ariane 4
 - d'où débordement d'une variable dans le code embarqué du SRI2
 - le programme du SRI2 s'arrête

$$\begin{array}{r} 99999999999999999999 \\ + 2 \\ \hline 10000000000000000001 \end{array}$$

$$= 1,23 \times 10^{16}$$
$$= 12300000000000000$$

Commission d'enquête

J.-L. Lions, Gilles Kahn

- erreur due à un bogue logiciel dans le SRI2:
 - l'accélération horizontale d'Ariane 5 est 5 fois plus grande que celle d'Ariane 4
 - d'où débordement d'une variable dans le code embarqué du SRI2
 - le programme du SRI2 s'arrête
- le programme de secours du SRI1 prend le contrôle:
 - mais le programme du SRI1 est le même que celui du SRI2
 - et il s'est arrêté pour la même raison que pour le SRI2
- sans SRI, la fusée n'a plus aucune bonne direction:
 - déviation des tuyères
 - et explosion

Commission d'enquête

J.-L. Lions, Gilles Kahn



- la partie fautive du code du SRI servait au réalignement d'Ariane 4 au cas où le compte à rebours s'arrête avant $H + 40s$
- ce code était inutile pour Ariane 5
- code gardé car bien testé et marchant bien sur Ariane 4

le code de la PROM

Gilles Kahn,
Robert Ehrlich

During the course of the inquiry board's investigation... were recovered. The values stored in the EEPROMs were identical except for the cycle number at which the failure occurred. Both the telemetry and the values recovered from the EEPROM indicated that an exception had been raised, but more importantly the EEPROM contains the: fault table; real time executive trace table; and exception context table. From this information it is possible to precisely determine the context of the failure including the software instruction executed.

From the information about the failure it can be stated that an Operand Error occurred at 00005FEA. This corresponds to the Ada statement

```
450 P_M_DERIVE(T_ALG.E_BH) := UC_16S_EN_16NS  
(TDB.T_ENTIER_16S  
451 ((1.0/C_M_LSB_BH) *  
452 G_M_INFO_DERIVE(T_ALG.E_BH)));
```

and in particular the FMOVE.W instruction in the compiled assembler

```
* Source line 450, column 7  
000140 F22D 5500 80B0 FMOVE.D -32592(A5),FP2 (IVS, 8268),  
000146 F23C 5523 40F8 FMUL.D # $40F86A0000000000,FP2  
6A00 0000 0000  
000152 F204 7100 FMOVE.W FP2,D4  
000156 3344 000A MOVE.W D4,10(A1)
```

which has been confirmed by later software simulations using the recorded telemetry.

Le code ADA

sum
1996

```
end if;
L_M_DON_32 := TDB.T_ENTIER_32S ((1.0/C_M_LSB_DON) *
                                G_M_INFO_DERIVE(T_ALG.E_DON))
if L_M_DON_32 > 32767 then
  P_M_DERIVE(T_ALG.E_DON) := 16#7FFF#;
elsif L_M_DON_32 < -32768 then
  P_M_DERIVE(T_ALG.E_DON) := 16#8000#;
else
  P_M_DERIVE(T_ALG.E_DON) := UC_16S_EN_16NS(
    TDB.T_ENTIER_16S(L_M_DON_32));
end if;

P_M_DERIVE(T_ALG.E_DOE) := UC_16S_EN_16NS (TDB.T_ENTIER_16S
((1.0/C_M_LSB_DOE) *
G_M_INFO_DERIVE(T_ALG.E_DOE))

L_M_BV_32 := TDB.T_ENTIER_32S ((1.0/C_M_LSB_BV) *
                                G_M_INFO_DERIVE(T_ALG.E_BV));
if L_M_BV_32 > 32767 then
  P_M_DERIVE(T_ALG.E_BV) := 16#7FFF#;
elsif L_M_BV_32 < -32768 then
  P_M_DERIVE(T_ALG.E_BV) := 16#8000#;
else
  P_M_DERIVE(T_ALG.E_BV) := UC_16S_EN_16NS(TDB.T_ENTIER_16S(L_M
end if;

501 P_M_DERIVE(T_ALG.E_BH) := UC_16S_EN_16NS (TDB.T_ENTIER_16S
((1.0/C_M_LSB_BH) *
G_M_INFO_DERIVE(T_ALG.E_BH)))

end LIRE_DERIVE;
--$finprocedure

--(
procedure LIRE_SEUIL (P_M_SEUIL : out TDB.T_ENTIER_16NS) is
  --\
```

Le code ADA

sum
1996

```
end if;
L_M_DON_32 := TDB.T_ENTIER_32S ((1.0/C_M_LSB_DON) *
                                G_M_INFO_DERIVE(T_ALG.E_DON))
if L_M_DON_32 > 32767 then
  P_M_DERIVE(T_ALG.E_DON) := 16#7FFF#;
elsif L_M_DON_32 < -32768 then
  P_M_DERIVE(T_ALG.E_DON) := 16#8000#;
else
  P_M_DERIVE(T_ALG.E_DON) := UC_16S_EN_16NS(
    TDB.T_ENTIER_16S(L_M_DON_32));
end if;

P_M_DERIVE(T_ALG.E_DOE) := UC_16S_EN_16NS (TDB.T_ENTIER_16S
      ((1.0/C_M_LSB_DOE) *
       G_M_INFO_DERIVE(T_ALG.E_DOE))

L_M_BV_32 := TDB.T_ENTIER_32S ((1.0/C_M_LSB_BV) *
                                G_M_INFO_DERIVE(T_ALG.E_BV));
if L_M_BV_32 > 32767 then
  P_M_DERIVE(T_ALG.E_BV) := 16#7FFF#;
elsif L_M_BV_32 < -32768 then
  P_M_DERIVE(T_ALG.E_BV) := 16#8000#;
else
  P_M_DERIVE(T_ALG.E_BV) := UC_16S_EN_16NS(TDB.T_ENTIER_16S(L_M
end if;

501 P_M_DERIVE(T_ALG.E_BH) := UC_16S_EN_16NS (TDB.T_ENTIER_16S
      ((1.0/C_M_LSB_BH) *
       G_M_INFO_DERIVE(T_ALG.E_BH)))

end LIRE_DERIVE;
--$finprocedure

--(
procedure LIRE_SEUIL (P_M_SEUIL : out TDB.T_ENTIER_16NS) is
  --\
```

Le code ADA corrigé

alignement.adb

Fri Feb 7 16:59:10 1997

9

```
L_M_BV_32 := TDB.T_ENTIER_32S ((1.0/C_M_LSB_BV) *
                                G_M_INFO_DERIVE(T_ALG.E_BV));
if L_M_BV_32 > 32767 then
  P_M_DERIVE(T_ALG.E_BV) := 16#7FFF#;
elsif L_M_BV_32 < -32768 then
  P_M_DERIVE(T_ALG.E_BV) := 16#8000#;
else
  P_M_DERIVE(T_ALG.E_BV) := UC_16S_EN_16NS(TDB.T_ENTIER_16S(L_M_BV_32));
end if;
```

502

```
L_M_BH_32 := TDB.T_ENTIER_32S ((1.0/C_M_LSB_BH) *
                                G_M_INFO_DERIVE(T_ALG.E_BH));
if L_M_BH_32 > 32767 then
  P_M_DERIVE(T_ALG.E_BH) := 16#7FFF#;
elsif L_M_BH_32 < -32768 then
  P_M_DERIVE(T_ALG.E_BH) := 16#8000#;
else
  P_M_DERIVE(T_ALG.E_BH) := UC_16S_EN_16NS(TDB.T_ENTIER_16S(L_M_BH_32));
end if;
```

```
end LIRE_DERIVE;
--$finprocedure
```

```
--(
procedure LIRE_SEUIL (P_M_SEUIL : out TDB.T_ENTIER_16NS) is
```

OCTOBRE 1996



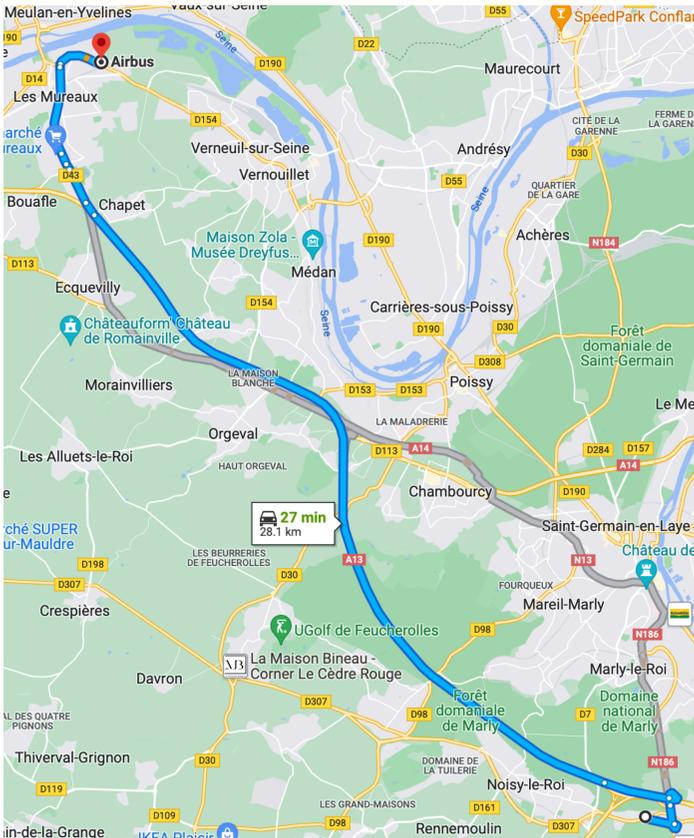
Robert Ehrlich, Georges Gonthier, François Rouaix
Marcin Skubiszewski, Alain Deutsch, Damien Doligez

Alain Deutsch



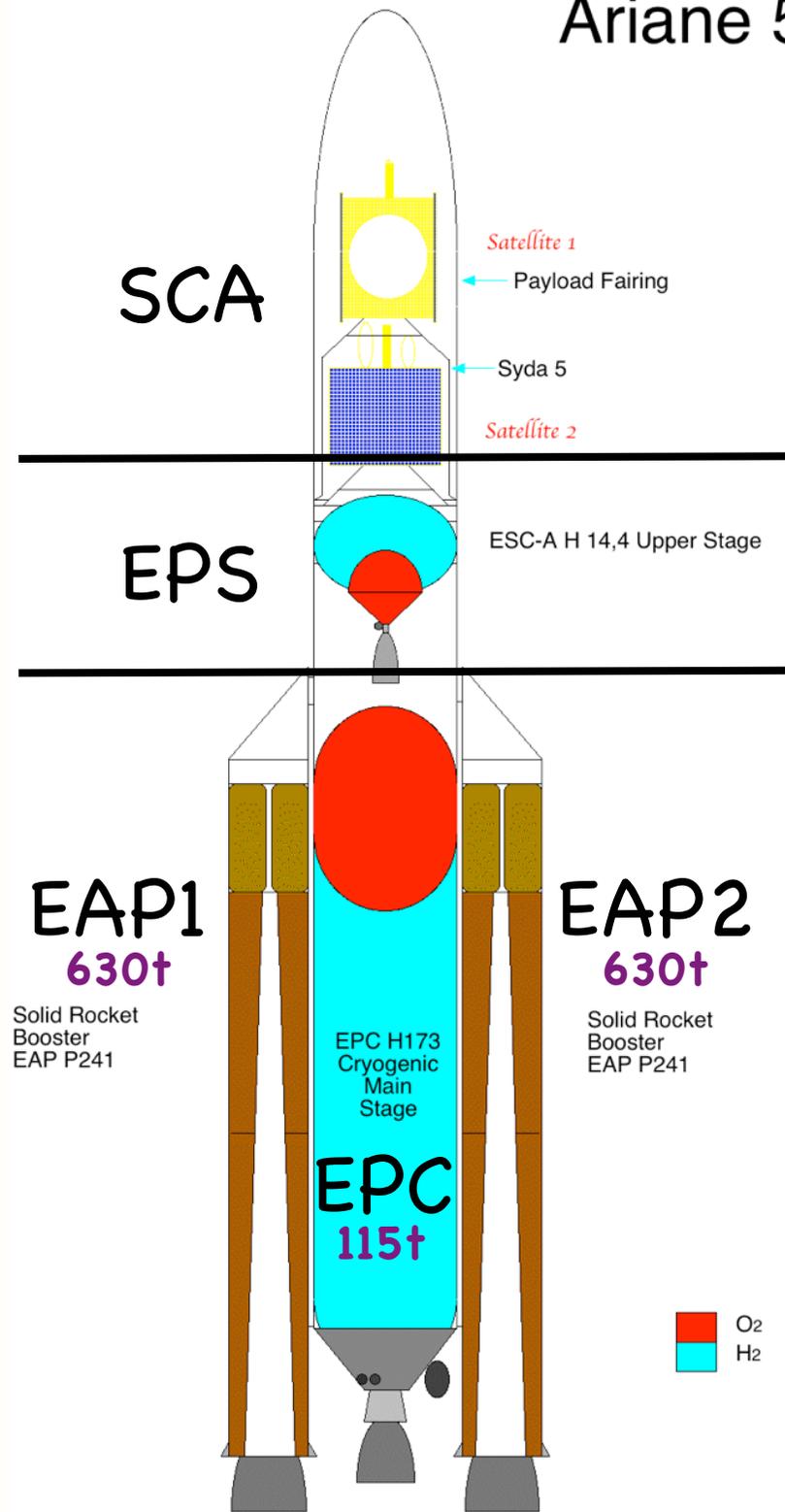
1965 - 2006

Les mercredis





Ariane 5 ECA



Que faire ?

- 140000 lignes de ADA + assembleur 68000
 - analyse “bottom-up”
 - grosse documentation, assez générale
- compiler le code
 - pour le manipuler
- 3 modules logiciel
 - écrit avec des règles strictes de codage
 - multi-tâche avec beaucoup de variables partagées

Que faire ?

- 140000 lignes de ADA

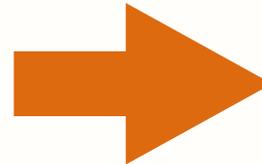
- DEC VMS + VT100's
- grosse documentation

- compiler le code

- compilateur Alsys

- 3 modules logiciel

- écrit avec des règles strictes de codage
- multi-tâche avec beaucoup de variables partagées



bandes format
backup VMS

- 140000 lignes de ADA

- Unix + Sun 3
- Emacs (ADA mode)

- compiler le code

- compilateur GNAT (gratuit)

```

-----
-- Project       : Ariane 5 OnBoard Software
-- Product      : Flight Program (LV)
-- Source File Name : ACTUATORS.ADA
-- Source File History :
--
-- Version   Date   Version Description/Mod Number(s)  Programmer
-----
-- 01.01.00  05.10.94  Initial version                                JCM.SP.
--
-- 01.02.00  05.01.95  UNIT TESTS V1                                  JCM.SP.
--
-- 01.03.00  25.01.95  A5-FA-1811-A53: removed use of S. Parsons
-- COMMUNICATIONS.PUT_COMBUSTION_MESSAGE
-- due to an incorrect and non agreed MN.
--
--           25.01.95  A5-FA-1811-A54: S. Parsons
-- EVFCVi change to FCVi due to an error
-- in bdas
--
-- 01.04.00  28-Feb-95  A5-FA-1811-A100-ASAI: JCM.
-- Add of acquisitions of measures and
-- downstream date from SRI in VARY_SPEED.
--
--           28-Feb-95  A5-FA-1811-A95-ASAI: JCM.
-- The values of the constants AZSRIN and
-- AZSRIS must be exchanged.
--
-- 02.05.00  28.02.95  Updated to ST6.1, TA6.1, HOOD 3.1 S.Parsons,
-- (Baseline V2.0) JCM.
-- Removal of SWITCH_OFF_LVS.
-- Incorporated Mod. note numbers : JCM.
-- A5-MN-1811-0039-ASAI R1
-- A5-MN-1811-0049-ASAI R1
-- A5-MN-1811-0050-ASAI
-- A5-MN-1811-0054-ASAI R2
-- A5-MN-1811-0064-ASAI R1
--
-- 02.06.00  10.05.95  Updated to ST6.1, TA6.2 HOOD 3.1 OR
-- + FMs : S042/95, S058/95 (Baseline V2.1) JCM.
-- Incorporated Mod. note numbers :
-- A5-MN-1811-1143-EASM,
-- A5-MN-1811-1146-EASM,
-- A5-MN-1811-1149-EASM
--
-- 02.07.00  29.05.95  Idem JCM.
--
-- 02.08.00  10.06.95  Idem JCM.
--
-- 02.09.00  21.07.95  BASELINE V2.1 - Correction of NCR-339-ASAI :
-- Add of the reset of out indicators in
-- SCA_AUTOPILOT procedure. JCM.
--
-- 03.10.00  31.10.95  Baseline V3.3 JCM.

```

```

--
-- Source Name      Specification Name      Fortran Model
-----
-- INITIALISE      PRESENTER VALEUR INITIALES CON- INSCAR
--                 -TROLE EN ROULIS (M22551)
--                 INITIALISER CONTROLE EN ROULIS INSCAR
--                 (M22552)
--                 REINITIALISER PARAMETRES SCA SCABAL
--                 (M2261)
--
-- PERFORM_ROLL    CONTROLLER LE ROULIS EN PHASE SCAROUL
--                 EPC/EPS (M22553)
--
-- SCA_AUTOPILOT   REGLER PILOTAGE SCA (M2263) SCABAL
--
-- VARY_SPEED      CONTROLLER LES VARIATIONS DE SCABAL
--                 VITESSE (M2262)
--
-- CHECK_VALVES    COMMANDER/CONTROLLER LES EV N/A
--                 DU SCA (M22556)
--
-- COMMAND_VALVES  COMMANDER/CONTROLLER LES EV N/A
--                 DU SCA (M22556)
--
-- SHORT_EPS_IGNITION_ N/A N/A
-- AUTHORIZED
--
-- GET_PAYLOAD_RELEASE N/A N/A

```

```

-- Implementation of A5-MN-1811-1196-EASM :
-- Add call to COMMUNICATIONS.SET_CYCLIC_OPERATION_ON
-- just after the begin statement of procedures
-- VARY_SPEED and SCA_AUTOPILOT.
--
-- 03.11.00  08.11.95  Baseline V-3.3 - Unit Tested (part.) JCM
--
-- 03.12.00  17.11.95  Baseline V3.4 JCM.
-- - Implementation of FM-S221/95 :
-- Add call of procedure REINIT_SCA_VALVES.
-- - Implementation of A5-MN-1811-1220-EASM (NCR-564) :
-- remove the GET_FUEL_MASS function.
--
-- 03.13.00  29.11.95  Baseline V3.4 JCM.
-- - Implementation of MN-2017 : Add the assignment
-- of OLD_IUOVLV at the beginning of the SCA_AUTOPILOT
-- algorithm.
--
-- 04.14.00  12.01.96  A5-MN-1811-1234-EASM : replacement O. Rigaud
-- of ACTUATORS_MISSION_DATA.AZSRI[N|S]
-- by FLIGHT_MISSION_DATA.AZSRI[N|S]
--
-- 05.15.00  31.07.96  FM S345/96 : computation of O. Rigaud
-- TALLUEPS global variable according
-- to flight phase, before SCA
-- autopilot execution
--
-- OMLVN 05.16.00  16.10.96  A5-MN-1811-1274-EASM, O. Rigaud
-- A5-FA-1811-945-ASAI,
-- A5-FA-1811-947-ASAI : addition of
-- GET_PAYLOAD_RELEASE_NUMBER function
--
-- Constituent Program Units : 8
--
-- INITIALISE
-- PERFORM_ROLL
-- SCA_AUTOPILOT
-- VARY_SPEED
-- CHECK_VALVES
-- COMMAND_VALVES
-- SHORT_EPS_IGNITION_AUTHORISED
-- GET_PAYLOAD_RELEASE_NUMBER
--
-- Source File Description :
--
-- ACTUATORS shall calculate roll commands which are sent to the actuators
-- as a series of FCVi and SCA LV on/off commands. It shall also verify
-- the operation of the FCVi and SCA LVs and take the appropriate
-- recovery actions.
-- The resultant recovery actions shall be made available to the caller.
--
-- ACTUATORS shall generate roll commands for the EPC, EPS and SCA flight
-- phases. For the SCA phase it will manage the variation of the speed and
-- allow the fuel mass to be obtained.
--
-- Portability Considerations :
--
-- NONE.
--
-- Subprogram Mapping :

```

```

--
-- *****
-- FUNCTIONAL DESCRIPTION:
--
-- Initialises, or reinitialises, the parameters for each beginning of a
-- new flight phase (EPC, EPS, SCAi).
-- (see FM A5-MN-1811-0037-ASAI)
--
--
-- procedure INITIALISE is separate;
-- *****
--
-- FUNCTIONAL DESCRIPTION:
--
-- Activates the roll control calculations and sends the calculated
-- roll commands to the actuators.
--
--
-- procedure PERFORM_ROLL is separate;
-- *****
--
-- FUNCTIONAL DESCRIPTION:
--
-- Performs the autopilot during the SCA phase and
-- calculates SCA commands for the actuators.

```

```

ACTUATORS_INTERNAL.CALCULATE_TBRN;
if ACTUATORS_INTERNAL.FIRST then
  ACTUATORS_INTERNAL.FIRST := false;
end if;
end if;

-- To update the digital message
COMMUNICATIONS.PUT_SCA_AUTOPILOT_2_MESSAGE
(MESSAGE =>
  (DTOUV => ACTUATORS_INTERNAL.TBRN_TAB,
  PMASSIV => ACTUATORS_INTERNAL.PMASS,
  ANOMAL => ACTUATORS_INTYPES.NOZZLE_FAILURE_TYPE'pos
    (ACTUATORS_INTERNAL.ANOMAL),
  CASEI => ACTUATORS_INTERNAL.PILOT_MODE,
  IANOM => COMMUNICATIONS_TYPES.IANOM_TYPE
    (ACTUATORS_INTERNAL.IANOM),
  ITEST => ACTUATORS_INTYPES.FAILURE_TYPE'pos
    (ACTUATORS_INTERNAL.ITEST)));

COMMUNICATIONS.PUT_SCA_AUTOPILOT_1_MESSAGE
(MESSAGE =>
  (STATE => ACTUATORS_INTERNAL.STATE_VCT,
  UNBAL_2 => ACTUATORS_INTERNAL.UNBAL (2),
  UNBAL_3 => ACTUATORS_INTERNAL.UNBAL (3),
  DVFIL => ACTUATORS_INTERNAL.DVFIL,
  QS => ACTUATORS_INTERNAL.QS,
  RATES => ACTUATORS_INTERNAL.ROLL_RATES));

-- Output parameters :
RESET_UCTM_REQUIRED := ACTUATORS_INTERNAL.RESET_UCTM_TO_REQUIRE;
PAYLOAD_RELEASE_REQUIRED :=
  ACTUATORS_INTERNAL.RELEASE_PAYLOAD_INDICATOR;
STOP_PROCESSING_SCA := ACTUATORS_INTERNAL.FLPASC;

-- To reset the indicators :
ACTUATORS_INTERNAL.RESET_UCTM_TO_REQUIRE := false;
ACTUATORS_INTERNAL.RELEASE_PAYLOAD_INDICATOR := false;
ACTUATORS_INTERNAL.FLPASC := false;

end SCA_AUTOPILOT;
-- ++++++

```

```

COMMUNICATIONS.SET_CYCLIC_OPERATION_ON (
  NAME => COMMUNICATIONS_TYPES.TRAJ_CONTROL_SCA_SPEED_ACTIVATED);

-- To check variations of speed :
-----
-- Initialisation of roll component of accelerometric increment
ACTUATORS_INTERNAL.X_ACCEL_INCREMENT :=
  SRI.GET_ACCEL_INCREMENT_IN_X_AXIS;

-- Acquisition date
ACTUATORS_INTERNAL.DOWNSTREAM_DATE := SRI.GET_DOWNSTREAM_DATE;
-- SRI validity
ACTUATORS_INTERNAL.NUMSRI := SRI.GET_VALIDITY;

-- Check of SRI switch :
if ACTUATORS_INTERNAL.NUMSRI /= ACTUATORS_INTERNAL.NUMSRI and then
  ACTUATORS_INTERNAL.NUMSRI = FLIGHT_TYPES.BACKUP_ONLY
then
  ACTUATORS_INTERNAL.FIRST := true;
  ACTUATORS_INTERNAL.VXBIAS :=
    ACTUATORS_INTERNAL.VXBIAS +
    ( ACTUATORS_INTERNAL.X_ACCEL_INCREMENT -
    ACTUATORS_INTERNAL.VXMOLD );
  ACTUATORS_INTERNAL.COS_AZSRI :=
    UNIVERSAL.COS (FLIGHT_MISSION_DATA.AZSRIS);
  ACTUATORS_INTERNAL.SIN_AZSRI :=
    UNIVERSAL.SIN (FLIGHT_MISSION_DATA.AZSRIS);
  ACTUATORS_INTERNAL.NUMSRI := ACTUATORS_INTERNAL.NUMSRI;
end if;

-- Check of speed and saving of date and value if any change appeared :
if ACTUATORS_INTERNAL.X_ACCEL_INCREMENT /= ACTUATORS_INTERNAL.VXMOLD
then
  ACTUATORS_INTERNAL.DVSKIP := ACTUATORS_INTERNAL.DOWNSTREAM_DATE;
  ACTUATORS_INTERNAL.VXMOLD := ACTUATORS_INTERNAL.X_ACCEL_INCREMENT;
end if;

end VARY_SPEED;
-- ++++++

```

actuators.ada Wed Oct 16 09:59:41 1996 7

```

--
-- FUNCTIONAL DESCRIPTION:
--
-- Performs a check on the current status of the valves.
-- The check shall be made with respect to sent commands and recovery
-- action shall be taken. The status of the ESS shall be indicated
-- to the caller.
-- The status has no meaning on each call after SCA_AUTOPILOT
-- and therefore should not be checked.
--
-- FORMAL PARAMETERS:
--
-- Return parameter : Status of the ESS, as returned by LN2 level.
--
--
function CHECK_VALVES
return FLIGHT_TYPES.ELECTRO_SOLENOID_STATUS_TYPE is separate;
-- ++++++
--
-- FUNCTIONAL DESCRIPTION:
--
-- Sends the commands to open and close the actuators.
-- These commands have been previously calculated by the roll control and
-- sca autopilot algorithms.
--
--
procedure COMMAND_VALVES is separate;
-- ++++++
--
-- FUNCTIONAL DESCRIPTION:
--
-- Returns the authorisation state of the short eps ignition.
-- ST variable 'xreals'.
--
--
function SHORT_EPS_IGNITION_AUTHORISED return BOOLEAN is
begin
return ACTUATORS_INTERNAL.XREALS;

```

actuators.ada Wed Oct 16 09:59:41 1996 8

```

begin
return ACTUATORS_INTERNAL.PAYLOAD_NBR;
end GET_PAYLOAD_RELEASE_NUMBER;
-- ++++++
end ACTUATORS;

```

```

-- *****
-- FUNCTIONAL DESCRIPTION:
--
-- This function allows to calculate the quaternion error.
--
-- FORMAL PARAMETERS:
--
-- A : First quaternion
-- B : Second quaternion
--
-- DESIGN:
--
-- The error is calculate by multiplying the first quaternion by
-- the inversion of the second quaternion
--
-- [optional subprogram tags]
-- RETURN VALUE:
--
-- The quaternion error.
--
--
function CALC_QUATERNION_ERROR
( A : in FLIGHT_TYPES.ARRAY_1_4_FLOAT_TYPE;
  B : in FLIGHT_TYPES.ARRAY_1_4_FLOAT_TYPE )
return FLIGHT_TYPES.ARRAY_1_4_FLOAT_TYPE is

  INVERTED_QUATERNION :
    FLIGHT_TYPES.ARRAY_1_4_FLOAT_TYPE := (others => 0.0);
  QUATERNION_ERROR :
    FLIGHT_TYPES.ARRAY_1_4_FLOAT_TYPE := (others => 0.0);

begin
  -- To build the negative quaternion :
  for I in 1 .. FLIGHT_TYPES.ARRAY_1_4_FLOAT_TYPE'last - 1
  loop
    INVERTED_QUATERNION (I) := -(B (I));
  end loop;

  INVERTED_QUATERNION (FLIGHT_TYPES.ARRAY_1_4_FLOAT_TYPE'last) :=
    B (FLIGHT_TYPES.ARRAY_1_4_FLOAT_TYPE'last);

  -- Multiply the quaternions :

```

```

actuators_utilities_ada      Mon Jun 12 20:05:29 1995      1
-----
-- Project           : Ariane 5 OnBoard Software
-- Product           : Flight Program (LV)
-- Source File Name  : ACTUATORS_UTILITIES_ADA
-- Source File History :
--
--      Version   Date   Version Description/Mod Number(s) Programmer
--      -----
--      00.00.01  05.10.94  Initial version                               JCM
--
--      00.00.02  01.02.95  Update to ST 6.1, coding standards           S. Parsons
--
--      02.03.00  22.05.95  Idem                                         JCM.
--
--      02.04.00  10.06.95  Idem (Baseline V2.1).                       JCM.
--
-- Constituent Program Units : 5
--
-- MTXMUL_GEN
-- LIMIT
-- CALC_ANGULAR_VELOCITY
-- MULTIPLY_QUATERNIONS
-- CALC_QUATERNION_ERROR
--
-- Source File Description :
-- This package contains some usefull operations and utilities for ACTUATORS
-- functionalities.
-- Portability Considerations :
--
-- {tbs}
--
-- Subprogram Mapping :
--
-- Source Name          Specification Name          Fortran Model
-- -----
--
-- MTXMUL_GEN          N/A
-- LIMIT               LIMITER SYMETRIQUEMENT LES          Unknown
-- ENTREES(M22644)

```

```

actuators_utilities_ada      Mon Jun 12 20:05:29 1995      2
-----
package ACTUATORS_UTILITIES is
--
-- *****
-- FUNCTIONAL DESCRIPTION:
--
-- This function performs a matrix vector multiplication .
--
-- FORMAL PARAMETERS:
--
-- M : input matrix.
-- V : input vector.
--
-- RETURN VALUE:
--
-- output vector.
--
--
generic
  type ITEM_TYPE is private;
  NULL_ITEM : ITEM_TYPE;

  type FIRST_INDEX_TYPE is range <>;
  type SECOND_INDEX_TYPE is range <>;

  type MATRIX_TYPE is array (FIRST_INDEX_TYPE, SECOND_INDEX_TYPE) of
    ITEM_TYPE;
  type IN_VECTOR_TYPE is array (SECOND_INDEX_TYPE) of ITEM_TYPE;
  type OUT_VECTOR_TYPE is array (FIRST_INDEX_TYPE) of ITEM_TYPE;

  with function "*" (X, Y : ITEM_TYPE) return ITEM_TYPE is <>;
  with function "+" (X, Y : ITEM_TYPE) return ITEM_TYPE is <>;

function MTXMUL_GEN
( M : in MATRIX_TYPE;
  V : in IN_VECTOR_TYPE) return OUT_VECTOR_TYPE;
-- *****

--
-- *****
-- procedure CALC_ANGULAR_VELOCITY
-- (FAILURE : in ACTUATORS_INTYPES.FAILURE_TYPE);
-- *****
--
-- *****
-- FUNCTIONAL DESCRIPTION:
--

```

```

-- A : First quaternion
-- B : Second quaternion
-- DESIGN:
-- The error is calculate by multiplying the first quaternion by
-- the inversion of the second quaternion
-- RETURN VALUE:
-- The quaternion error.
--
function CALC_QUATERNION_ERROR
( A : in FLIGHT_TYPES.ARRAY_1_4_FLOAT_TYPE;
  B : in FLIGHT_TYPES.ARRAY_1_4_FLOAT_TYPE )
return FLIGHT_TYPES.ARRAY_1_4_FLOAT_TYPE;
-- *****
-- FUNCTIONAL DESCRIPTION:
-- This function allows to limit the input value (X) to the limit (LIMIT).
-- FORMAL PARAMETERS:
-- X : FLOAT input value
-- LIMIT : Limit, has to be positive
-- RETURN VALUE:
-- clipped value of the parameter
--
function LIMIT
( X : in FLIGHT_TYPES.FLOAT_TYPE;
  XLIMIT : in FLIGHT_TYPES.FLOAT_TYPE )
return FLIGHT_TYPES.FLOAT_TYPE;
-- *****
end ACTUATORS_UTILITIES;

```

```

-- Project Name : Ariane 5 OnBoard Software - Flight Program (LV)
-- Source File Name : ACYCLIC1.ADA
-- Source File History :
--
-- Version Date Version Description/Mod Number(s) Programmer
-- -----
-- 00.01 25.10.94 Initial Version PCB
-- 00.02 18.11.94 Mod Note 1041,1063 PCB
-- 00.03 12.12.94 communications.process_it_bf PCB
-- 00.04 23.02.95 Mod Note 1077 SJH
-- 00.05 23.02.95 Mod Note 1077 correction SJH
-- 00.06 07.03.95 Mod Note 1076 VT
-- Also updated for UT purposes
-- by creating an ACYCLIC1_INTERNAL
-- package.
-- 00.07 31.03.95 Updated in line with MN 1135. VT
-- 00.08 26.04.95 Updated in line with MN 1136. VT
-- 00.09 15.05.95 Correct header and correct SJH
-- Mod Note 1136.
-- 00.10 19.06.95 NCR 138 incorporated SJH
-- 00.11 10.08.95 Implement the MN 1170 A. Peres
-- and MN 1148, 1076
-- 00.12 25.08.95 Add call to PHASE.SET (MN 1104)
-- in SET_EAP_IGNITION P. Bricker
-- 00.13 29.08.95 Add OBC1 guard on IGNITE_VULCAIN
-- and PREPARE_EAP_IGNITION (NCR 437),
-- and add EAPs ignited guard on
-- DISCONNECT_EPC_ES (NCR 300) P. Bricker
-- 00.14 23.11.95 Add call to
-- PROPULSION.SWITCH_EPC_TO_FLIGHT_MODE
-- in SET_EAP_IGNITION (NCR 568) P. Bricker
-- OMLVN 00.15 05.01.96 Add call to LAUNCHER.INHIBIT_OBC_SWITCH JCM
-- in SET_EAP_IGNITION in order to perform
-- OBC switch inhibition the same way on
-- both OBCs. FM-S237/95.
-- Operation OBC_IS now belongs to COMMUNICATIONS,
-- according to MN-1234.
--
-- Source File Description :

```

```

acycl1c1.ada      Fri Jan 12 17:22:07 1996      2
--
-- Package body for the object ACYCLIC1
--
-- Target Dependencies :
--
-- None
--
-- Subprogram Mapping :
--
-- Source Name          Specification Name          Fortran Model
-- -----
-- ACYCLIC1.ADA         M211 (Partially)          None
--
-----
--H Ada package body for object ACYCLIC1
--H Generated by HOOD Toolset at 18:48:02_on_22-09-1994
--H OBJECT ACYCLIC1 IS ACTIVE
--
--H REQUIRED INTERFACE
With ACYCLIC1_INTERNAL;
with TRAJECTORY_TYPES;
with COMMUNICATIONS;
with COMMUNICATIONS_TYPES;
with PROPULSION;
with ACYCLIC2;
with PHASE;
with PHASE_TYPES;
With LNL;
With LNL_TYPES;
with LAUNCHER;

Package Body ACYCLIC1 Is

-- Renames of operation in order to improve readability :
function "=" (LEFT, RIGHT : COMMUNICATIONS_TYPES.OBC_TYPE) return BOOLEAN
renames COMMUNICATIONS_TYPES."=";

-- PROVIDED OPERATIONS

```

```

acycl1c1.ada      Fri Jan 12 17:22:07 1996      3
-- DESCRIPTION
-- This procedure stops the DASDC message being sent and causes
-- the activation of the SRI release sequence.
RC : LNL_TYPES.RC_T;
Begin
-- Stop the 72 ms delay for sending the DASDC messages.
LNL.T10_STOP_SUSPENDED_DELAY ( TASK_ID => ACYCLIC1_INTERNAL.ACYCLIC1_TASK,
RC => RC);
End ACTIVATE_SRI;

Procedure IGNITE_VULCAIN Is
-- DESCRIPTION
-- Allows the Vulcain ignition sequence to be activated.
Begin
if COMMUNICATIONS.OBC_IS = COMMUNICATIONS_TYPES.OBC1 then
ACYCLIC1_INTERNAL.OBCS.IGNITE_VULCAIN;
end if;

End IGNITE_VULCAIN;

Procedure PREPARE_EAP_IGNITION Is
-- DESCRIPTION
-- Allows EAP ignition
begin
if COMMUNICATIONS.OBC_IS = COMMUNICATIONS_TYPES.OBC1 then
ACYCLIC1_INTERNAL.OBCS.EAP_PRE_IGNITION;
end if;

End PREPARE_EAP_IGNITION;

procedure SET_EAP_IGNITION is
-- DESCRIPTION
-- This operation is to be called on the redundant OBC following
-- the observation of the CEX closure during the EAP ignition
-- window,
-- or when the EAPs have been ignited in the nominal OBC.
-- Performs also the inhibition of the OBC switch as soon
-- as were are in flight : called by ACYCLIC2 if OBC is BC
-- and by EXECUTIVE if OBC is RT.
begin

```

```

--
ACYCLIC1_INTERNAL.ES_TO_BE_DISCONNECTED := ES;

select
  ACYCLIC1_INTERNAL.OBCS.DISCONNECT_EPC_ES;
else
  ACYCLIC1_INTERNAL.EPC_ES_DISCONNECTION_REQUESTED := TRUE;
end select;

else

  -- EAPs have not yet ignited, ACYCLIC2 performs the
  -- DISCONNECT_EPC_ES
  ACYCLIC2.DISCONNECT_EPC_ES(ES => ES);

end if;

End DISCONNECT_EPC_ES;

Procedure DISCONNECT_CASE_ES (ES : In PROPULSION_TYPES.ES_TYPE) Is
-- DESCRIPTION
-- Allows the CASE electrical system disconnection sequence to be
-- activated
begin
  ACYCLIC1_INTERNAL.ES_TO_BE_DISCONNECTED := ES;
  select
    ACYCLIC1_INTERNAL.OBCS.DISCONNECT_CASE_ES;
  else
    ACYCLIC1_INTERNAL.CASE_ES_DISCONNECTION_REQUESTED := TRUE;
  end select;
End DISCONNECT_CASE_ES;

Procedure COMMUTE Is
-- DESCRIPTION
-- This operation should not exist????
Begin
  ACYCLIC1_INTERNAL.OBCS.COMMUTE;
End COMMUTE;

Procedure COMMUTE_EPE (FAILED_AXIS : in TRAJECTORY_TYPES.JACK_AXIS_TYPE) Is
-- DESCRIPTION
-- This operation activates the entry point COMMUTE_EPE in
-- task OBCS.
Begin

```

```

begin
  return ACYCLIC1_INTERNAL.FRAME_STARTED_IN_OBC2 ;
end FRAME_STARTED_BY_OBC2 ;

function EAPS_HAVE_IGNITED return BOOLEAN is
-- DESCRIPTION
-- This operation is to return TRUE when it has detected that
-- the EAP booster engines have ignited
--
begin
  return ACYCLIC1_INTERNAL.EAPS_IGNITED ;
end EAPS_HAVE_IGNITED ;

End ACYCLIC1;

```

```

acycllic1_ada      Fri Aug  4 11:13:31 1995      1

--H Ada package specification for object ACYCLIC1
--H Generated by HOOD Toolset on 07:51:00_on_21-09-1994

--H OBJECT ACYCLIC1 IS ACTIVE
--H DESCRIPTION
--H \FC
--H -----|-----
--H --|
--H -- Project Name: Ariane 5 OnBoard Software - PV
--H --
--H -- Object Name: ACYCLIC1
--H --
--H -- Object History:
--H --
--H -- Version   Date      Version Description/Mod Number(s) Designer
--H -- -----|-----|-----|-----|-----|-----
--H -- 00.00    19.09.94    5.1.
--H -- 00.01    15.02.95    ST6.1.1 Modifications :-
--H --                               Mod Note 1076, new SET_EAP_IGNITION
--H --                               procedure.
--H -- 00.02    26.04.95    Updated in line with MN 1136.
--H -- 00.03    02.08.95    Updated as per
--H --                               A5-MN-1811-1170-EASM
--H --                               C.Jones
--H -- 00.04    04.08.95    Updated as per
--H --                               A5-MN-1811-1076-EASM-R3
--H --                               A5-MN-1811-1148-EASM
--H --                               C.Jones
--H --
--H -----|-----|-----|-----|-----|-----|-----
--H --
--H ACYCLIC1 shall allow the SRI release sequence to be activated from the
--H operation ACTIVATE_SRI.
--H
--H The Vulcain may be ignited via the operation IGNITE_VULCAIN. The date of
--H ignition shall be obtained from DATE and the sequence activated. If the
--H sequence is successful the EAPs may then be ignited otherwise the Vulcain
--H shall be stopped (via ACYCLIC2).
--H
--H As described above once the Vulcain has been correctly ignited the EAP
--H ignition sequence may be activated via the operation PREPARE_EAP_IGNITION.
--H This operation activates the EAP pre-ignition sequence. If the EAP
--H pre-ignition is successful ACYCLIC1 shall suspend until just before the
--H EAP ignition is activated, it shall then activate the EAP ignition sequenc
--H (via ACYCLIC2) otherwise the Vulcain stop sequence shall be activated.
--H
--H Once the EAPs are ignited one may only disconnect the CASE or EPC electric
--H systems via the operations DISCONNECT_CASE_ES and DISCONNECT_EPC_ES. If

```

```

acycllic1_ada      Fri Aug  4 11:13:31 1995      2

--H
--H \FP
--H IMPLEMENTATION AND SYNCHRONISATION CONSTRAINTS
--H NONE
with PROPULSION_TYPES;
with TRAJECTORY_TYPES;
package ACYCLIC1 is
--H PROVIDES OPERATIONS

  procedure START;

  procedure ACTIVATE_SRI;
  --H LSER      -- Allows the SRI release sequence to be activated
  --

  procedure IGNITE_VULCAIN;
  --H LSER      -- Allows the ignition sequence for both the Vulcain to be
  -- activated. The EPC ignition sequence will be activated
  -- at the date of H0.
  --

  procedure PREPARE_EAP_IGNITION;
  --H LSER      -- Allows the EAP pre-ignition and ignition sequence to be
  -- activated. The pre-ignition sequence shall be activated
  -- and if it is successful the EAP ignition sequence shall
  -- be activated. In the case of the pre-ignition being
  -- unsuccessful the EAP ignition shall be aborted and the
  -- EPC stop sequence shall be activated.

  procedure DISCONNECT_EPC_ES (ES : in PROPULSION_TYPES.ES_TYPE);
  -- Allows an EPC electrical system disconnection sequence to
  -- be activated.
  --

  procedure DISCONNECT_CASE_ES (ES : in PROPULSION_TYPES.ES_TYPE);
  --H LSER      -- Allows a CASE electrical system disconnection sequence to
  -- be activated.
  --

  procedure COMMUTE;
  -- Allows the commutation to be handled and the recovery
  -- sequences to be activated.

  procedure SET_EAP_IGNITION;
  -- This operation is to be called on the redundant OBC following the observati

```

Variables partagées

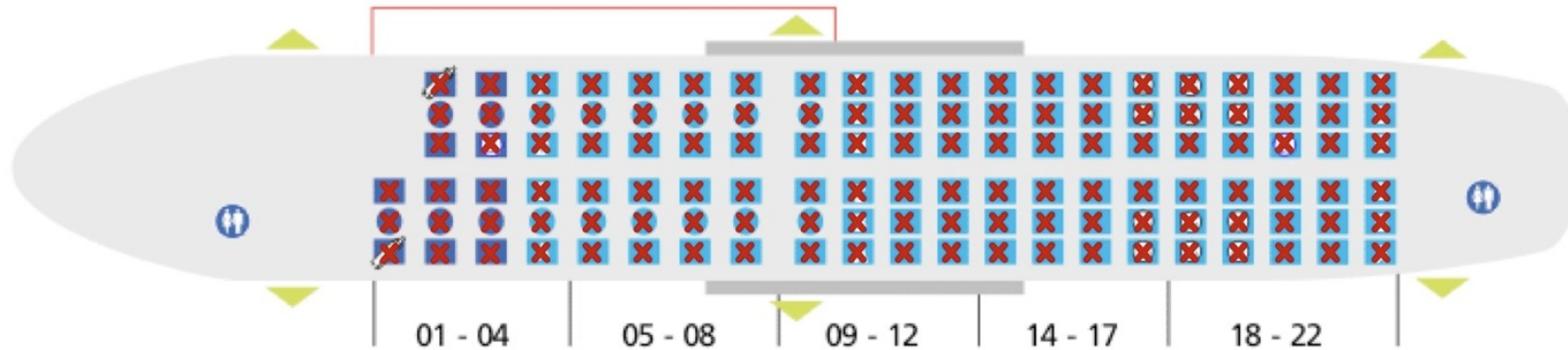
Airbus A 318 "Europe"
118 passagers

L'ESPACE AFFAIRES

Tempo



Cabine L'Espace Affaires modulable en fonction du nombre de passagers



- Sorties & Sorties de secours
- Emplacement des ailes
- Sièges neutralisés

- Toilettes
- Berceaux

- Sièges pour enfants non accompagnés
- Sièges pour passagers devant être portés de/vers leur siège

COPYRIGHT AIR FRANCE - REPRODUCTION INTERDITE



Bob

y-a-t'il une place ?

"oui"

je la prends

y-a-t'il une place ?

"oui"

je la prends



Alice

Variables partagées

- soient 2 tâches T_1 (alice) et T_2 (bob)
- x est une variable partagée entre T_1 et T_2 (nb de sièges libres)

T_1

```
If  $x > 0$  then  
  Réserver ();  
   $x := x - 1$ ;
```

T_2

```
If  $x > 0$  then  
  Réserver ();  
   $x := x - 1$ ;
```

Variables partagées

- soient 2 tâches T_1 (alice) et T_2 (bob)
- x est une variable partagée entre T_1 et T_2 (nb de sièges libres)

T_1

```
If  $x > 0$  then  
   $x := x - 1;$   
  Réserver();
```

T_2

```
If  $x > 0$  then  
   $x := x - 1;$   
  Réserver();
```

Détection des variables partagées

- PV (programme de vol) est un programme Ada avec 5 tâches
- trouver les variables partagées !
- 80000 lignes ADA + 10000 lignes d'assembleur MC68000
- en prenant compte des alias

Alias

```
function RéserveAlice (avion) =  
  If avion.placeLibres > 0 then  
    RéserverSiègeAlice(avion);  
    avion.placeLibres := avion.placeLibres - 1;;
```

```
function RéserveBob (avion) =  
  If avion.placeLibres > 0 then  
    RéserverSiègeBob(avion);  
    avion.placeLibres := avion.placeLibres - 1;;
```

```
function VoyageMSR () =  
  RéserveAlice (airbusAF318);  
  RéserveBob (boeingAF737);;
```

The diagram consists of three curved arrows pointing downwards and to the right. The top arrow starts at the end of the 'RéserveAlice' function definition and points to the 'airbusAF318' argument in the 'VoyageMSR' function. The middle arrow starts at the end of the 'RéserveBob' function definition and points to the 'boeingAF737' argument in the 'VoyageMSR' function. The bottom arrow starts at the end of the 'RéserveAlice' function definition and points to the 'RéserveBob' function call within the 'VoyageMSR' function.

Alias

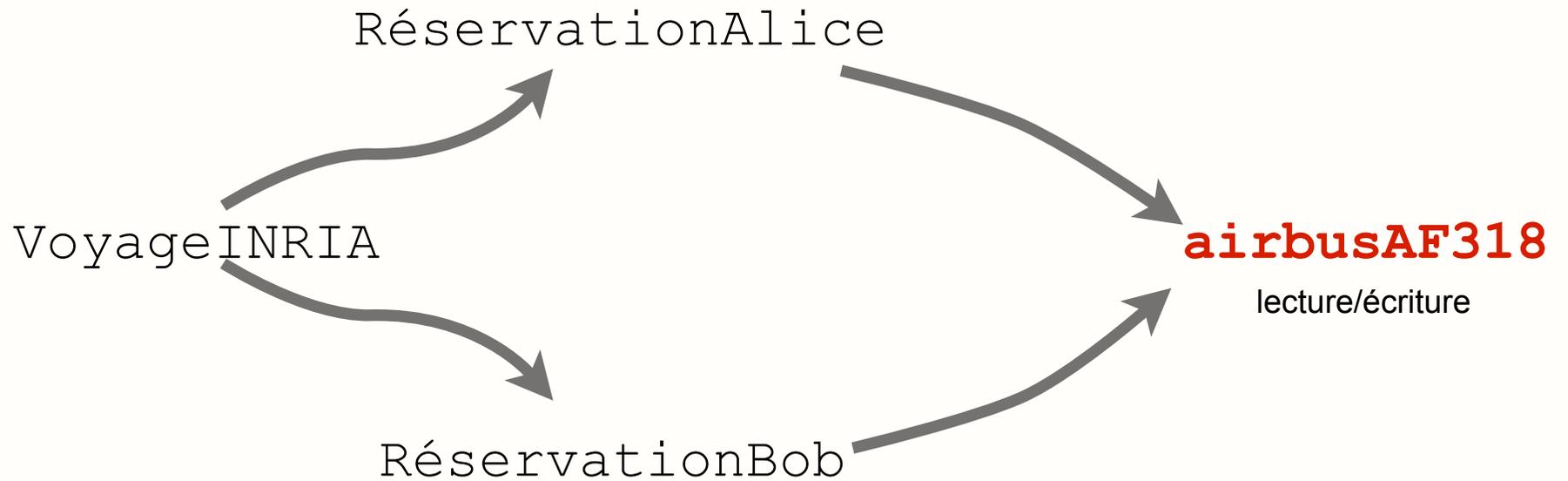
```
function RéserveAlice (avion) =  
  If avion.placeLibres > 0 then  
    RéserverSiègeAlice(avion);  
    avion.placeLibres := avion.placeLibres - 1;;
```

```
function RéserveBob (avion) =  
  If avion.placeLibres > 0 then  
    RéserverSiègeBob(avion);  
    avion.placeLibres := avion.placeLibres - 1;;
```

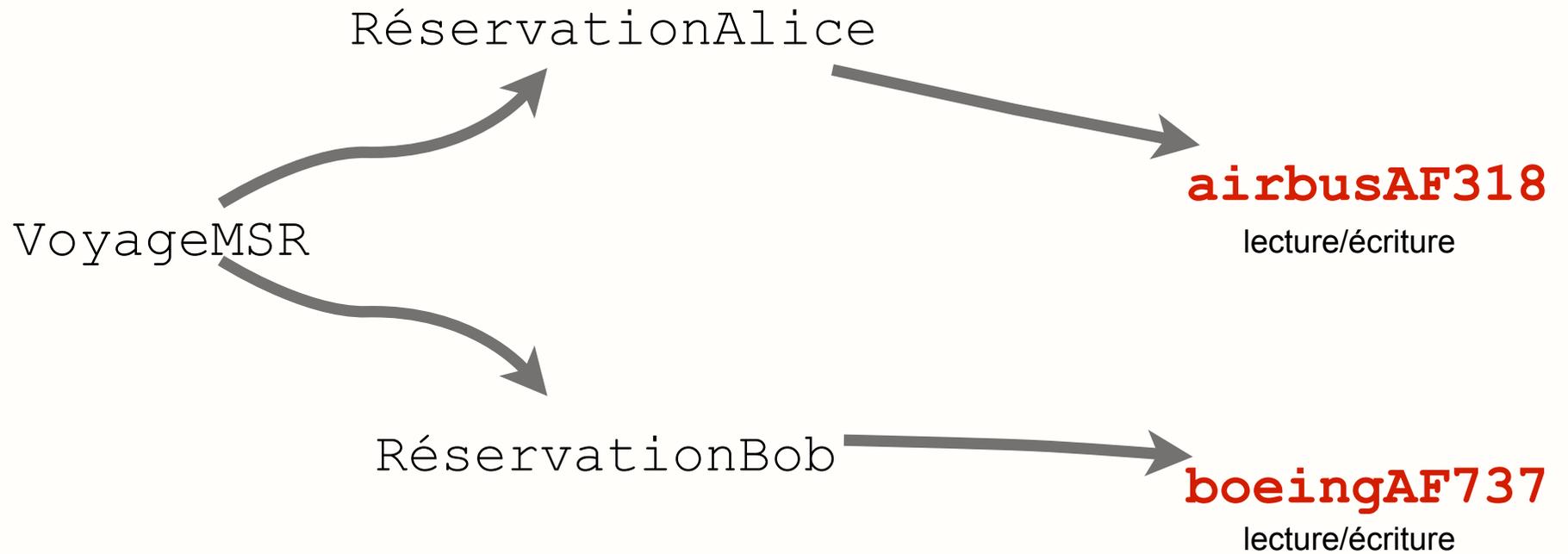
```
function VoyageINRIA () =  
  RéserveAlice (airbusAF318);  
  RéserveBob (airbusAF318);;
```



Accès aux variables partagées



Accès aux variables partagées

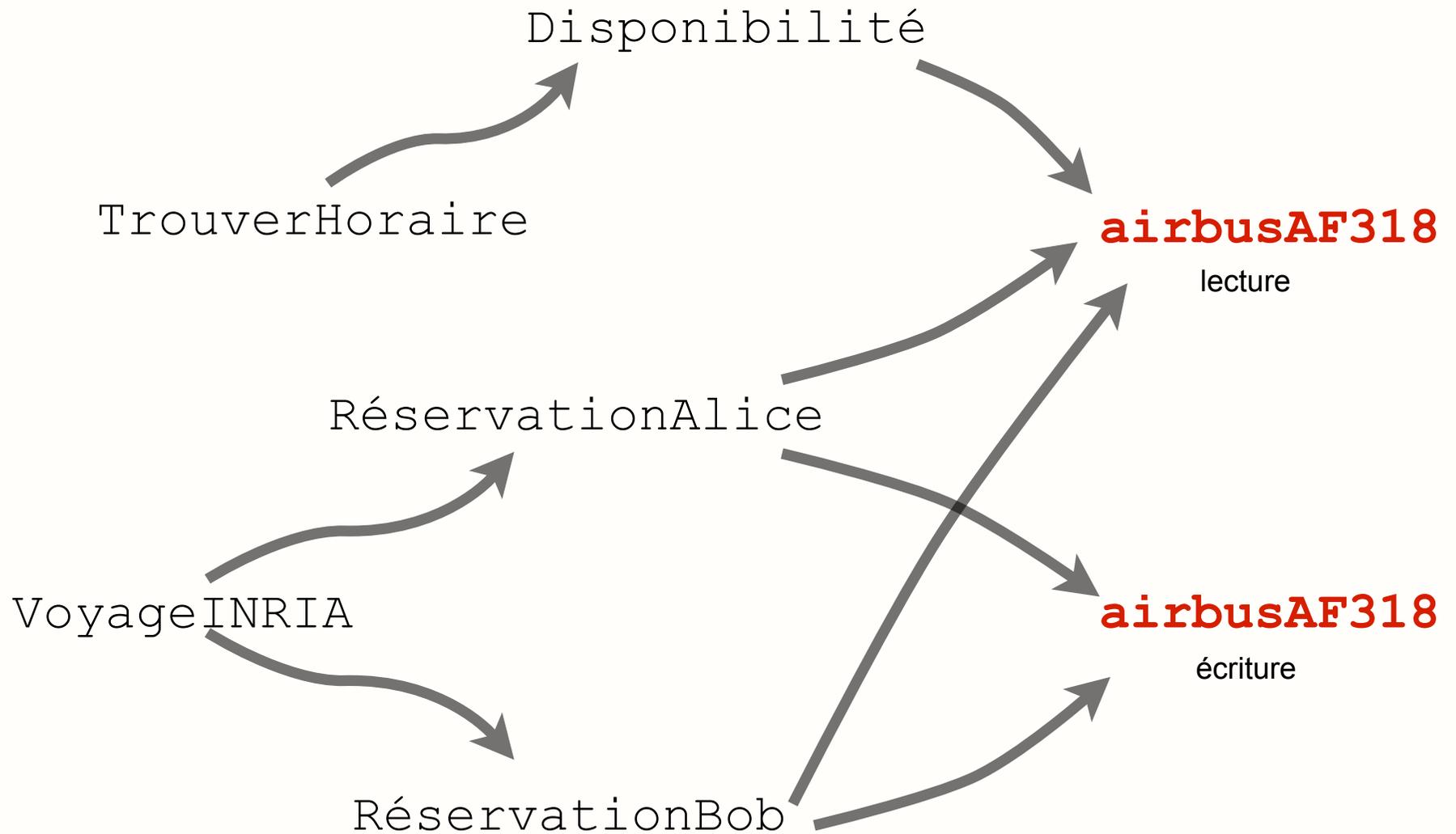


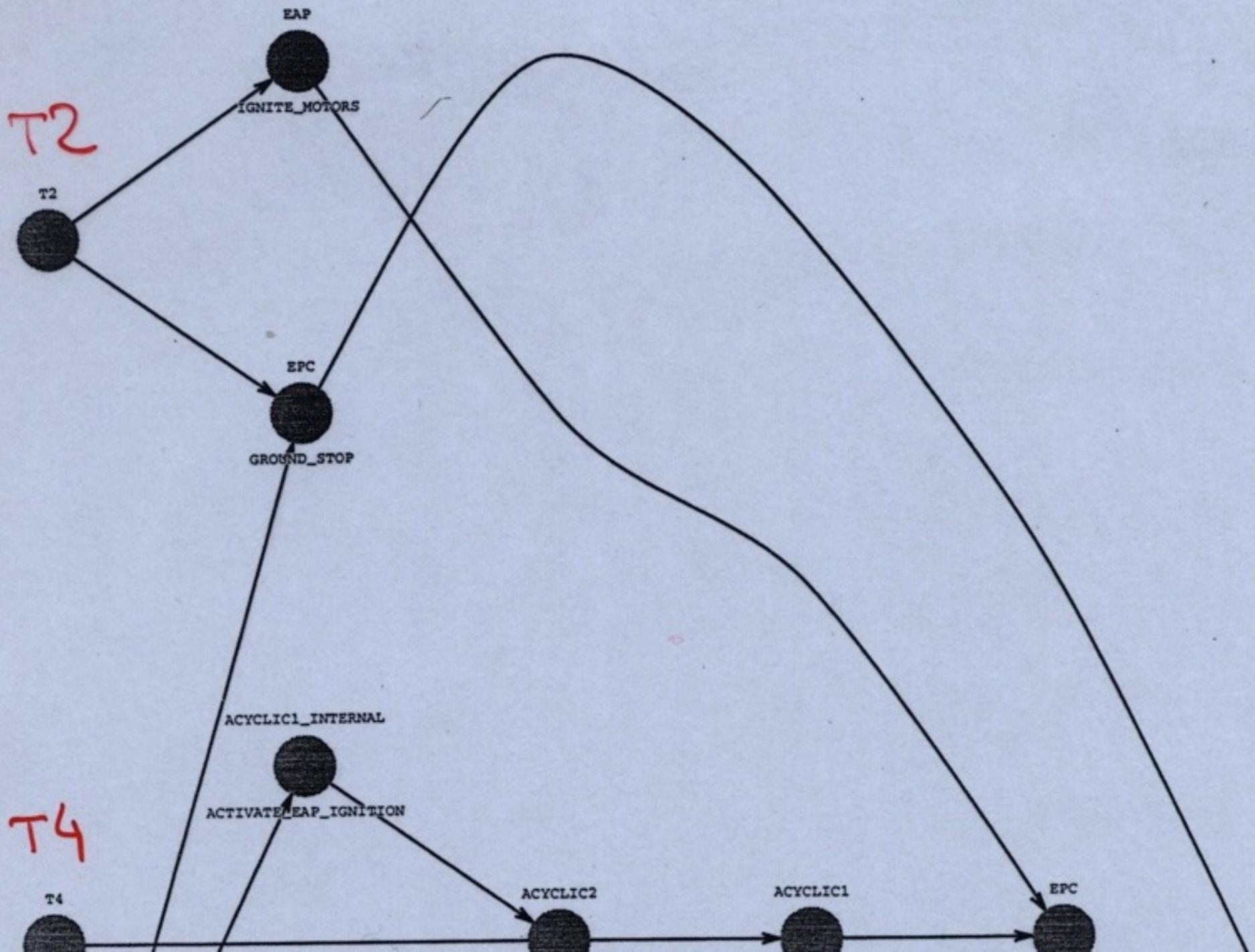
Alias

```
function Disponibilité (avion) : string =  
  If avion.placeLibres > 0 then  
    return "oui" else "non";;
```

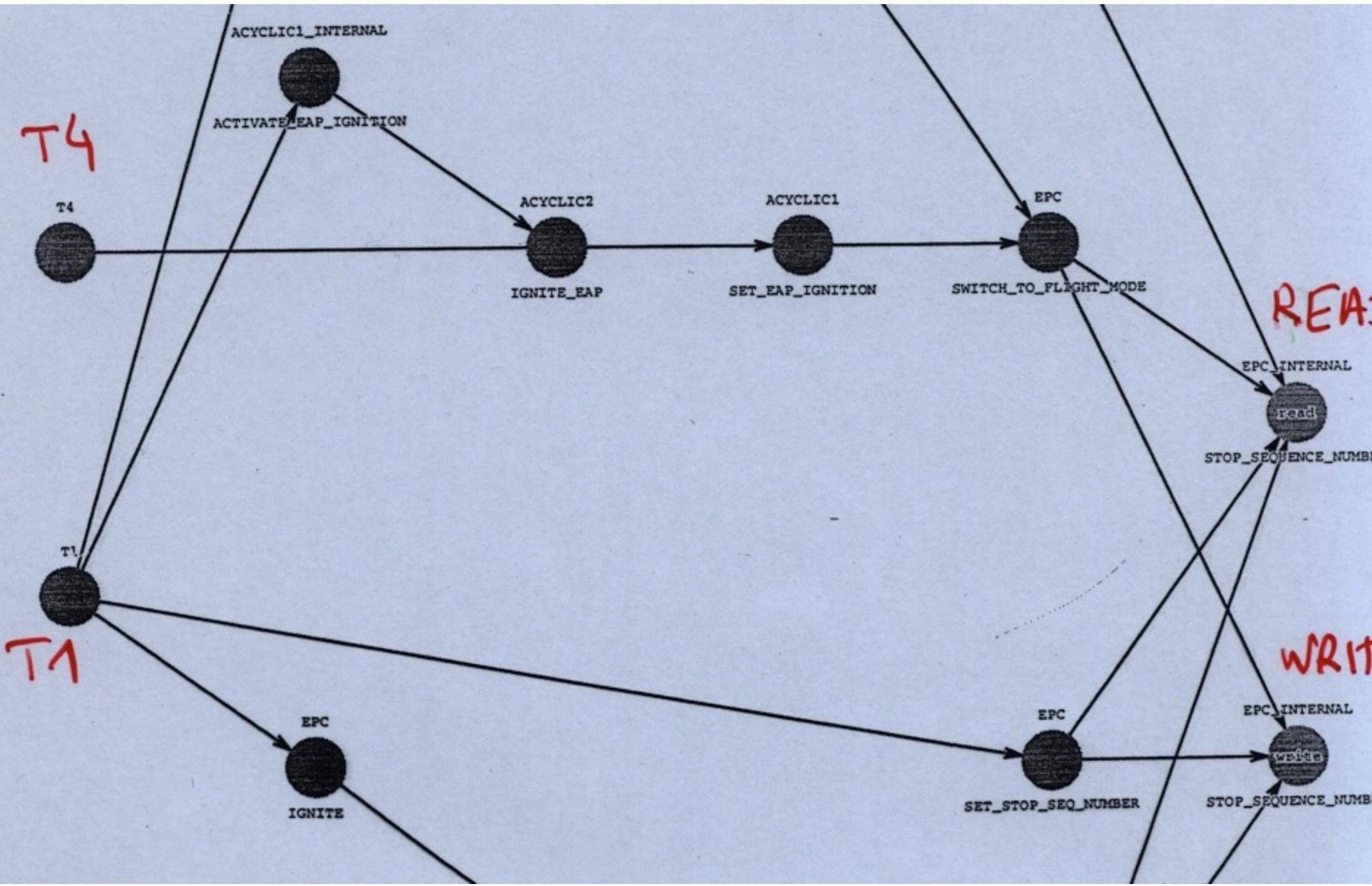
```
function TrouverHoraire() =  
  If Disponibilité (airbusAF318) = "oui" then  
    RépondreVoyageOK ();
```

Accès aux variables partagées

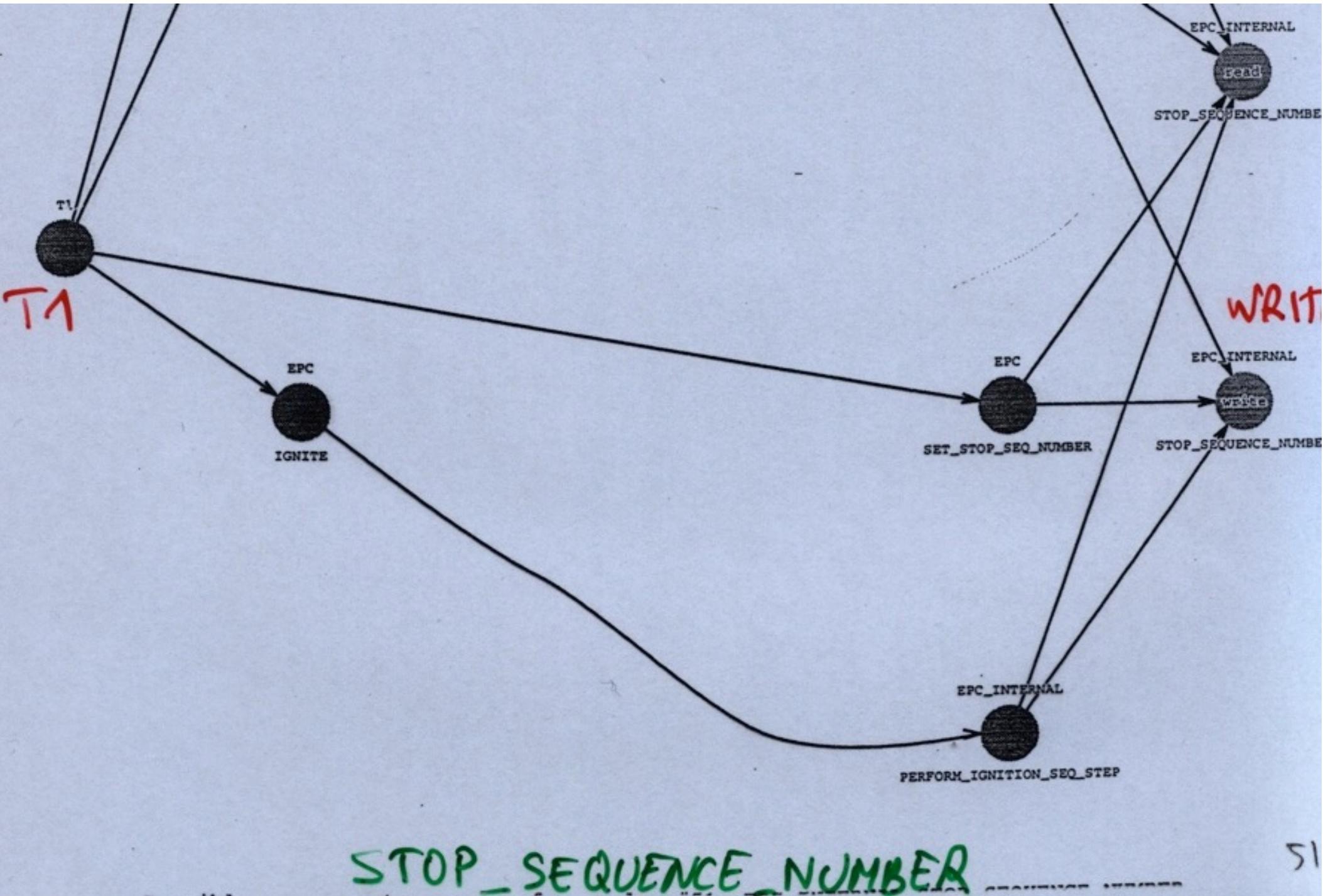




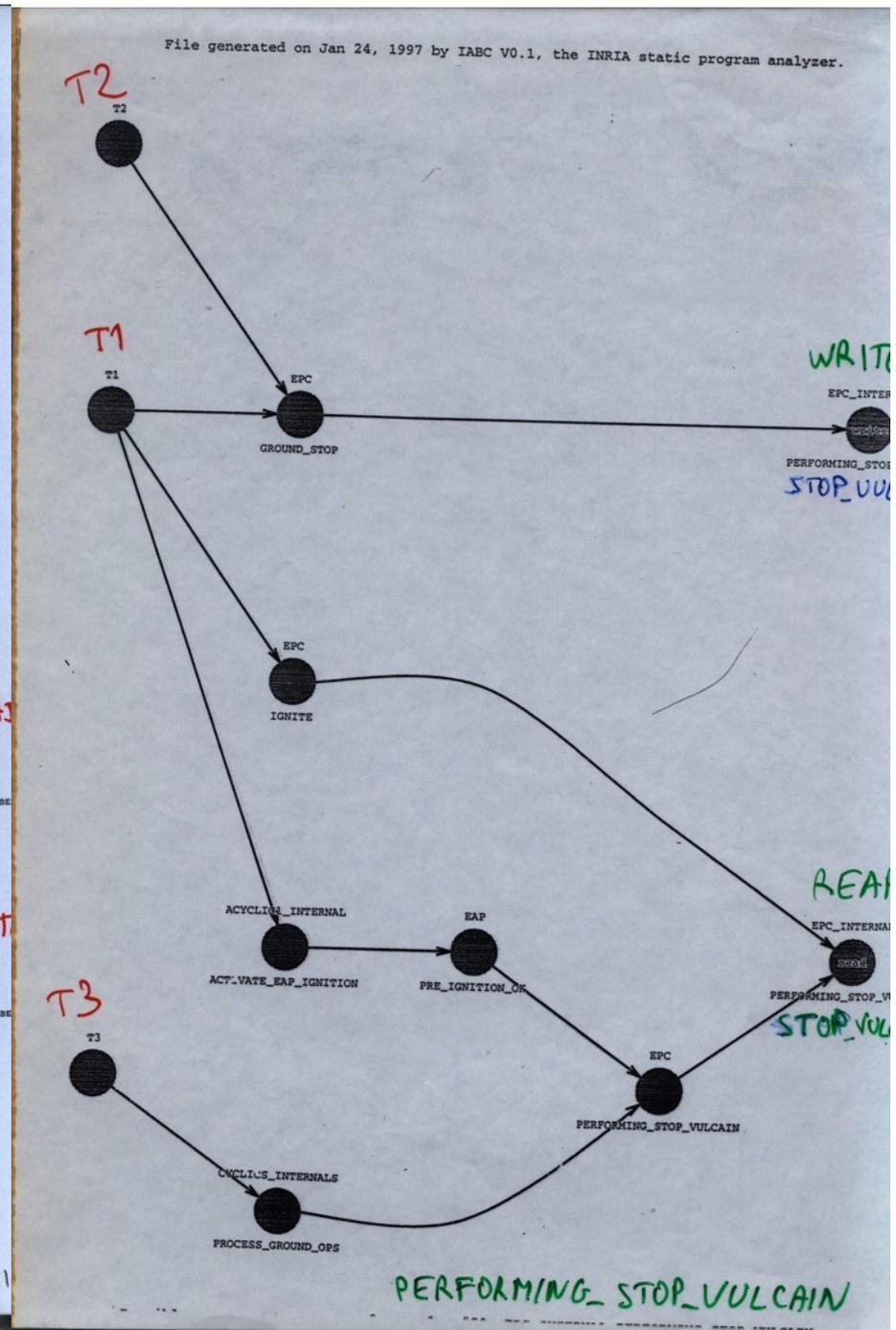
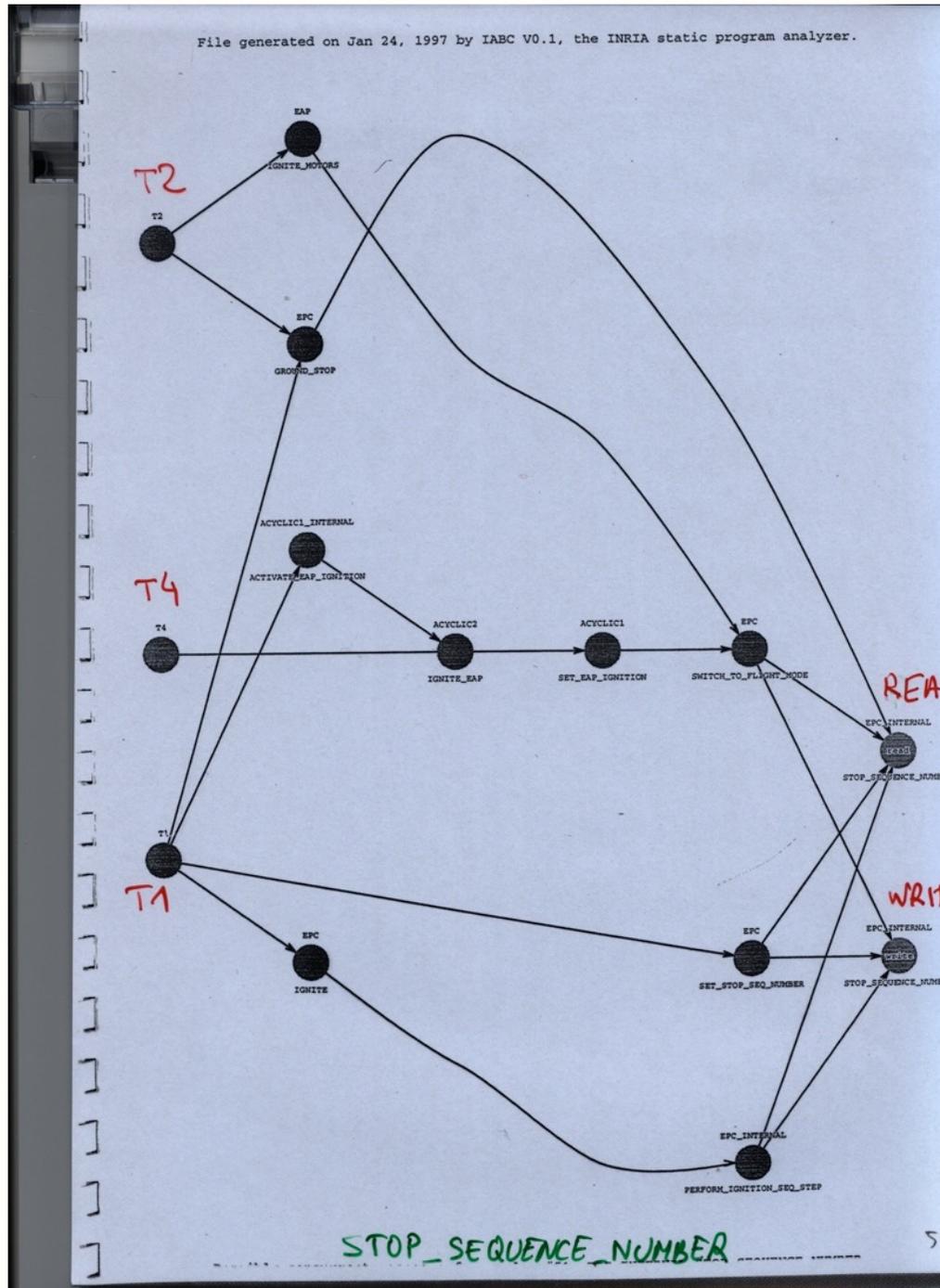
Access to shared variables



Access to shared variables



Access to shared variables

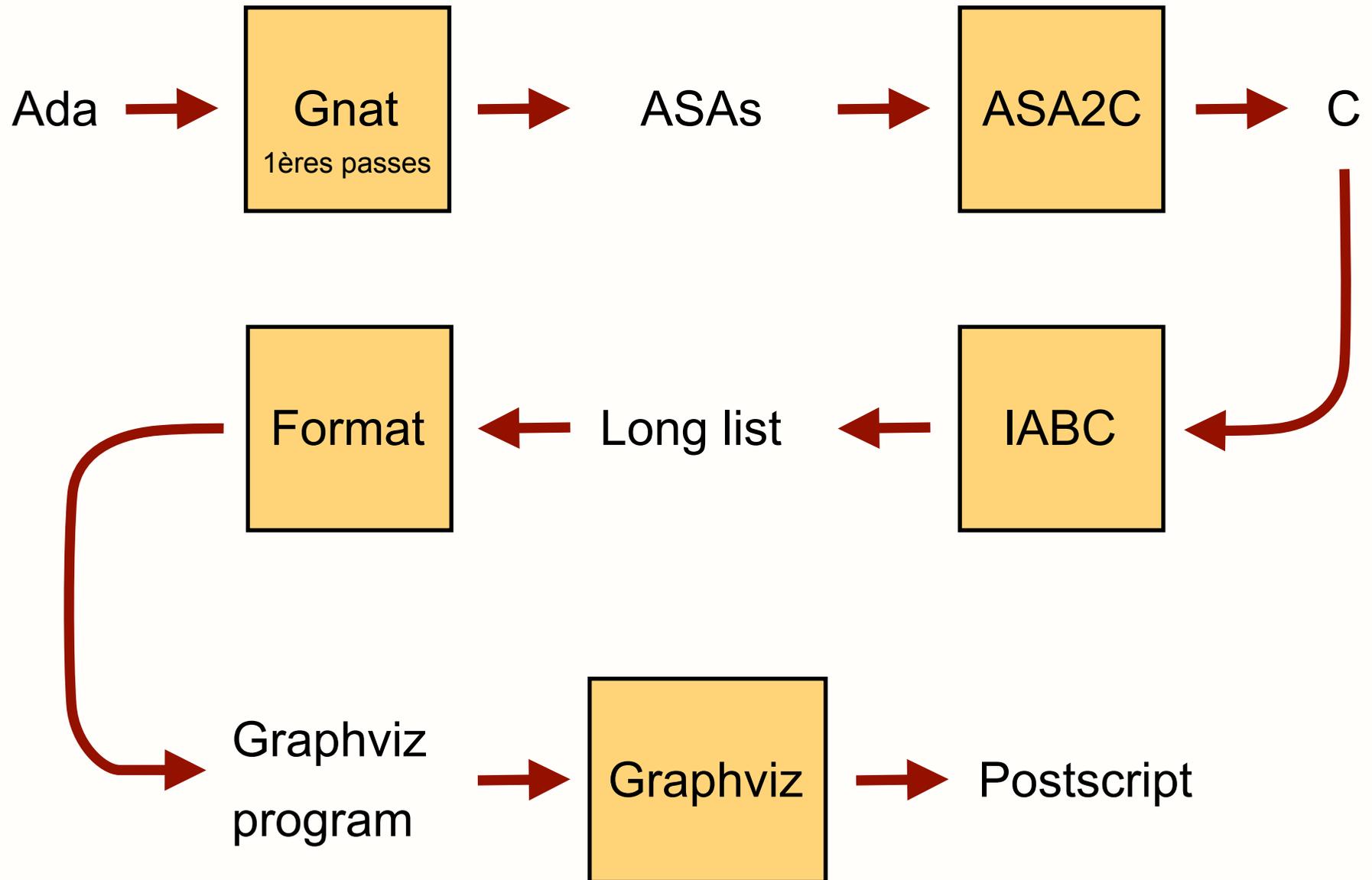


Détection d'alias

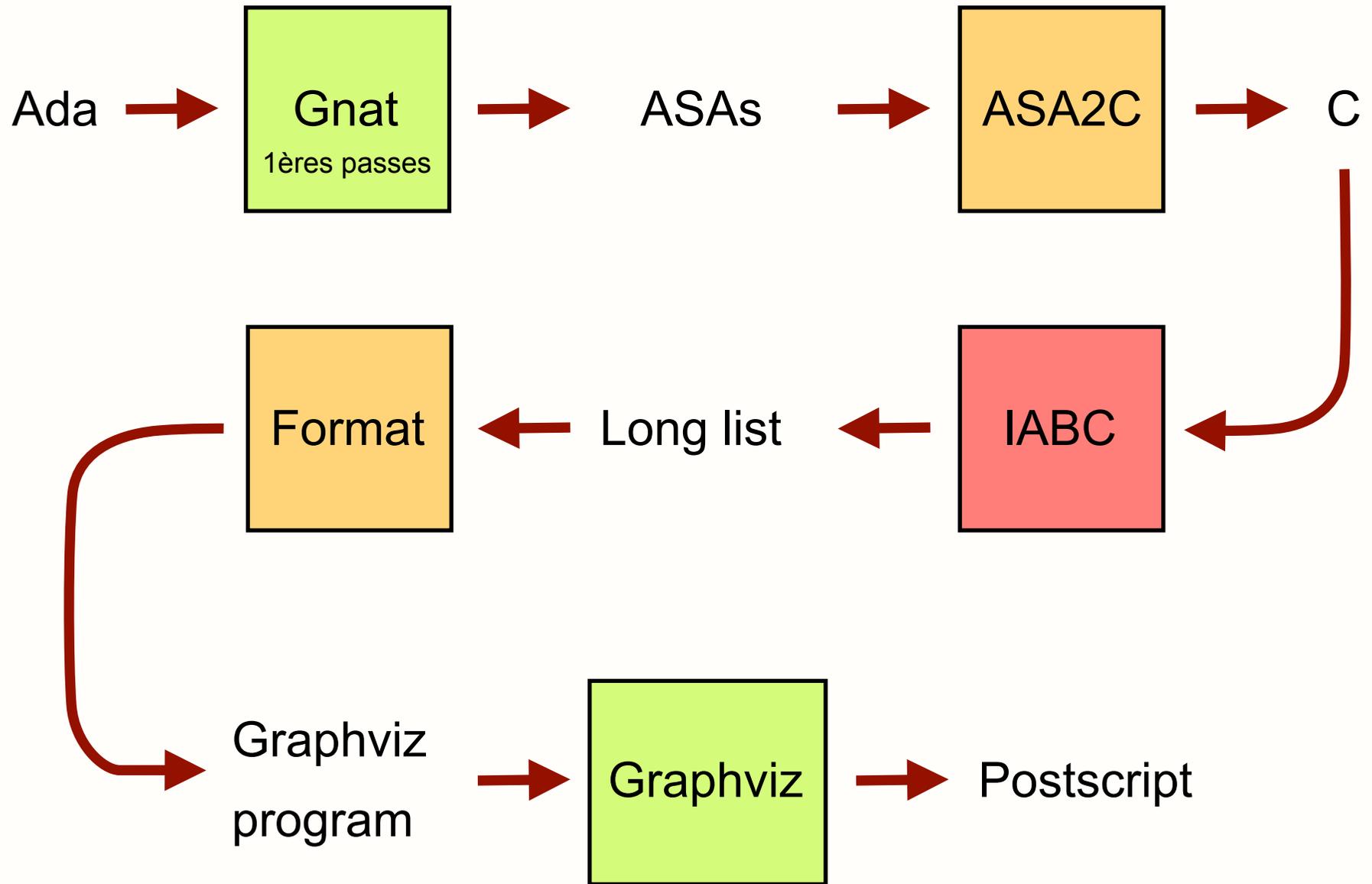
- **travail de 10 ans** d'Alain Deutsch --- IABC (*INRIA Interprocedural Array Bounds Checker*)
- grande communauté de “alias analysis”, “points-to analysis”
- **analyse statique**, *abstract interpretation*
- **approximation** du résultat
- qui garantit l'absence d'alias

- IABC marchait pour des programmes **C**
- programme **efficace** et plutôt **précis**

De Ada à C et IABC



De Ada à C et IABC



Parties en assembleur

- lire la documentation
- et faire des *stubs* ADA

JANVIER 1997

PV

Unité
échange

SRI

```
-- algorithms between phases, must be allowed to execute - even during
-- blackout.
```

```
Task Body CYCLICS_OBCS_TYPE Is
Begin
```

TASK

Début T2

```
accept START;
```

```
-----
--                                GROUND PHASE                                --
-----
```

```
-- Initialise missionised activation times
INITIALISE;
```

```
accept ACTIVATE_PILOT_CYCLE;
-- perform operations for cycle 1
PERFORM_CYCLE_1;
```

```
-- update frame id in UCTM for DM 6
COMMUNICATIONS.PUT_FRAME_ID
  ( FRAME_ID => COMMUNICATIONS_TYPES.GROUND_EAP );
```

```
GROUND_PHASE:
loop
```

```
-- wait for start of a new pilot cycle
accept ACTIVATE_PILOT_CYCLE;
```

```
PROCESS_GROUND_OPS;
```

```
-- Check to see if the phase needs to be changed for
-- the next cycle the PHASE TO CHANGE indicator is
```

```
PERFORM_CYCLE_1;
```

```
-- update frame id in UCTM for DM 6  
COMMUNICATIONS.PUT_FRAME_ID  
  ( FRAME_ID => COMMUNICATIONS_TYPES.GROUND_EAP );
```

```
GROUND_PHASE:
```

```
loop
```

```
-- wait for start of a new pilot cycle  
accept ACTIVATE_PILOT_CYCLE;
```

```
PROCESS_GROUND_OPS;
```

```
-- Check to see if the phase needs to be changed for  
-- the next cycle the PHASE_TO_CHANGE indicator is  
-- updated during the processing of the cycle. The  
-- command to change the frame is called during this  
-- processing.  
exit when PHASE_TO_CHANGE;
```

```
end loop GROUND_PHASE;
```

```
-----  
--  EAP          EAP PHASE  
-----
```

```
-- Set launcher in flight, so that trajectory algorithms are  
-- consistent between critical pilot (EXECUTIVE) and  
-- other trajectory algorithms (CYCLICS).  
PHASE.SET_IN_FLIGHT;
```

```
EAP_FLIGHT_PHASE:
```

```
loop
```

```
-- wait for start of a new pilot cycle
accept ACTIVATE_PILOT_CYCLE;

PROCESS_EAP_OPS;

-- Check to see if the phase needs to be changed for
-- the next cycle the PHASE_TO_CHANGE indicator is
-- updated during the processing of the cycle. The
-- command to change the frame is called during this
-- processing.
exit when PHASE_TO_CHANGE;

end loop EAP_FLIGHT_PHASE;

-----
--      EPC          EPC PHASE          -----
-----

-- 3rd CYCLICS phase (while launcher is in EPC flight )
PHASE.SET(NEW_PHASE => PHASE_TYPES.EPC_WITH_FAIRING);

-- update frame id in UCTM for DM 6
COMMUNICATIONS.PUT_FRAME_ID (FRAME_ID => COMMUNICATIONS_TYPES.EPC);

-- Set up initialisation for roll control
TRAJECTORY.INITIALISE_ACTUATORS;

-- As this is the first call to this function, set up the earliest
-- start time of CALCULATE_CONTROL_INTEGRAL (tleapeff + 10), 139 cycl
-- after start of EPC.
-- CALCULATE_CONTROL_INTEGRAL will be executed in Navigate_cycle_2.
EPC_CONTROL_INTEGRAL_START := OVERALL_CYCLE_COUNT + 139;
```

```

-- update frame id in UCTM for DM 6
COMMUNICATIONS.PUT_FRAME_ID (FRAME_ID => COMMUNICATIONS_TYPES.EPC);

-- Set up initialisation for roll control
TRAJECTORY.INITIALISE_ACTUATORS;

-- As this is the first call to this function, set up the earliest
-- start time of CALCULATE_CONTROL_INTEGRAL (tleapeff + 10), 139 cycl
-- after start of EPC.
-- CALCULATE_CONTROL_INTEGRAL will be executed in Navigate_cycle_2.
EPC_CONTROL_INTEGRAL_START := OVERALL_CYCLE_COUNT + 139;

-- Set the guidance counter so that the out_atmosphere guidance
-- is resynchronised. The ST states that the start of out_atmosphere
-- guidance is started at "tleapeff".
GUIDANCE_CYCLE_COUNT := 1;

EPC_FLIGHT_PHASE:
loop

    -- wait for start of a new pilot cycle
    accept ACTIVATE_PILOT_CYCLE;

    PROCESS_EPC_OPS;

    -- Check to see if the phase needs to be changed for
    -- the next cycle the PHASE_TO_CHANGE indicator is
    -- updated during the processing of the cycle. The
    -- command to change the frame is called during this
    -- processing.
    exit when PHASE_TO_CHANGE;

end loop EPC_FLIGHT_PHASE;

```

EPS

EPS PHASE

```
-- EPC flight is now complete. Further processing will depend on
-- the flight mission.
--
-- There are 2 possible missions;
-- Mission TYPE_1          and          Mission TYPE_2
--
-- EPS flight              no initial EPS flight for this mission
-- SCA (ballistic)        SCA (ballistic)
-- EPS flight              EPS flight
-- SCA (ballistic)        SCA (ballistic)
-- EPS flight              EPS flight
-- SCA FIN                 SCA FIN

-- the mission type is obtained from FLIGHT_MISSION_DATA.
if FLIGHT_MISSION_DATA.MISSION_TYPE_IS = FLIGHT_TYPES.TYPE_1 then

    PHASE.SET(NEW_PHASE => PHASE_TYPES.EPS_STOPPED);

    -- update frame id in UCTM for DM 6
    COMMUNICATIONS.PUT_FRAME_ID (FRAME_ID => COMMUNICATIONS_TYPES.EPS);

    -- Reset digital message 1 and 2 with thrust estimation data.
    COMMUNICATIONS.PUT_THRUST_ESTIMATE_MESSAGE (MESSAGE => ( others => .0.0)

    TRAJECTORY.INITIALISE_EPS_PILOT;

    TRAJECTORY.INITIALISE_ACTUATORS;
    -- for a type 1, actuators initialisation for roll control
    -- is activated once, on the first EPS phase.

    EPS_FLIGHT_PHASE:
    loop
```

```
-- update frame id in UCTM for DM 6
COMMUNICATIONS.PUT_FRAME_ID (FRAME_ID => COMMUNICATIONS_TYPES.EPS);

-- Reset digital message 1 and 2 with thrust estimation data.
COMMUNICATIONS.PUT_THRUST_ESTIMATE_MESSAGE (MESSAGE => ( others => .0.0)

TRAJECTORY.INITIALISE_EPS_PILOT;

TRAJECTORY.INITIALISE_ACTUATORS;
-- for a type 1, actuators initialisation for roll control
-- is activated once, on the first EPS phase.

EPS_FLIGHT_PHASE:
loop

    -- wait for start of a new pilot cycle
    accept ACTIVATE_PILOT_CYCLE;

    PROCESS_EPS_OPS;

    -- Check to see if the phase needs to be changed for
    -- the next cycle the PHASE_TO_CHANGE indicator is
    -- updated during the processing of the cycle. The
    -- command to change the frame is called during this
    -- processing.
    exit when PHASE_TO_CHANGE;

end loop EPS_FLIGHT_PHASE;

end if; -- FLIGHT_MISSION_DATA.MISSION_TYPE_IS
```

```
-- Ballistype SCA PHASE --
```

97/01/24
12:25:07

SHARED-SCALAR-DATA-TABLE-SUMMARY.txt

1

File generated on Jan 24, 1997 by IABC V0.1, the INRIA static program analyzer.

JTL

Level 0 Concurrency Analysis for scalars accessed by:

T1) ACYCLIC1_INTERNAL.OBCS_TYPE (task), "acycl1c1_internal_ada" line 63, column 4:
Task type OBCS_TYPE Is
^

T2) ACYCLIC2_INTERNAL.OBCS_TYPE (task), "acycl2c2_internal_ada" line 69, column 4:
Task type OBCS_TYPE Is
^

T3) CYCLICS_INTERNALS.CYCLICS_OBCS_TYPE (task), "cyclics_internal_ada" line 194, column 2:
Task type CYCLICS_OBCS_TYPE Is
^

T4) EXECUTIVE_INTERNAL.OBCS_TYPE (task), "executive_internal_ada" line 50, column 3: *man*
task type OBCS_TYPE is
^

T5) GUIDANCE_CONTROL_INTERNAL.OBCS_TYPE (task), "guidance_control_internal_ada" line 51, column 0:
Task type OBCS_TYPE Is
^

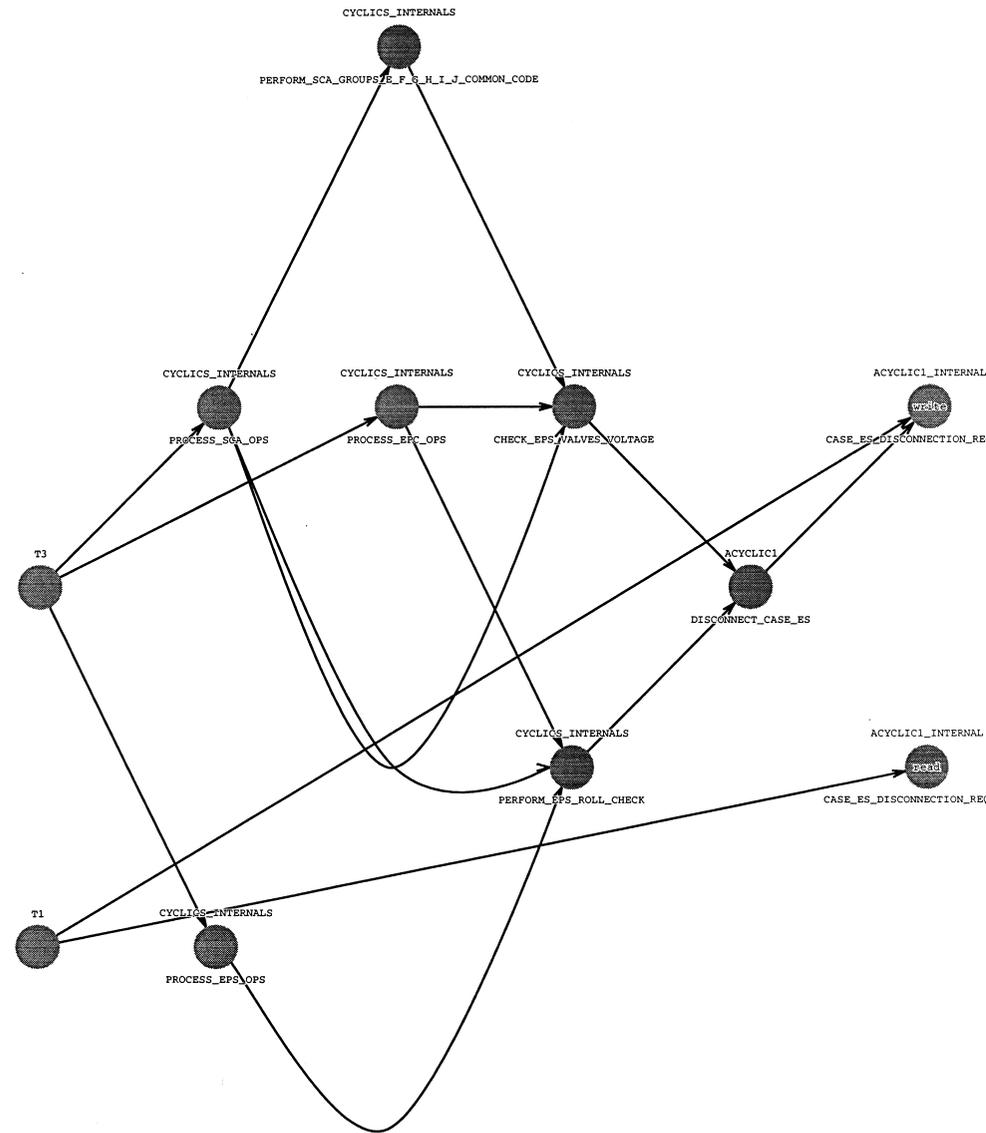
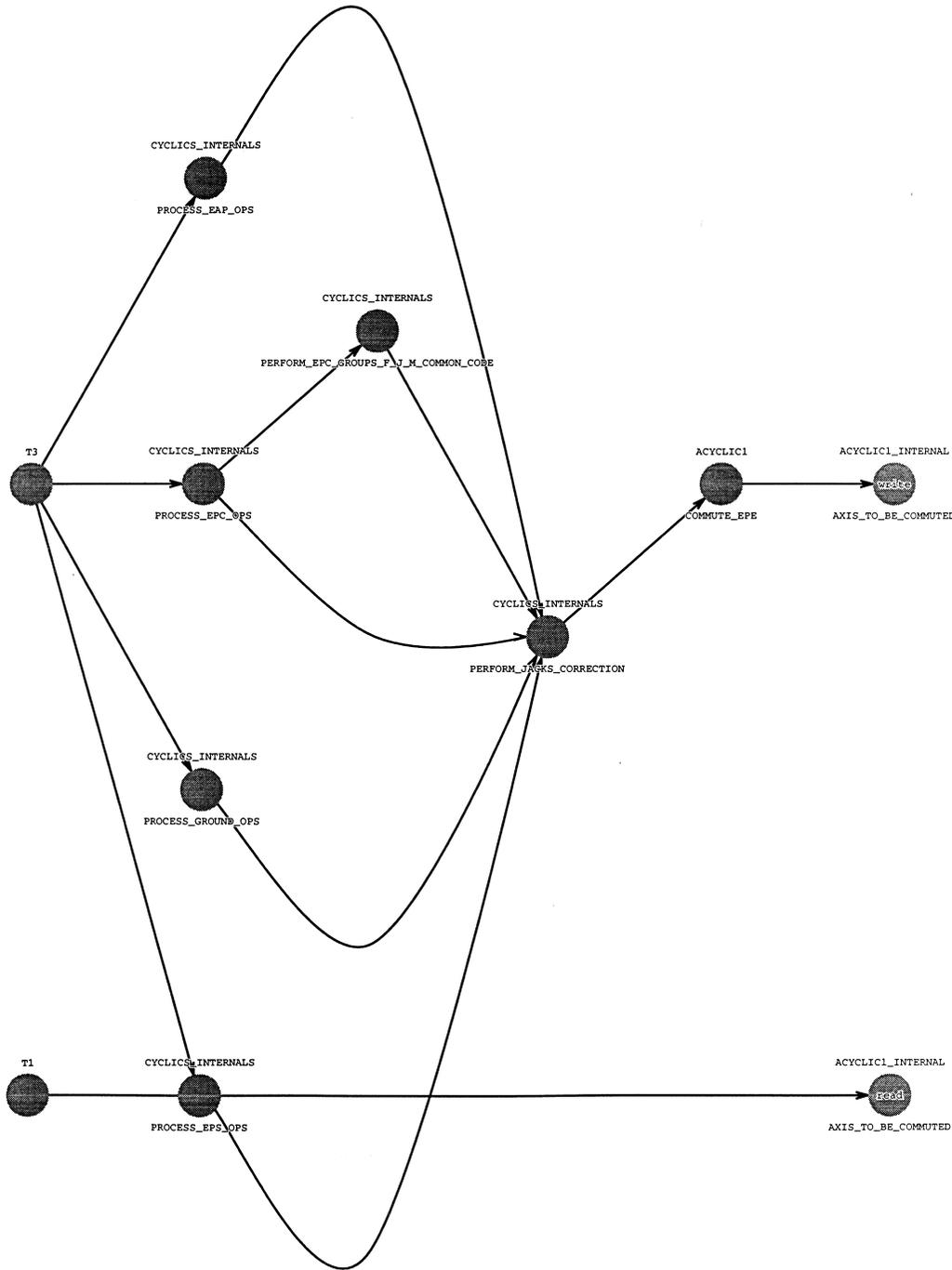
	variable	type	reads	writes
> 1	ACTUATORS_INTERNAL.PAYLOAD_NBR	-32768..32767	{T2}	{T3}
> 2	ACYCLIC1_INTERNAL.ACYCLIC1_TASK_ID	1..2147483647	{T3}	{T1}
> 3	ACYCLIC1_INTERNAL.AXIS_TO_BE_COMMUTED	u_axis..v_axis	{T1}	{T3}
> 4	ACYCLIC1_INTERNAL.CASE_ES_DISCO...N_REQUESTED	false..true	{T1}	{T1, T3}
> 5	ACYCLIC1_INTERNAL.COMMUTE_EPE_REQUESTED	false..true	{T1}	{T1, T3}
> 6	ACYCLIC1_INTERNAL.EAPS_IGNITED	false..true	{T1, T3}	{T1, T4}
> 7	ACYCLIC1_INTERNAL.EPC_ES_DISCON...N_REQUESTED	false..true	{T1}	{T1, T3}
> 8	ACYCLIC1_INTERNAL.ES_TO_BE_DISCONNECTED	es1..es2	{T1}	{T3}
> 9	ACYCLIC1_INTERNAL.FRAME_STARTED_IN_OBC2	false..true	{T3}	{T1}
> 10	ACYCLIC2_EVENT_QUEUE.POP_POINTER	event_1..event_8	{T2}	{T1, T2}
> 11	ACYCLIC2_EVENT_QUEUE.PUSH_POINTER	event_1..event_8	{T1}	{T3}
> 12	ACYCLIC2_INTERNAL.DISCONNECT_ES	false..true	{T2}	{T3}
> 13	ACYCLIC2_INTERNAL.ES_TO_BE_DISCONNECTED	es1..es2	{T2}	{T3}
14	BGY_INTERNAL.DIAL_BGY1	-32768..32767	{T3, T4}	{T3, T4}
15	BGY_INTERNAL.DIAL_BGY2	-32768..32767	{T3, T4}	{T3, T4}
16	BGY_INTERNAL.FIRST_PASS	false..true	{T3, T4}	{T3, T4}
17	BGY_INTERNAL.ILC1	-32768..32767	{T3, T4}	{T3, T4}
18	BGY_INTERNAL.ILC2	-32768..32767	{T3, T4}	{T3, T4}
19	BGY_INTERNAL.ITG1	-32768..32767	{T3, T4}	{T3, T4}
20	BGY_INTERNAL.ITG2	-32768..32767	{T3, T4}	{T3, T4}
21	BGY_INTERNAL.MASIL1	no_failure..definitive_failure	{T3, T4}	{T3, T4}
22	BGY_INTERNAL.MASIL2	no_failure..definitive_failure	{T3, T4}	{T3, T4}
23	BGY_INTERNAL.MASIT1	no_failure..definitive_failure	{T3, T4}	{T3, T4}
24	BGY_INTERNAL.MASIT2	no_failure..definitive_failure	{T3, T4}	{T3, T4}

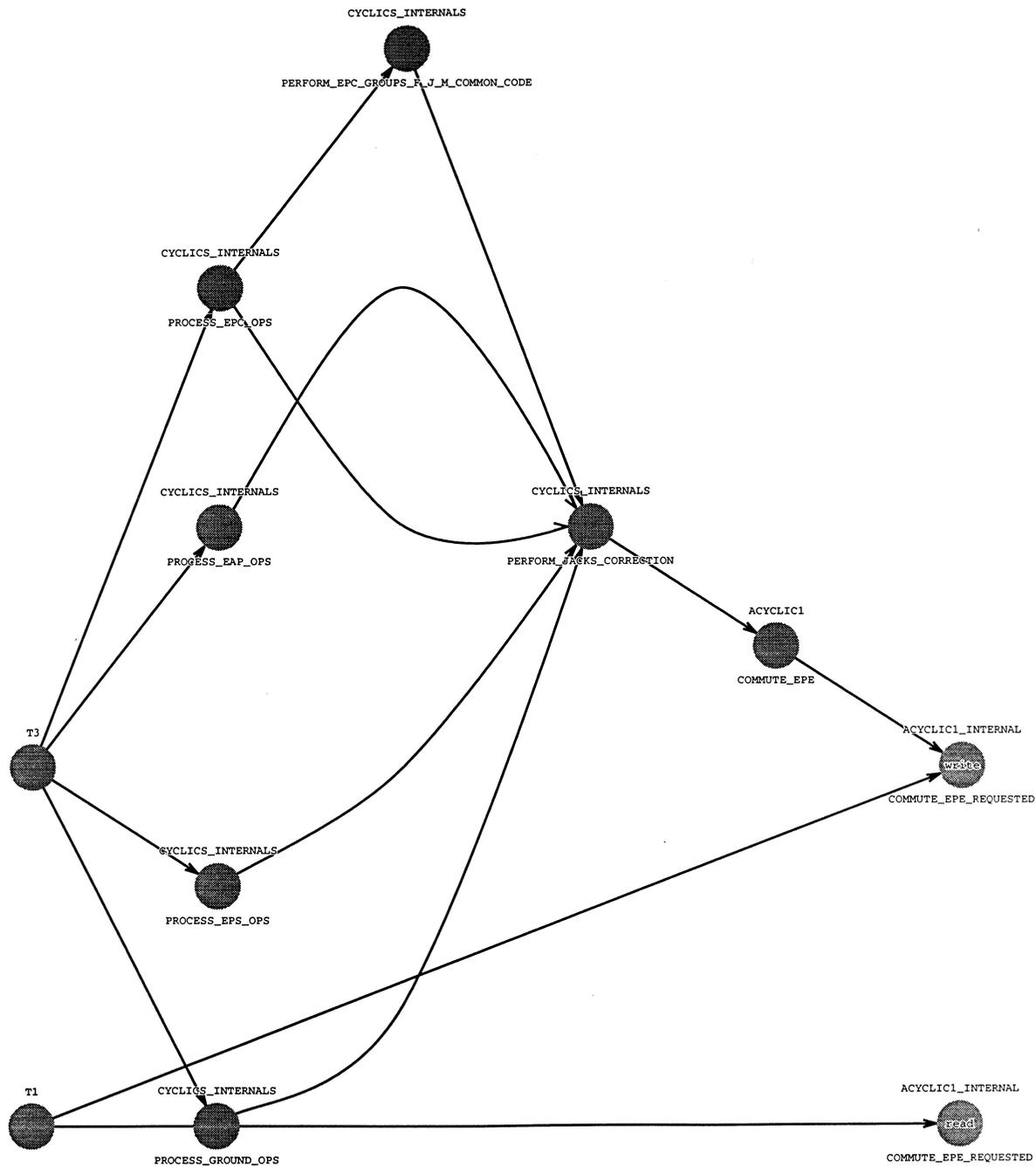
SHARED-SCALAR-DATA-TABLE-SUMMARY.txt

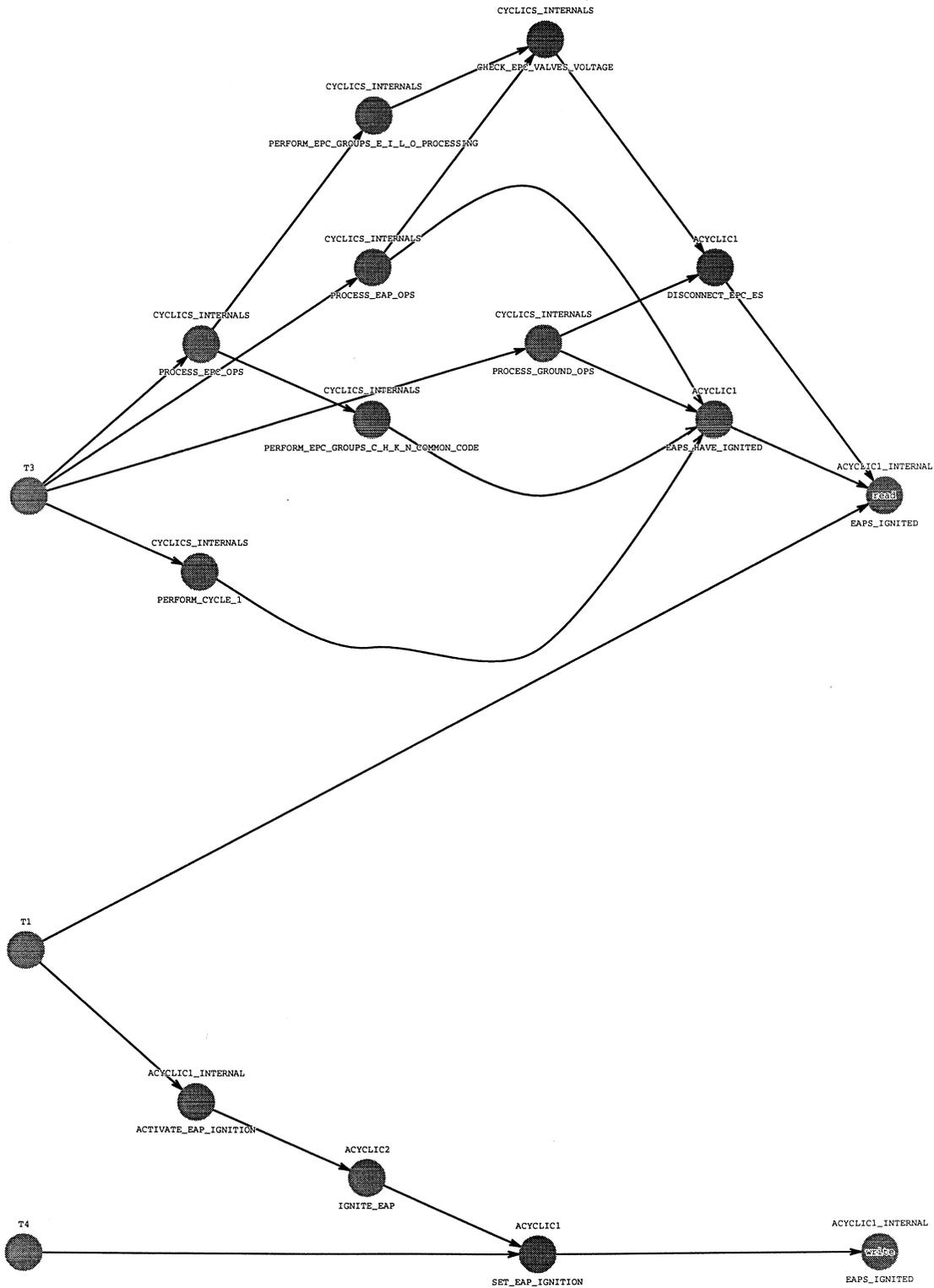
25	BGY_INTERNAL.NTEN1	-32768..32767	{T3,T4}	{T3,T4}
26	BGY_INTERNAL.NTEN2	-32768..32767	{T3,T4}	{T3,T4}
27	BGY_INTERNAL.PGL1	-32768..32767	{T3,T4}	{T3,T4}
28	BGY_INTERNAL.PGL2	-32768..32767	{T3,T4}	{T3,T4}
29	BGY_INTERNAL.PGT1	-32768..32767	{T3,T4}	{T3,T4}
30	BGY_INTERNAL.PGT2	-32768..32767	{T3,T4}	{T3,T4}
31	CYCLICS_INTERNALS.DOUBLE_FAILURE	false..true	{T3,T4}	{T3,T4}
32	CYCLICS_INTERNALS.EPS_PRESSURISE_FIRST_CALL	false..true	{T2}	{T3}
33	CYCLICS_INTERNALS.IN_EAP_TO_EPC_TRANSITION	false..true	{T4}	{T3}
34	CYCLICS_INTERNALS.INHIBIT_TRAJECTORY	false..true	{T3,T4}	{T3,T4}
35	CYCLICS_INTERNALS.OVERALL_CYCLE_COUNT	1..2147483647	{T4}	{T3}
36	CYCLICS_INTERNALS.SRI_BGY_MEASUREMENT_FAILURE	false..true	{T3,T4}	{T3,T4}
37	CYCLICS_INTERNALS.SRI_NEVER_COMMUTED	false..true	{T3,T4}	{T3,T4}
38	DATE.EPC_STOP_MOTOR_REASON	no_stop..low_level	{T3}	{T3,T5}
39	DATE.EPS_IGNITION_COUNTER	0..3	{T2,T5}	{T3}
40	DATE.EPS_STOP_MOTOR_REASON	no_extinction..date	{T3}	{T3,T5}
41	DATE_PAYLOAD_SETUP.ACTIVE_POINTER	0..22	{T2}	{T3}
42	ELECTRICAL_SYSTEM_INTERNAL.EPE_EPS_COMMUTE_OK	false..true	{T1,T2,T3}	{T1,T2,T3}
43	ELECTRICAL_SYSTEM_INTERNAL.EPH..._COMMUTE_OK	false..true	{T1,T2,T3}	{T1,T2,T3}
44	ELECTRICAL_SYSTEM_INTERNAL.EPH..._COMMUTE_OK	false..true	{T1,T2,T3}	{T1,T2,T3}
45	ELECTRICAL_SYSTEM_INTERNAL.EPH_EPC_COMMUTE_OK	false..true	{T1,T2,T3}	{T1,T2,T3}
46	ELECTRICAL_SYSTEM_INTERNAL.ES_C...SCONNECT_OK	false..true	{T1,T2,T3}	{T1,T2,T3}
47	ELECTRICAL_SYSTEM_INTERNAL.ES_E...SCONNECT_OK	false..true	{T1,T2}	{T1,T2}
48	EPC_INTERNAL.EPH_DISCONNECTED	false..true	{T3}	{T2}
49	EPC_INTERNAL.INHIBIT_DETECT_CORRECT	false..true	{T3}	{T1}
50	EPC_INTERNAL.PERFORMING_STOP_VULCAIN	false..true	{T1,T3}	{T1,T2}
51	EPC_INTERNAL.STOP_SEQUENCE_NUMBER	stop_sequence_0..stop...uence_3	{T1,T2,T4}	{T1,T2,T4}
52	EPC_TANKS_INTERNAL.CONTINUE_EPC...CYCLIC_PROC	false..true	{T3}	{T1,T2}
53	EQUIP.I_AM_BC	false..true	{T1,T2,T3,T4}	{T1,T2,T4}
54	EQUIP.THE_SRI1_IS_NOMINAL	false..true	{T1,T2,T3,T4}	{T1,T3,T4}
55	EV.CAN_SWITCH_OBC	false..true	{T1,T2,T3}	{T1,T4}
56	GUIDANCE_INTERNAL.BLOC	false..true	{T3,T5}	{T3,T5}
57	GUIDANCE_INTERNAL.DTHA_FLAG	of_16_cycles..of_64_cycles	{T3,T5}	{T3,T5}
58	GUIDANCE_INTERNAL.ISEG_FLT	-32768..32767	{T3,T5}	{T3,T5}
59	GUIDANCE_INTERNAL.KBAL	false..true	{T3}	{T5}
60	GUIDANCE_INTERNAL.KD	-32768..32767	{T3}	{T5}
61	GUIDANCE_INTERNAL.KPANNE	false..true	{T3,T5}	{T3,T5}
62	GUIDANCE_INTERNAL.NBSEG	-32768..32767	{T3,T5}	{T3,T5}
63	GUIDANCE_INTERNAL.NCOIF	-32768..32767	{T3,T5}	{T3,T5}
64	GUIDANCE_INTERNAL.NCOM	-32768..32767	{T5}	{T3,T5}
65	GUIDANCE_INTERNAL.NFIN	-32768..32767	{T3,T5}	{T3,T5}
66	GUIDANCE_INTERNAL.NSEP	-32768..32767	{T3,T5}	{T3,T5}
67	GUIDANCE_INTERNAL.NUMBAL	-32768..32767	{T3,T5}	{T3,T5}
68	OBC_INTERNAL.H0R	-2147483648..2147483647	{T3,T5}	{T4}
69	OBC_INTERNAL.INHIBIT_DASDC_MESSAGE	false..true	{T1,T3}	{T1,T2,T4}

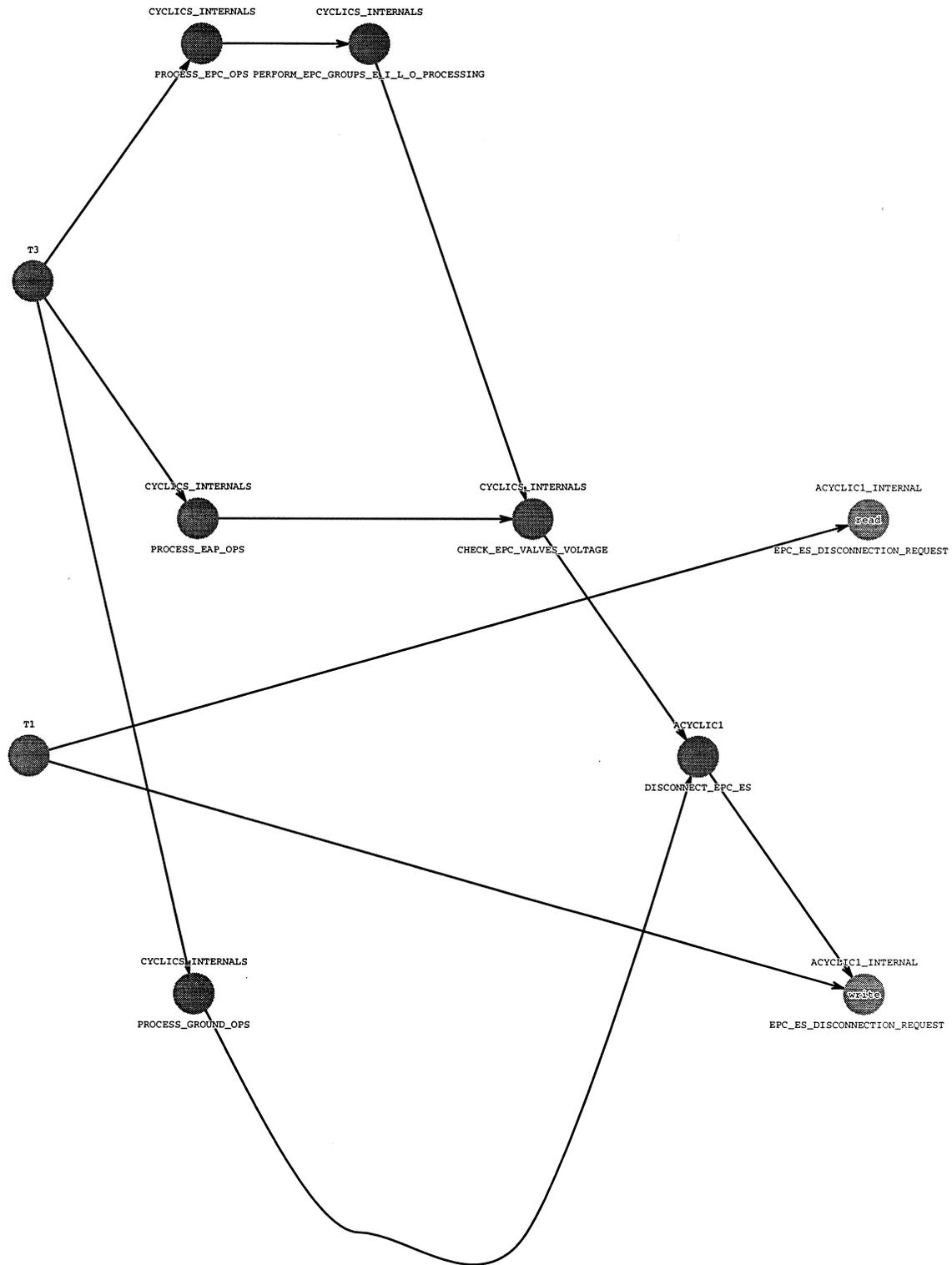
SHARED-SCALAR-DATA-TABLE-SUMMARY.txt

✕ 70	OBC_INTERNAL.MONITOR_FIRST_PASS	false..true	{T1, T3}	{T1, T3}
✕ 71	OBC_INTERNAL.PHASE1_TRANSITION_INHIBITED	false..true	{T4}	{T1, T2, T4}
✕ 72	PHASE_INTERNAL.CURRENT_PHASE	undefined..eps_gdp	{T1, T2, T3, T4}	{T1, T2, T3, T4}
✕ 73	PHASE_INTERNAL.ON_GROUND	false..true	{T1, T2, T4}	{T3}
74	PILOT_INTERNAL.FIXED_BRAQUAGE_REQUESTED	false..true	{T3, T4}	{T3, T4}
75	PILOT_INTERNAL.IEPX	-32768..32767	{T4}	{T3}
76	PILOT_INTERNAL.NFIEPX	-32768..32767	{T4}	{T3}
77	PILOT_INTERNAL.NRAZ	-32768..32767	{T3, T4}	{T3, T4}
✕ 78	SIMU_DATA.NEXT_INPUT_NB	1..32	{T1, T2, T3, T4}	{T1, T2, T3, T4}
79	SRI_INTERNAL.DIAL_AMONT_SRIN	-32768..32767	{T3, T4}	{T3, T4}
80	SRI_INTERNAL.DIAL_AMONT_SRIS	-32768..32767	{T3, T4}	{T3, T4}
81	SRI_INTERNAL.IALAT	-32768..32767	{T3, T4}	{T3, T4}
82	SRI_INTERNAL.IHACC	false..true	{T4}	{T3}
83	SRI_INTERNAL.IHATT	false..true	{T4}	{T3}
84	SRI_INTERNAL.LGCFIL	false..true	{T3, T4}	{T3, T4}
85	SRI_INTERNAL.NOCYN	-2147483648..2147483647	{T3, T4}	{T3, T4}
86	SRI_INTERNAL.NOCYNP	-2147483648..2147483647	{T3, T4}	{T3, T4}
87	SRI_INTERNAL.NOCYS	-2147483648..2147483647	{T3, T4}	{T3, T4}
88	SRI_INTERNAL.NOCYSP	-2147483648..2147483647	{T3, T4}	{T3, T4}
89	SRI_INTERNAL.SRI_VALIDITY	both_ok..backup_only	{T3, T4}	{T3, T4}
90	SRI_INTERNAL.UPSTREAM_FIRST_PASS	false..true	{T3, T4}	{T3, T4}
✕ 91	TIMER_INTERNAL.CURRENT_NAVIGATION_CYCLE	cycle_1..cycle_8	{T1, T3}	{T1, T3}
92	TIMER_INTERNAL.PHASE_CYCLE_COUNTER	0..32767	{T1, T4}	{T1, T4}
→ 93	TIMER_INTERNAL.PREDICTED_AWOKEN_DATE	-2147483648..2147483647	{T3}	{T1, T2, T3, T4}
✕ 94	TIMER_INTERNAL.SIXTEEN_CYCLE_COUNT	cycle_1..cycle_16	{T1, T3}	{T1, T3}
95	UCTM_INTERNAL.ACTUAL_H0	-2147483648..2147483647	{T1, T2, T3}	{T4}
→ 96	UCTM_INTERNAL.APP_CR_BC_ID_INDEX	1..2	{T1, T2, T3, T4}	{T1, T2, T3, T4}
→ 97	UCTM_INTERNAL.CURRENT_READ_POINTER_EVENT	1..8	{T1, T2, T3, T4}	{T1, T2, T3, T4}
✕ 98	UCTM_INTERNAL.CURRENT_WRITE_POINTER_EVENT	1..8	{T1, T2, T3}	{T1, T2, T3}
99	UCTM_INTERNAL.KD_DM	1..25	{T3}	{T3, T5}
100	UCTM_INTERNAL.MSG_6_BC_ID_INDEX	1..3	{T1, T2, T3, T4}	{T1, T2, T3, T4}
✕ 101	UCTM_INTERNAL.OBC_STATE	initialise..applicat...bc_mode	{T1, T2, T3, T4}	{T1, T2, T4}
→ 102	UCTM_INTERNAL.TELE_IS_STOPPED	false..true	{T3}	{T1, T2, T4}









JSL

File generated on Jan 24, 1997 by IABC V0.1, the INRIA static program analyzer.

Level 0 Concurrency Analysis for non scalars accessed by:

- T1) ACYCLIC1_INTERNAL.OBCS_TYPE (task), "acyclic1_internal_.ada" line 63, column 4:
Task type OBCS_TYPE Is
^
- T2) ACYCLIC2_INTERNAL.OBCS_TYPE (task), "acyclic2_internal_.ada" line 69, column 4:
Task type OBCS_TYPE Is
^
- T3) CYCLICS_INTERNALS.CYCLICS_OBCS_TYPE (task), "cyclics_internal_.ada" line 194, column 2:
Task type CYCLICS_OBCS_TYPE Is
^
- T4) EXECUTIVE_INTERNAL.OBCS_TYPE (task), "executive_internal_.ada" line 50, column 3:
task type OBCS_TYPE is
^
- T5) GUIDANCE_CONTROL_INTERNAL.OBCS_TYPE (task), "guidance_control_internal_.ada" line 51, column 0:
Task type OBCS_TYPE Is
^

	variable	type	reads	writes
1	<ASM-GLOBALS>		{}	{T1, T2, T3, T4}
2	ACYCLIC2_EVENT_QUEUE.QUEUE <i>sem 92</i>	array(event_1..event...record)	{T1, T2}	{T1, T2, T3}
3	ATTITUDE_INTERNAL.GHA_COMMANDS	#<record>	{T3}	{T3, T5}
4	ATTITUDE_INTERNAL.PSID	float	{T4}	{T3}
5	ATTITUDE_INTERNAL.TETAD	float	{T4}	{T3}
6	ATTITUDE_INTERNAL.VPSID	float	{T4}	{T3}
7	ATTITUDE_INTERNAL.VTETAD	float	{T4}	{T3}
8	BGY_INTERNAL.MEAN_VELOCITIES	#<record>	{T3, T4}	{T3, T4}
9	BGY_INTERNAL.XPIL	#<record>	{T4}	{T3}
10	CYCLICS_INTERNALS.TIME_AUTOPILO...CLE_STARTED	float	{T3}	{T4}
11	DATA_TABLES_INTERNAL.A6L	float	{T4}	{T3}
12	DATA_TABLES_INTERNAL.A6T	float	{T4}	{T3}
13	DATA_TABLES_INTERNAL.BF_0	#<record>	{T4}	{T3}
14	DATA_TABLES_INTERNAL.BLCOMP	float	{T4}	{T3}
15	DATA_TABLES_INTERNAL.BMAX	array(1..3) of float	{T4}	{T3}
16	DATA_TABLES_INTERNAL.BPF_0	#<record>	{T4}	{T3}
17	DATA_TABLES_INTERNAL.GAINC	array(1..29) of float	{T4}	{T3}
18	DATA_TABLES_INTERNAL.GAING	array(1..29) of float	{T4}	{T3}
19	DATA_TABLES_INTERNAL.GAINK	array(1..29) of float	{T4}	{T3}
20	DATA_TABLES_INTERNAL.GAINST	array(1..3) of float	{T4}	{T3}
21	DATA_TABLES_INTERNAL.GICL	float	{T4}	{T3}
22	DATA_TABLES_INTERNAL.GICT	float	{T4}	{T3}
23	DATA_TABLES_INTERNAL.KL	array(1..3) of float	{T4}	{T3}
24	DATA_TABLES_INTERNAL.KT	array(1..3) of float	{T4}	{T3}

25	DATA_TABLES_INTERNAL.PF_1_2	#<record>	{T4}	{T3}
26	DATA_TABLES_INTERNAL.PHID	float	{T4}	{T3}
27	DATA_TABLES_INTERNAL.VBMAX	array(1..3) of float	{T4}	{T3}
28	DATA_TABLES_INTERNAL.VPHID	float	{T4}	{T3}
29	DATA_TABLES_INTERNAL.XBVERU	float	{T4}	{T3}
30	DATA_TABLES_INTERNAL.XBVERV	float	{T4}	{T3}
31	DATE_DESCRIPTOR_TABLE	array(eap_start..epc...record>	{T1,T2,T3,T4,T5}	{T2,T3,T4,T5}
32	DATE_PAYLOAD_SETUP.LIST	array(1..22) of #<record>	{T2}	{T3}
33	EPC_INTERNAL.AT_HR1	#<record>	{T1,T2}	{T1,T2}
34	EPC_INTERNAL.AT_HR2	#<record>	{T1,T2}	{T1,T2}
35	EPC_INTERNAL.STEP_51_DATE	#<record>	{T1,T2}	{T1,T2}
36	EPC_INTERNAL.STOP_SEQ1_DATE	#<record>	{T1,T2}	{T1,T2}
37	EPC_INTERNAL.STOP_SEQ2_DATE	#<record>	{T1,T2}	{T1,T2}
38	EPC_INTERNAL.STOP_SUB_SEQ_DATE	#<record>	{T1,T2}	{T1,T2}
39	EPC_TANKS_INTERNAL.LOX_LH2_HE_VALVE_STATE	array(evppo1..evpe2)...record>	{T1,T3}	{T1,T3}
40	EQUIP.FUNCTIONAL_STATES	array(eq_null..eq_kb...service	{T1,T2,T3}	{T1,T2,T3,T4}
41	EQUIP.SEL_STATUS_WORD	#<record>	{T1,T2,T3,T4}	{T1,T2,T3,T4}
42	ES.AXIS_NOT_FAILED_INT	array(axis_u..axis_v...e..true	{T2}	{T1,T2,T3}
43	EV.STATE_CMD	array(group_epc_1..g...e..true	{T1,T2,T3}	{T1,T2,T3}
44	EV.STATE_CMD_2	array(group_epc_1..g...e..true	{T3}	{T1,T2,T3}
45	EV.STATE_COMMUTE	array(group_epc_1..g...e..true	{T3}	{T1,T2,T3}
46	GUIDANCE_INTERNAL.C_M	array(1..5, 1..5) of float	{T5}	{T3,T5}
47	GUIDANCE_INTERNAL.COEF	array(1..2) of float	{T3}	{T5}
48	GUIDANCE_INTERNAL.COMA	array(1..3) of float	{T3,T5}	{T3,T5}
49	GUIDANCE_INTERNAL.COMA2	array(1..3) of float	{T3,T5}	{T3,T5}
50	GUIDANCE_INTERNAL.COMABC	array(1..3) of float	{T3,T5}	{T3,T5}
51	GUIDANCE_INTERNAL.COMB	array(1..3) of float	{T3,T5}	{T3,T5}
52	GUIDANCE_INTERNAL.COMB2	array(1..3) of float	{T3,T5}	{T3,T5}
53	GUIDANCE_INTERNAL.COMBBC	array(1..3) of float	{T3,T5}	{T3,T5}
54	GUIDANCE_INTERNAL.COMMANDS	array(1..5) of array...f float	{T3}	{T5}
55	GUIDANCE_INTERNAL.DELTA1	array(1..35) of float	{T5}	{T3,T5}
56	GUIDANCE_INTERNAL.DINCDT	float	{T4}	{T3}
57	GUIDANCE_INTERNAL.DTGHA	float	{T3,T5}	{T3,T5}
58	GUIDANCE_INTERNAL.DUMAX	array(1..35) of float	{T5}	{T3,T5}
59	GUIDANCE_INTERNAL.DYMAX	array(1..25) of float	{T5}	{T3,T5}
60	GUIDANCE_INTERNAL.INC	array(1..25) of -32768..32767	{T3,T5}	{T3,T5}
61	GUIDANCE_INTERNAL.INCDT	float	{T4}	{T3}
62	GUIDANCE_INTERNAL.K1Z	array(1..5) of -32768..32767	{T5}	{T3,T5}
63	GUIDANCE_INTERNAL.K2Z	array(1..5) of -32768..32767	{T5}	{T3,T5}
64	GUIDANCE_INTERNAL.MBAL	array(0..3) of float	{T3,T5}	{T3,T5}
65	GUIDANCE_INTERNAL.MCOIF	float	{T5}	{T3}
66	GUIDANCE_INTERNAL.ME	array(1..2) of float	{T3}	{T5}
67	GUIDANCE_INTERNAL.METAG	array(1..2) of float	{T3}	{T5}
68	GUIDANCE_INTERNAL.NCT	array(1..5) of -32768..32767	{T3,T5}	{T3,T5}
69	GUIDANCE_INTERNAL.NORDRE	array(1..25) of -32768..32767	{T3}	{T5}

SHARED-NON-SCALAR-DATA-TABLE-SUMMARY.txt

255 non-integer

70	GUIDANCE_INTERNAL.NORMER	float	{T3}	{T5}
71	GUIDANCE_INTERNAL.NPROP	array(1..5) of -32768..32767	{T5}	{T3, T5}
72	GUIDANCE_INTERNAL.TACQSRIAV	float	{T3, T5}	{T3, T5}
73	GUIDANCE_INTERNAL.TALLUEPS	float	{T3, T5}	{T3, T5}
74	GUIDANCE_INTERNAL.TC	array(0..5) of float	{T3, T5}	{T3, T5}
75	GUIDANCE_INTERNAL.TCB	array(1..3, 0..5) of float	{T3}	{T5}
76	GUIDANCE_INTERNAL.TCOIFPRED	float	{T3, T5}	{T3, T5}
77	GUIDANCE_INTERNAL.TEPCCS	float	{T5}	{T3, T5}
78	GUIDANCE_INTERNAL.TEPCPRED	float	{T3, T5}	{T3, T5}
79	GUIDANCE_INTERNAL.TEPSPRED	float	{T3, T5}	{T3, T5}
80	GUIDANCE_INTERNAL.TRES	array(0..5) of float	{T3, T5}	{T3, T5}
81	GUIDANCE_INTERNAL.TVALID	float	{T3, T5}	{T3, T5}
82	GUIDANCE_INTERNAL.U1	array(1..35) of float	{T3, T5}	{T3, T5}
83	GUIDANCE_INTERNAL.UB	array(1..3, 1..35) of float	{T3}	{T5}
84	GUIDANCE_INTERNAL.XPTSEG	array(0..5, 1..6) of float	{T3, T5}	{T3, T5}
85	GUIDANCE_INTERNAL.XPTSEGB	array(1..3, 0..5, 1...f float	{T3}	{T5}
86	JACKS_INTERNALS.JACKS_DATA	array(eap1..eps) of #<record>	{T3, T4}	{T3, T4}
87	LN1.LN1_PROCEDURE_ACTIVE	#<record>	{}	{T1, T2, T3, T4}
88	LN1.UES10_CONTROL_OBSERVER_ACTIVE	#<record>	{}	{T1, T4}
89	LN1.UES12_FAST_WRITE_ACTIVE	#<record>	{}	{T1, T2, T3, T4}
90	LN1.UES2B_CHANGE_MODE_ANSWER_ACTIVE	#<record>	{}	{T1, T2, T4}
91	LN1.UES4_WRITE_DATA_ACTIVE	#<record>	{}	{T1, T2, T3, T4}
92	LN1.UES5_READ_DATA_ACTIVE	#<record>	{}	{T1, T2, T3, T4}
93	LN1.UES6_RUN_FRAME_ACTIVE	#<record>	{}	{T1, T3, T4}
94	LN1.UES7_STOP_FRAME_ACTIVE	#<record>	{}	{T1, T2, T4}
95	LN1.UES8A_SEND_DATA_ACTIVE	#<record>	{}	{T1, T2, T3, T4}
96	LN1.UES8B_DATA_ACQUISITION_ACTIVE	#<record>	{}	{T1, T2, T3, T4}
97	LN1.UT1_RECONFIGURATION_ACTIVE	#<record>	{}	{T1, T2, T3}
98	LN1.UT7_CONTROL_HARDWARE_ACTIVE	#<record>	{}	{T1, T2, T3, T4}
99	LN2_TIME.SUSPENSION_DATE	#<record>	{T3}	{T1, T2, T3}
100	MEAS_INTERNAL.BGY_DATA	#<record>	{T3, T4}	{T3, T4}
101	MEAS_INTERNAL.SRI_UPSTREAM_DATA	#<record>	{T3, T4}	{T3, T4}
102	NAVIGATION_INTERNAL.ACCNGL	array(1..3) of float	{T5}	{T3}
103	NAVIGATION_INTERNAL.NVRELA	float	{T4}	{T3}
104	NAVIGATION_INTERNAL.TACQSRIAV	float	{T5}	{T3}
105	NAVIGATION_INTERNAL.VEQ	array(1..3) of float	{T5}	{T3}
106	NAVIGATION_INTERNAL.XEQ	array(1..3) of float	{T5}	{T3}
107	OBC_INTERNAL.H0_OBT	#<record>	{T1}	{T4}
108	OBC_INTERNAL.RELBDP	array(bdp1case..bdp2.....65535	{T1, T3}	{T1, T3}
109	OBC_INTERNAL.VALBDP	array(bdp1case..bdp2.....65535	{T1, T3}	{T1, T3}
110	PILOT_INTERNAL.ATTITUDE_STATE	#<record>	{T1}	{T3, T4}
111	PILOT_INTERNAL.BLC	float	{T3, T4}	{T3, T4}
112	PILOT_INTERNAL.BLCDT	float	{T3, T4}	{T3, T4}
113	PILOT_INTERNAL.BLCE	float	{T3, T4}	{T3, T4}
114	PILOT_INTERNAL.BLCEPCDT	float	{T3, T4}	{T3, T4}

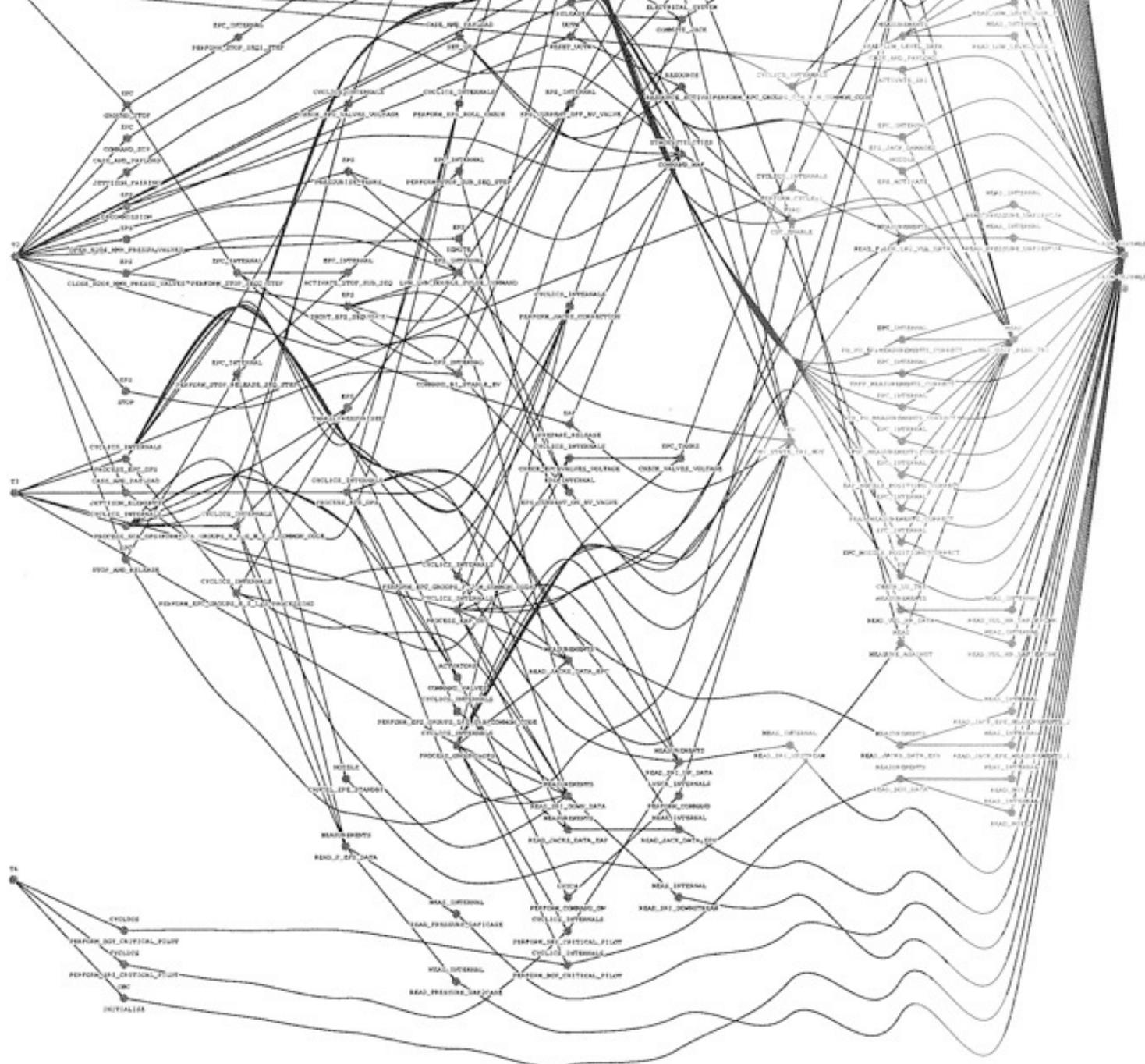
SHARED-NON-SCALAR-DATA-TABLE-SUMMARY.txt

115	PILOT_INTERNAL.BLFDT	float	{T3,T4}	{T3,T4}
116	PILOT_INTERNAL.BRAQUAGE_EAP1	#<record>	{T3,T4}	{T3,T4}
117	PILOT_INTERNAL.BRAQUAGE_EAP2	#<record>	{T3,T4}	{T3,T4}
118	PILOT_INTERNAL.BRAQUAGE_EPX	#<record>	{T3,T4}	{T3,T4}
119	PILOT_INTERNAL.BRC	float	{T3,T4}	{T3,T4}
120	PILOT_INTERNAL.BRCDT	float	{T3,T4}	{T3,T4}
121	PILOT_INTERNAL.BTC	float	{T3,T4}	{T3,T4}
122	PILOT_INTERNAL.BTCDT	float	{T3,T4}	{T3,T4}
123	PILOT_INTERNAL.BTCE	float	{T3,T4}	{T3,T4}
124	PILOT_INTERNAL.BTCEPCDT	float	{T3,T4}	{T3,T4}
125	PILOT_INTERNAL.BTFDT	float	{T3,T4}	{T3,T4}
126	PILOT_INTERNAL.CLC	float	{T3,T4}	{T3,T4}
127	PILOT_INTERNAL.COSALP_IEXP	float	{T4}	{T3}
128	PILOT_INTERNAL.COSIN_PHI	float	{T3,T4}	{T3,T4}
129	PILOT_INTERNAL.COSIN_PSI	float	{T3,T4}	{T3,T4}
130	PILOT_INTERNAL.CTC	float	{T3,T4}	{T3,T4}
131	PILOT_INTERNAL.DTPIL	float	{T4}	{T3}
132	PILOT_INTERNAL.FILTERED_ATTITUDE	#<record>	{T3}	{T4}
133	PILOT_INTERNAL.PUELEPX	array(0..2) of float	{T3,T4}	{T3,T4}
134	PILOT_INTERNAL.PUETEPX	array(0..2) of float	{T3,T4}	{T3,T4}
135	PILOT_INTERNAL.PYSLEPX	array(0..2) of float	{T3,T4}	{T3,T4}
136	PILOT_INTERNAL.PYSLINT	float	{T4}	{T3}
137	PILOT_INTERNAL.PYSTEPX	array(0..2) of float	{T3,T4}	{T3,T4}
138	PILOT_INTERNAL.PYSTINT	float	{T4}	{T3}
139	PILOT_INTERNAL.SINALP_IEXP	float	{T4}	{T3}
140	PILOT_INTERNAL.SINUS_PHI	float	{T3,T4}	{T3,T4}
141	PILOT_INTERNAL.SINUS_PSI	float	{T3,T4}	{T3,T4}
142	PILOT_INTERNAL.UELEPX	array(0..2, 1..5) of float	{T3,T4}	{T3,T4}
143	PILOT_INTERNAL.UETEPX	array(0..2, 1..5) of float	{T3,T4}	{T3,T4}
144	PILOT_INTERNAL.VPUELEPX	array(0..2) of float	{T3,T4}	{T3,T4}
145	PILOT_INTERNAL.VPUETEPX	array(0..2) of float	{T3,T4}	{T3,T4}
146	PILOT_INTERNAL.VPYSLEPX	array(0..2) of float	{T3,T4}	{T3,T4}
147	PILOT_INTERNAL.VPYSLINT	float	{T4}	{T3}
148	PILOT_INTERNAL.VPYSTEPX	array(0..2) of float	{T3,T4}	{T3,T4}
149	PILOT_INTERNAL.VPYSTINT	float	{T4}	{T3}
150	PILOT_INTERNAL.XPIL	array(1..29) of float	{T3,T4}	{T3,T4}
151	PILOT_INTERNAL.YSLEPX	array(0..2, 1..5) of float	{T3,T4}	{T3,T4}
152	PILOT_INTERNAL.YSLINT	array(1..5) of float	{T4}	{T3}
153	PILOT_INTERNAL.YSTEPX	array(0..2, 1..5) of float	{T3,T4}	{T3,T4}
154	PILOT_INTERNAL.YSTINT	array(1..5) of float	{T4}	{T3}
155	PILOT_INTERNAL.YTETAM	float	{T3}	{T4}
156	SRI_INTERNAL.ALF	array(1..3) of float	{T3,T4}	{T3,T4}
157	SRI_INTERNAL.ATTITUDE_ANGLES	#<record>	{T3,T4}	{T3,T4}
158	SRI_INTERNAL.ATTN_INDIC	array(1..3) of -32768..32767	{T3,T4}	{T3,T4}
159	SRI_INTERNAL.ATTNDT	array(1..3) of float	{T3,T4}	{T3,T4}

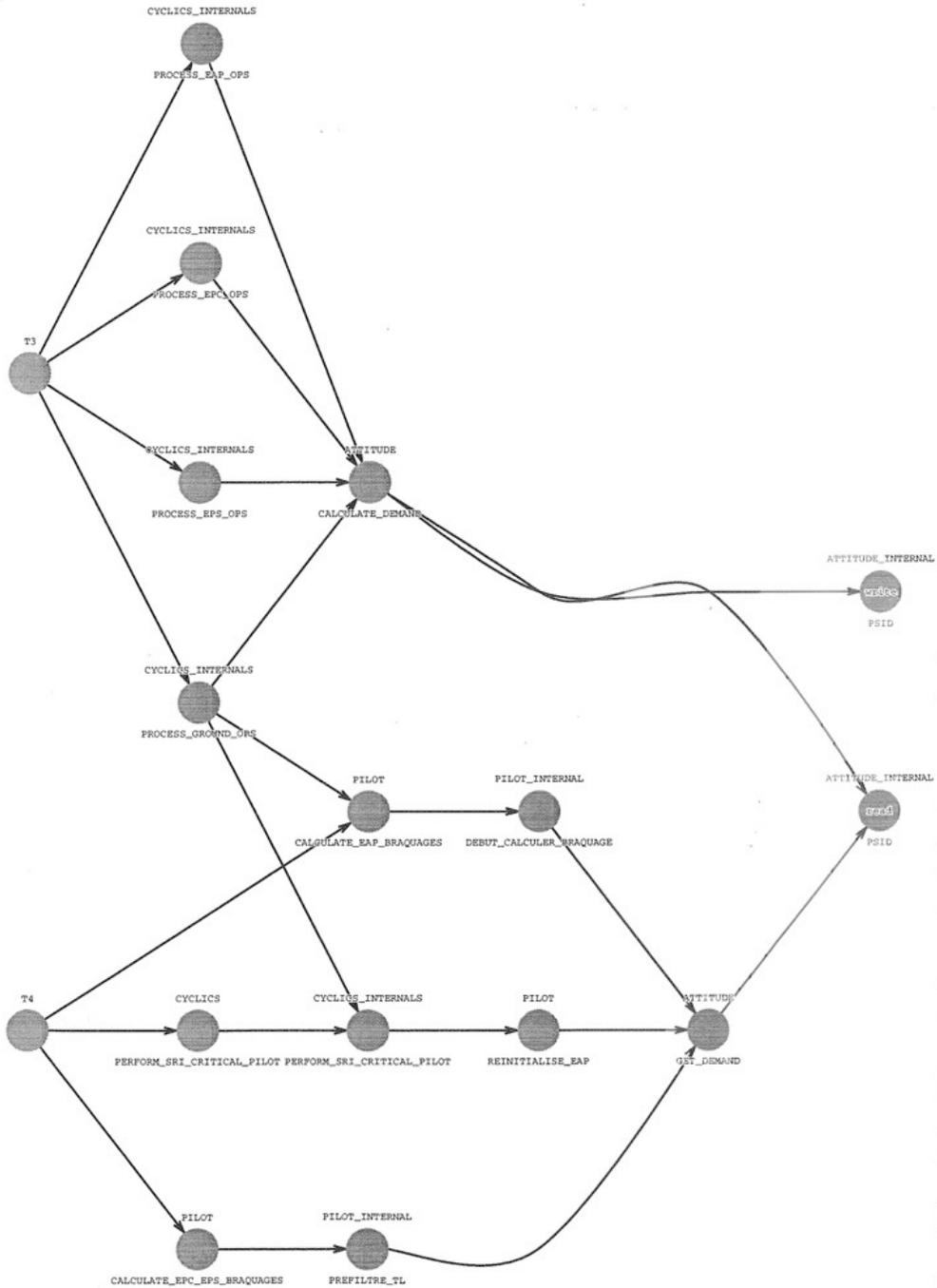
160	SRI_INTERNAL.ATTS_INDIC	array(1..3) of -32768..32767	{T3,T4}	{T3,T4}
161	SRI_INTERNAL.ATTSDT	array(1..3) of float	{T3,T4}	{T3,T4}
162	SRI_INTERNAL.BRC	float	{T3}	{T3,T4}
163	SRI_INTERNAL.IATT	array(1..3) of -32768..32767	{T3,T4}	{T3,T4}
164	SRI_INTERNAL.NREJA	array(1..3) of -32768..32767	{T3,T4}	{T3,T4}
165	SRI_INTERNAL.PHINO	float	{T3,T4}	{T3,T4}
166	SRI_INTERNAL.PHIN1	float	{T3,T4}	{T3,T4}
167	SRI_INTERNAL.PHINF	float	{T3,T4}	{T3,T4}
168	SRI_INTERNAL.PHINP	float	{T3,T4}	{T3,T4}
169	SRI_INTERNAL.PHIS0	float	{T3,T4}	{T3,T4}
170	SRI_INTERNAL.PHIS1	float	{T3,T4}	{T3,T4}
171	SRI_INTERNAL.PHISF	float	{T3,T4}	{T3,T4}
172	SRI_INTERNAL.PHISP	float	{T3,T4}	{T3,T4}
173	SRI_INTERNAL.PREDICTED_ATTITUDE_ANGLES	#<record>	{T4}	{T3}
174	SRI_INTERNAL.PSINO	float	{T3,T4}	{T3,T4}
175	SRI_INTERNAL.PSIN1	float	{T3,T4}	{T3,T4}
176	SRI_INTERNAL.PSINF	float	{T3,T4}	{T3,T4}
177	SRI_INTERNAL.PSINP	float	{T3,T4}	{T3,T4}
178	SRI_INTERNAL.PSIS0	float	{T3,T4}	{T3,T4}
179	SRI_INTERNAL.PSIS1	float	{T3,T4}	{T3,T4}
180	SRI_INTERNAL.PSISF	float	{T3,T4}	{T3,T4}
181	SRI_INTERNAL.PSISP	float	{T3,T4}	{T3,T4}
182	SRI_INTERNAL.TACQSRIAM	float	{T3,T4}	{T3,T4}
183	SRI_INTERNAL.TETAN0	float	{T3,T4}	{T3,T4}
184	SRI_INTERNAL.TETAN1	float	{T3,T4}	{T3,T4}
185	SRI_INTERNAL.TETANF	float	{T3,T4}	{T3,T4}
186	SRI_INTERNAL.TETANP	float	{T3,T4}	{T3,T4}
187	SRI_INTERNAL.TETAS0	float	{T3,T4}	{T3,T4}
188	SRI_INTERNAL.TETAS1	float	{T3,T4}	{T3,T4}
189	SRI_INTERNAL.TETASF	float	{T3,T4}	{T3,T4}
190	SRI_INTERNAL.TETASP	float	{T3,T4}	{T3,T4}
191	SYSTEM.ADDRESS.*	#<record>	{T1,T2,T3,T4}	{T1,T3}
192	TIMER_INTERNAL.DRIFT	float	{T1,T2,T3}	{T4}
193	TIMER_INTERNAL.REFERENCE_OBT	#<record>	{T1,T2,T3}	{T4}
194	TIMER_INTERNAL.START_OF_CURRENT_CYCLE	float	{T1,T3,T4}	{T1,T4}
195	TIMER_INTERNAL.START_OF_CYCLE_OBT	#<record>	{T3,T4}	{T1,T4}
196	TIMER_INTERNAL.START_OF_NEXT_CYCLE	float	{T1,T3,T4}	{T1,T4}
197	UCTM_INTERNAL.BGY_STATUS_DM	#<record>	{T3}	{T3,T4}
198	UCTM_INTERNAL.COEFF_DM	array(1..2) of float	{T3}	{T3,T5}
199	UCTM_INTERNAL.EAP_NOMINAL_DEFLECTION_DM	#<record>	{T3}	{T3,T4}
200	UCTM_INTERNAL.EAP_NOMINAL_PILOT_VECTOR_2_DM	#<record>	{T3}	{T3,T4}
201	UCTM_INTERNAL.EAP_QDP_DEFLECTION_DM	#<record>	{T3}	{T3,T4}
202	UCTM_INTERNAL.EAP_QDP_PILOT_VECTOR_2_DM	#<record>	{T3}	{T3,T4}
203	UCTM_INTERNAL.EPC_EPS_DEFLECTION_DM	#<record>	{T3}	{T4}
204	UCTM_INTERNAL.EPC_EPS_PILOT_VECTOR_2_DM	#<record>	{T3}	{T3,T4}

SHARED-NON-SCALAR-DATA-TABLE-SUMMARY.txt

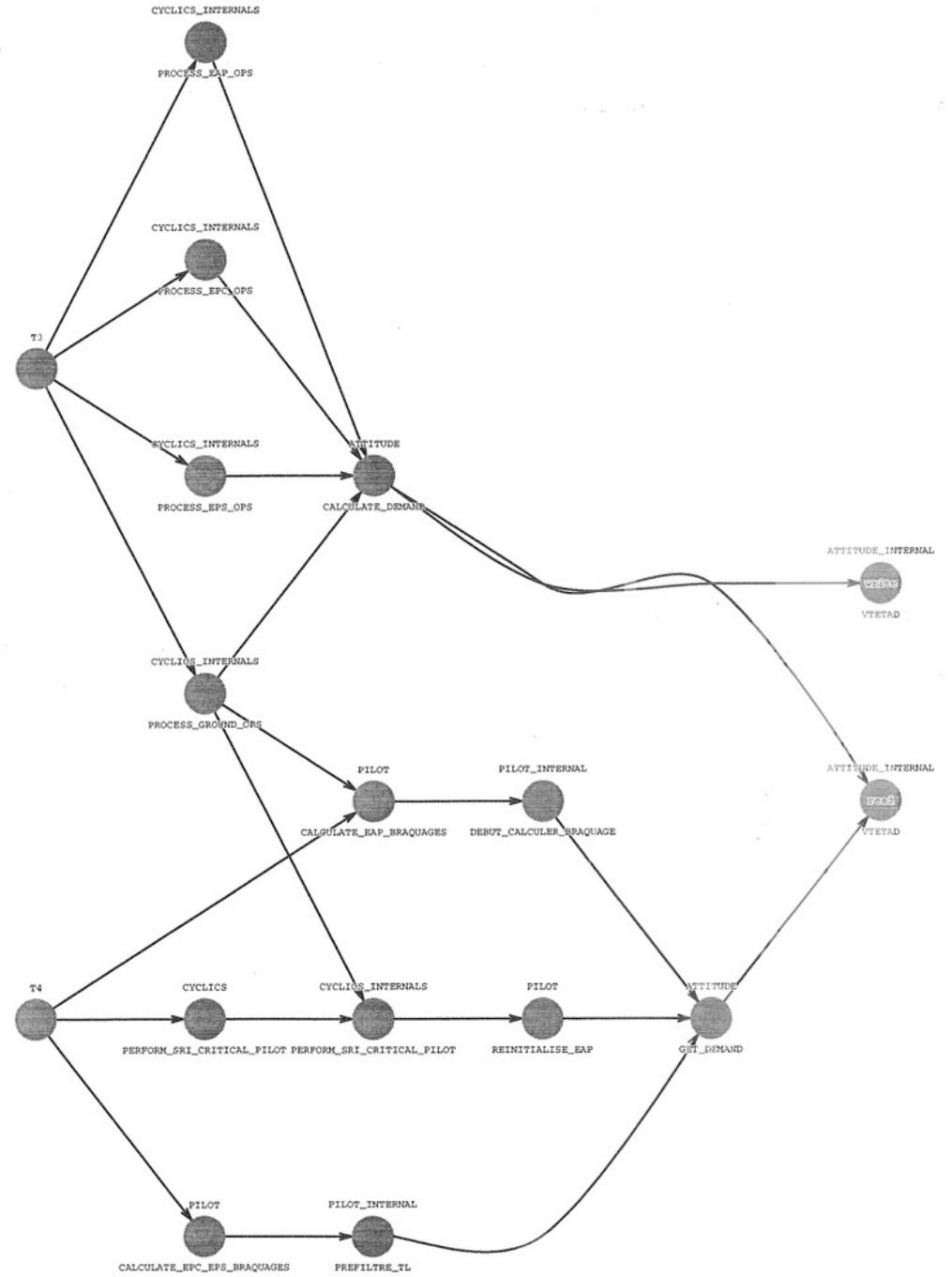
205	UCTM_INTERNAL.GUIDANCE_CONSTRAINTS	array(1..25) of -32768..32767	{T3}	{T3,T5}
206	UCTM_INTERNAL.GUIDANCE_SEGMENT_COMMANDS	array(1..5) of array...f float	{T3}	{T3,T5}
207	UCTM_INTERNAL.HO_DM	#<record>	{T1,T2,T3}	{T4}
208	UCTM_INTERNAL.LIST_EVENT	array(1..8) of #<record>	{T1,T2,T3,T4}	{T1,T2,T3,T4}
209	UCTM_INTERNAL.MSG_1_EAP_NOMINAL_DM	#<record>	{T3}	{T3,T4}
210	UCTM_INTERNAL.MSG_1_EAP_QDP_DM	#<record>	{T3}	{T3,T4}
211	UCTM_INTERNAL.MSG_1_EPC_EPS_DM	#<record>	{T3}	{T3,T4}
212	UCTM_INTERNAL.MSG_6_DM	#<record>	{T1,T2,T3,T4}	{T1,T2,T3,T4,T5}
213	UCTM_INTERNAL.MSG_8_DM	#<record>	{T3}	{T3,T4}
214	UCTM_INTERNAL.NORMER_DM	float	{T3}	{T3,T5}
215	UCTM_INTERNAL.SRI_ATTITUDE_STATUS_DM	array(1..3) of -32768..32767	{T3}	{T3,T4}
216	UCTM_INTERNAL.SRI_AXIS_STATUS_DM	#<record>	{T3}	{T3,T4}
217	UCTM_INTERNAL.VELOCITY_COMMAND_DM	#<record>	{T3}	{T3,T4}



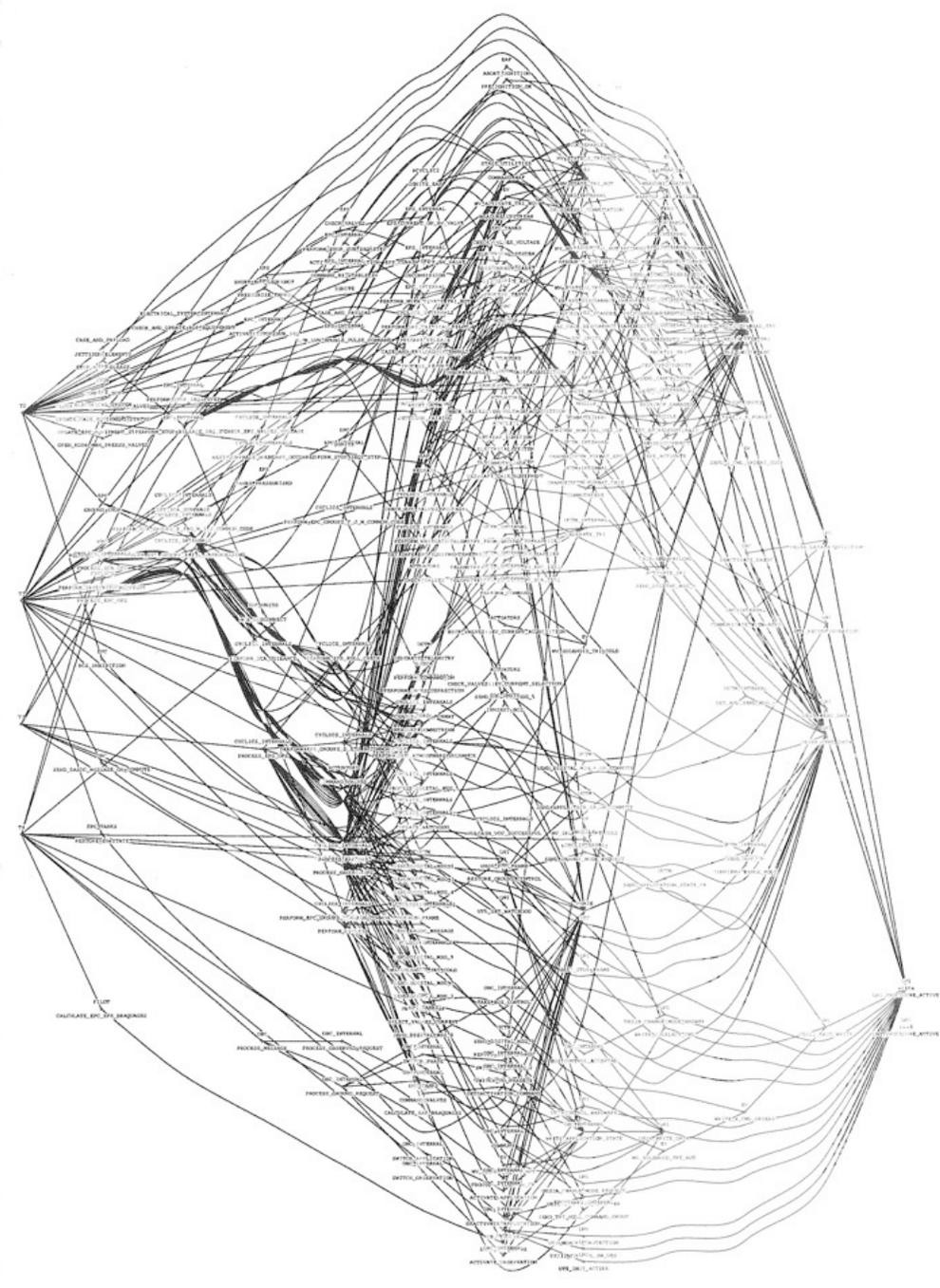
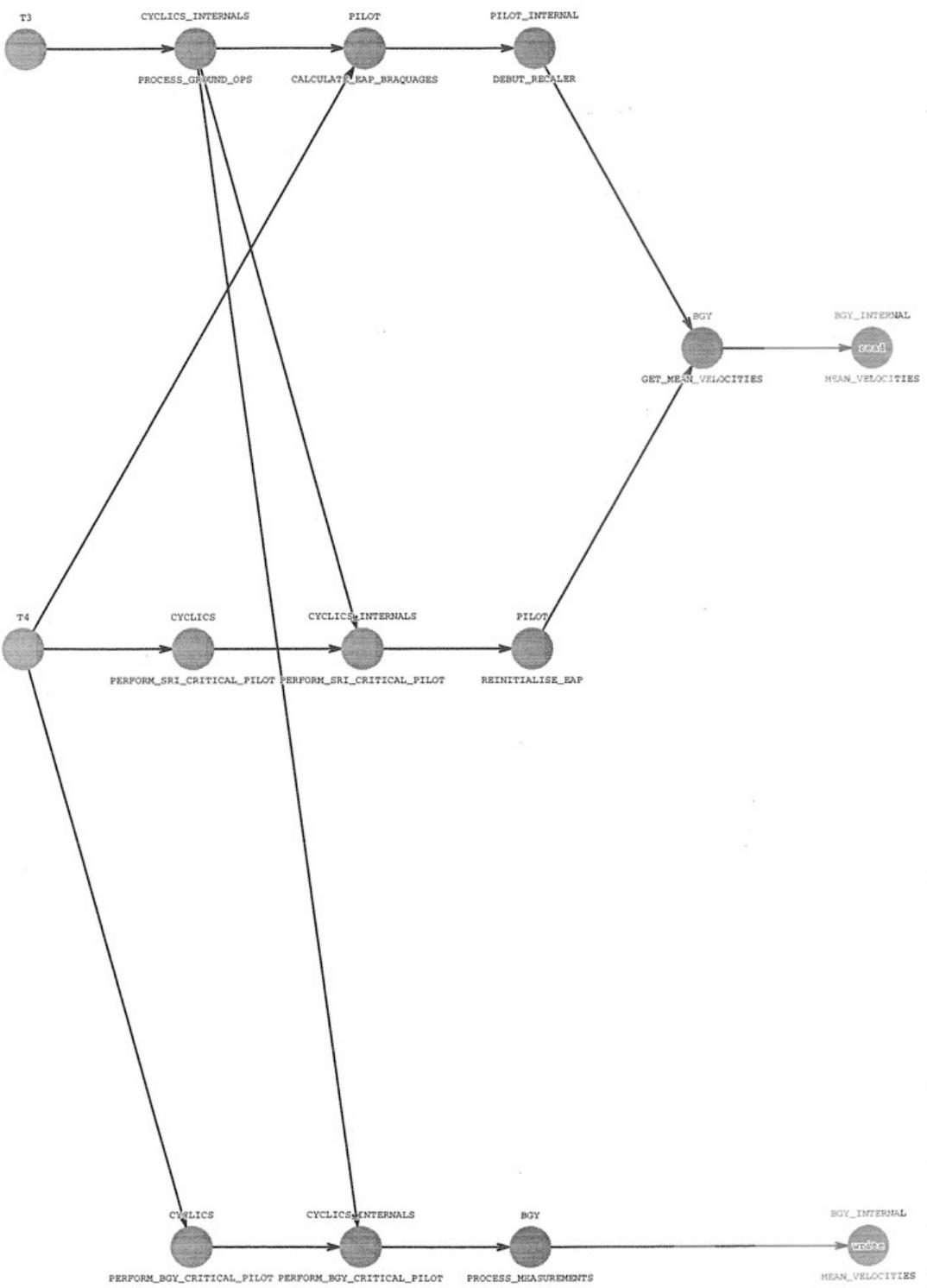
Possibly concurrent accesses for non-scalar #1: <ASM-GLOBALS>



Possibly concurrent accesses for non-scalar #4: ATTITUDE_INTERNAL.PSID



Possibly concurrent accesses for non-scalar #7: ATTITUDE_INTERNAL.VTETAD



Possibly concurrent accesses for non-scalar #8: BOY_INTERNAL.MEAN_VELOCITIES

Possibly concurrent accesses for non-scalar #87: LN1.LN1_PROCEDURE_ACTIVE

File generated on Mar 14, 1997 by IABC V0.1, the INRIA static program analyzer.

Level 0 Concurrency Analysis for scalars accessed by:

- T1) TRT_200HERTZ.TRAITER_INITIAL (procedure), "trt_200hertz.ads" line 74, column 0:
 procedure TRAITER_INITIAL (S : in TE.T_SELECTEUR := TE.SELECTEUR_NUL) ;
 ^
- T2) TRT_200HERTZ.TRAITER_OPERAT (procedure), "trt_200hertz.ads" line 81, column 0:
 procedure TRAITER_OPERAT (S : in TE.T_SELECTEUR := TE.SELECTEUR_NUL) ;
 ^
- T3) TRT_200HERTZ.TRAITER_TEST (procedure), "trt_200hertz.ads" line 88, column 0:
 procedure TRAITER_TEST (S : in TE.T_SELECTEUR := TE.SELECTEUR_NUL) ;
 ^
- T4) TRT_BUS.TRAITER_IT1 (procedure), "trt_bus.ads" line 76, column 0:
 procedure TRAITER_IT1;
 ^
- T5) TRT_BUS.TRAITER_IT2 (procedure), "trt_bus.ads" line 85, column 0:
 procedure TRAITER_IT2;
 ^
- T6) TRT_BUS.TRAITER_IT3 (procedure), "trt_bus.ads" line 94, column 0:
 procedure TRAITER_IT3;
 ^
- T7) TRT_DATA_STORE.TRAITER_ECR_EXCEPT_DS (procedure), "trt_data_store.ads" line 92, column 0:
 procedure TRAITER_ECR_EXCEPT_DS (S:in TE.T_SELECTEUR := TE.SELECTEUR_NUL);
 ^
- T8) TRT_DATA_STORE.TRAITER_ECRITURE_DS (procedure), "trt_data_store.ads" line 76, column 0:
 procedure TRAITER_ECRITURE_DS (S:in TE.T_SELECTEUR := TE.SELECTEUR_NUL);
 ^
- T9) TRT_DATA_STORE.TRAITER_Lecture_DS (procedure), "trt_data_store.ads" line 61, column 0:
 procedure TRAITER_Lecture_DS (S:in TE.T_SELECTEUR := TE.SELECTEUR_NUL);
 ^
- T10) TRT_INITIAL.INITIALISER_OPERAT (procedure), "trt_initial.ads" line 64, column 3:
 procedure INITIALISER_OPERAT (S : in TE.T_SELECTEUR := TE.SELECTEUR_NUL);
 ^
- T11) TRT_SURVEILLANCES.TRAITER_SURVEILLANCES (procedure), "trt_surveillances.ads" line 65, column 0:
 procedure TRAITER_SURVEILLANCES (S : in TE.T_SELECTEUR := TE.SELECTEUR_NUL) ;
 ^
- T12) TRT_SURVEILLANCES.TRAITER_TESTS_INTERNES (procedure), "trt_surveillances.ads" line 57, column 0:
 procedure TRAITER_TESTS_INTERNES (S : in TE.T_SELECTEUR := TE.SELECTEUR_NUL) ;
 ^

	variable	type	reads	writes
1	ALIGNEMENT.G_M_ETAT_ALIGN	false..true	{T3,T7,T1}	{T2}
2	ALIGNEMENT.G_M_SURVEILLANCE	false..true	{T3,T1}	{T2}
3	BOOLEENS_SYSTEME.G_M_VALEUR_BTf	false..true	{T2,T3,T12,T1}	{T2,T3,T1}
4	DIALOGUE_ARC.G_M_ECRIRE_ARC	false..true	{}	{T3,T1}
5	DIALOGUE_ARC.G_M_NOMBRE_MOTS_OPERAT	0..255	{T2,T3,T1}	{T2,T3,T1}
6	DIALOGUE_ARC.G_M_PANNE_ARC	false..true	{T2,T3,T1}	{T2,T3,T1}
7	DIALOGUE_BUS.G_M_CASE_TEST	0..65535	{T2,T3,T7,T1}	{T2,T3,T7,T1}
8	DIALOGUE_BUS.G_M_CYCLES_DEPUS_ANOMALIE	0..65535	{T2,T3,T1}	{T2,T3,T7,T1}
9	DIALOGUE_BUS.G_M_MODIFIE	false..true	{T2,T3,T7,T1}	{T2,T3,T7,T1}
10	DIALOGUE_BUS.G_M_NB_ANOMALIES_PROTONS	0..65535	{T2,T3,T7,T1}	{T8,T2,T3,T7,T1}
11	DIALOGUE_BUS.G_M_SEUIL_PROTONS_QUATER	0..65535	{T2,T3,T7,T1}	{T2,T3,T7,T1}
12	DIALOGUE_BUS.G_M_SEUIL_PROTONS_VITESSE	0..65535	{T3,T7,T1}	{T2}
13	DIALOGUE_BUS.G_M_SURVEILLANCE_PROTONS_ACTIVE	false..true	{T3,T7,T1}	{T2}
			{T2,T3,T7,T1}	{T2,T3,T7,T1}

SHARED-SCALAR-DATA-TABLE-SUMMARY.txt

14	GESTION_DATA_STORE.G_C_TEMPS_SA...LE_EN_COURS	0..2147483647	{T2,T3,T1}	{T2,T3,T1}
15	GESTION_DATA_STORE.G_M_ADRESSE_LUE	0..65535	{T2,T3,T1}	{T2,T3,T1}
16	GESTION_DATA_STORE.G_M_ADRESSE_PFA	integer mod 429496729...94967295	{T2,T3,T1}	{T9}
17	GESTION_DATA_STORE.G_M_ANGLE_A_SAUVEGARDER	false..true	{T2,T3,T1}	{T2,T3,T1}
18	GESTION_DATA_STORE.G_M_BORNE_MESSAGE_ETENDU	0..65535	{T2,T3,T7,T1}	{T8}
19	GESTION_DATA_STORE.G_M_CONTEXTE_PANNE_A_LIRE	false..true	{T2,T3,T7,T1}	{T2,T3,T7,T1}
20	GESTION_DATA_STORE.G_M_CRC_TROU...SAUVEGARDER	false..true	{T2,T3,T1}	{T2,T3,T1}
21	GESTION_DATA_STORE.G_M_DS_01_MODIFIEE	true..false	{}	{T8,T2}
22	GESTION_DATA_STORE.G_M_DS_04_MODIFIEE	true..false	{}	{T8,T2}
23	GESTION_DATA_STORE.G_M_DS_05_MODIFIEE	true..false	{}	{T8,T2}
24	GESTION_DATA_STORE.G_M_DS_06_MODIFIEE	true..false	{}	{T8,T2}
25	GESTION_DATA_STORE.G_M_DS_07_MODIFIEE	true..false	{}	{T8,T2}
26	GESTION_DATA_STORE.G_M_DS_08_MODIFIEE	true..false	{}	{T8,T2}
27	GESTION_DATA_STORE.G_M_DS_VALEUR_CS	0..65535	{T2,T3,T1}	{T2,T3,T1}
28	GESTION_DATA_STORE.G_M_ETAT_PREC	e_init0..e_pr	{T2,T3,T1}	{T2,T3,T1}
29	GESTION_DATA_STORE.G_M_LECTURE_DS_DEMANDEE	false..true	{T9,T2,T3,T1}	{T9,T2,T3,T1}
30	GESTION_DATA_STORE.G_M_MODIF_TABLE_CPT_STAT	false..true	{T2,T3,T1}	{T2,T3,T1}
31	GESTION_DATA_STORE.G_M_MODIF_TA...T_DECLENCHE	false..true	{T2,T3,T1}	{T2,T3,T1}
32	GESTION_DATA_STORE.G_M_NB_MSG_EDS	-32768..32767	{T8,T9,T2,T4,T3,T5,T6,T1}	{T8,T9,T2,T4,T3,T5,T6,T1}
33	GESTION_DATA_STORE.G_M_NB_MSG_LDS	-32768..32767	{T8,T9,T2,T4,T3,T5,T6,T1}	{T8,T9,T4,T5,T6}
34	GESTION_DATA_STORE.G_M_NLE_PANNE	false..true	{T2,T3,T1}	{T2,T3,T1}
35	GESTION_DATA_STORE.G_M_NOMBRE_A_LIRE	0..2147483647	{T2,T3,T1}	{T9}
36	GESTION_DATA_STORE.G_M_OK_ECRITURE_BAL	false..true	{T2,T4,T3,T5,T6,T1}	{T2,T3,T10,T1}
37	GESTION_DATA_STORE.G_M_PFA_LU	false..true	{T2,T3,T1}	{T2,T3,T1}
38	GESTION_DATA_STORE.G_M_ROMCHKT_A_SAUVEGARDER	false..true	{T2,T3,T1}	{T2,T3,T1}
39	GESTION_DATA_STORE.G_M_TABLE_09_A_SAUVEGARDER	false..true	{T2,T3,T1}	{T2,T3,T1}
40	GESTION_DATA_STORE.G_M_TABLE_0A_A_SAUVEGARDER	false..true	{T2,T3,T1}	{T2,T3,T1}
41	GESTION_DATA_STORE.G_M_TABLE_0A...S_MODIFIEES	false..true	{T2,T3,T1}	{T2,T3,T1}
42	GESTION_DATA_STORE.G_M_TABLE_0B_A_SAUVEGARDER	false..true	{T2,T3,T1}	{T2,T3,T1}
43	GESTION_DATA_STORE.G_M_TABLE_0D_A_SAUVEGARDER	false..true	{T2,T3,T1}	{T2,T3,T1}
44	GESTION_DATA_STORE.G_M_TABLE_0E_A_SAUVEGARDER	false..true	{T2,T3,T1}	{T2,T3,T1}
45	GESTION_DATA_STORE.G_M_VALEUR_PFA	0..65535	{T2,T3,T1}	{T2,T3,T1}
46	GESTION_ETATS.G_M_ETAT	e_init0..e_pr	{T8,T2,T3,T1}	{T2,T3,T1}
47	GESTION_ETATS.G_M_ETAT_ALIGNEMENT	e_null..e_algp	{T2}	{T2,T3,T1}
48	GESTION_ETATS.G_M_ETAT_CALIBRATION	e_null..e_cvlp	{T2,T3,T1}	{T2,T3,T1}
49	GESTION_ETATS.G_M_ETAT_CBA	false..true	{T3}	{T2}
50	GESTION_ETATS.G_M_ETAT_INTERNE	e_alim..e_hold	{T2,T3,T1}	{T2,T3,T1}
51	GESTION_ETATS.G_M_ETAT_TEST	e_module_1..e_module_3	{T3}	{T2,T3,T1}
52	GESTION_ETATS.G_M_PFA_ETAT	e_pr..e_null	{T8,T2,T3,T1}	{T2,T3,T1}
53	GESTION_ETATS.G_M_PFV_ETAT	e_pr..e_null	{T8,T2,T3,T1}	{T2,T3,T1}
54	SURVEILLANCES.G_C_COMPTEUR_TRAI...EMPERATURES	0..65535	{T3}	{T3,T1}
55	SURVEILLANCES.G_M_BOOLEEN_COUPL...URVEILLANCE	false..true	{T2}	{T2,T3,T7,T1}
56	SURVEILLANCES.G_M_BOOLEEN_COUPLEUR_TESTS	false..true	{T12}	{T12,T1}
57	SURVEILLANCES.G_M_BOOLEEN_DATA_STORE	false..true	{T11}	{T8,T11,T7,T12}
58	SURVEILLANCES.G_M_CHECKSUM_LUE	0..65535	{T11}	{T8}
59	SURVEILLANCES.G_M_CHECKSUM_ROM	0..65535	{T2,T3,T1}	{T11,T12}
60	SURVEILLANCES.G_M_CRC_LUS	false..true	{T2,T3,T1}	{T2,T3,T1}
61	SURVEILLANCES.G_M_DUREE_BUTEE	0..65535	{T3,T1}	{T2}
62	SURVEILLANCES.G_M_FIN_TESTS_INTERNES	false..true	{T3,T1}	{T3,T12,T1}
63	SURVEILLANCES.G_M_INCIDENT_ROM	false..true	{T2,T3,T1}	{T11,T2,T3,T12,T1}
64	SURVEILLANCES.G_M_MODIFICATION...E_COMPTEURS	false..true	{T2,T3,T1}	{T11,T2,T3,T1}
65	SURVEILLANCES.G_M_MODIFICATION_TABLE_PANNES	false..true	{T2,T3,T1}	{T11,T2,T3,T12,T1}
66	SURVEILLANCES.G_M_MODIFICATION_TABLE_TESTS	false..true	{T2,T3,T1}	{T11,T2,T3,T12,T1}
67	SURVEILLANCES.G_M_MOT_ETAT_ARC_DECLENCHE	0..65535	{T3}	{T3,T1}
68	SURVEILLANCES.G_M_SURVEILLANCE_IMAGE_ET_SAUT	false..true	{T3}	{T3,T1}

69	SURVEILLANCES.G_M_TEMPERATURE2_APRES_FILTRAGE	-32768..32767	{}	{T8,T2}
70	SURVEILLANCES.G_M_TEMPERATURE3_APRES_FILTRAGE	-32768..32767	{}	{T8,T2}
71	SURVEILLANCES.G_M_TEST_ROM_TD_FINI	false..true	{T11}	{T12}
72	SURVEILLANCES.G_M_TEST_SAUT_ALC_TERMINE	false..true	{T3}	{T3,T1}
73	SURVEILLANCES.G_M_VALEUR_CRC_ARC	0..65535	{T2,T3}	{T3,T1}
74	SURVEILLANCES.G_M_VALEUR_CRC_PP	0..65535	{T2,T3,T1}	{T12}
X75	SURVEILLANCES.G_M_VALEUR_CS	0..65535	{T2,T3,T1}	{T11,T2,T3,T1}
76	TENSIONS.G_M_AXE_EN_COURS	e_aucun..e_z	{T3}	{T3,T1}
77	TENSIONS.G_M_CDG	false..true	{T3}	{T3,T1}
78	TENSIONS.G_M_COMPTeur_TENSIONS	0..255	{T2,T3,T1}	{T2,T3,T1}
79	TENSIONS.G_M_DEBUT_SAUT	0..2147483647	{T2,T3,T1}	{T2,T3,T1}
80	TENSIONS.G_M_DEBUT_STABILITE	0..2147483647	{T3}	{T3,T1}
81	TENSIONS.G_M_ETAPE_STABILITE	e_stabilite_1..e_stabilite_3	{T3}	{T3,T1}
82	TENSIONS.G_M_ETAPE_TEST_ALC	e_test_stabilite_1..e...bilite_3	{T3}	{T3,T1}
83	TENSIONS.G_M_SAUT_EN_COURS	e_aucun..e_z	{T2}	{T2,T3,T1}
84	TENSIONS.G_M_STABILITE_NON_ARMEE	false..true	{T3}	{T3,T1}
85	TRT_200HERTZ.G_C_CYCLE	0..2147483647	{T2,T3,T1}	{T2,T3,T1}
86	TRT_200HERTZ.G_C_CYCLE_200	0..2147483647	{T2,T3,T1}	{T2,T3,T1}
87	TRT_200HERTZ.G_C_CYCLE_25_HERTZ	0..2147483647	{T2}	{T2,T3,T1}
88	TRT_200HERTZ.G_C_CYCLE_50_HERTZ	0..2147483647	{T2,T3,T1}	{T2,T3,T1}
89	TRT_200HERTZ.G_C_CYCLE_625_HERTZ	0..2147483647	{T2,T3,T1}	{T2,T3,T1}
90	TRT_200HERTZ.G_C_TEMPS_ALIGNMENT	0..2147483647	{T3,T1}	{T2}
91	TRT_200HERTZ.G_C_TEMPS_MST	0..2147483647	{T2,T3,T1}	{T2,T3,T1}
92	TRT_200HERTZ.G_C_TEMPS_VOL	0..2147483647	{T3,T1}	{T2}
93	TRT_200HERTZ.G_M_CODE_ACTION	e_ecrire_arc..e_fin_test	{T3}	{T3,T1}
94	TRT_200HERTZ.G_M_CODE_TEST	e_fin_tests_internes....ee_moins	{T3}	{T3,T1}
95	TRT_200HERTZ.G_M_ETAT_INTERNE	e_alim..e_hold	{T2,T3,T1}	{T2,T3,T1}
96	TRT_200HERTZ.G_M_NOMBRE_TESTS_COUPLEUR	-32768..32767	{}	{T3,T1}
97	TRT_BUS.G_M_COMMANDE	e_mst..e_hold	{T2,T3,T1}	{T4,T5,T6}
98	TRT_BUS.G_M_MODIFIE	false..true	{T2,T3,T1}	{T2,T4,T3,T5,T6,T1}

profitez des
LT

Collaboration avec l'Aérospatiale

(renommé EADS, puis Airbus)

- beau contrat (**700kf** in 3 months...)
- plusieurs **bogues** [gonthier] +++
- analyse d'intervalles [deutsch] ++
- travail sur le flottant [deutsch] +-

- simulation de l'automate du LSSI en Promela [gonthier] --
- participation aux comités de qualification committees du vol 502
++ [deutsch, gonthier, doligez, rouaix, skubi]
- article dans une conférence internationale conference
d'avionnique [deutsch, gonthier]

Analyse statique

[Cousot 's 78]

```
x := 1
```

```
{X1 = ∅}
```

```
while x < 10000 do
```

```
  {X2 = ∅}
```

```
  x := x + 1;
```

```
  {X3 = ∅}
```

```
done;
```

```
{X4 = ∅}
```

$$X_1 = [1, 1]$$

$$X_2 = (X_1 \cup X_3) \cap [0, 9999]$$

$$X_3 = X_2 \oplus 1$$

$$X_4 = (X_1 \cup X_3) \cap [10000, +\infty]$$

Analyse statique

```
x := 1
```

```
{X1 = [1, 1]}
```

```
while x < 10000 do
```

```
  {X2 = [1, 1]}
```

```
  x := x + 1;
```

```
  {X3 = [2, 2]}
```

```
done;
```

```
{X4 = ∅}
```

$$X_1 = [1, 1]$$

$$X_2 = (X_1 \cup X_3) \cap [0, 9999]$$

$$X_3 = X_2 \oplus 1$$

$$X_4 = (X_1 \cup X_3) \cap [10000, +\infty]$$

Analyse statique

```
x := 1
```

```
{X1 = [1, 1]}
```

```
while x < 10000 do
```

```
  {X2 = [1, 2]}
```

```
  x := x + 1;
```

```
  {X3 = [2, 3]}
```

```
done;
```

```
{X4 = ∅}
```

$$X_1 = [1, 1]$$

$$X_2 = (X_1 \cup X_3) \cap [0, 9999]$$

$$X_3 = X_2 \oplus 1$$

$$X_4 = (X_1 \cup X_3) \cap [10000, +\infty]$$

Analyse statique

```
x := 1
```

```
{X1 = [1, 1]}
```

```
while x < 10000 do
```

```
  {X2 = [1, 3]}
```

```
  x := x + 1;
```

```
  {X3 = [2, 4]}
```

```
done;
```

```
{X4 = ∅}
```

$$X_1 = [1, 1]$$

$$X_2 = (X_1 \cup X_3) \cap [0, 9999]$$

$$X_3 = X_2 \oplus 1$$

$$X_4 = (X_1 \cup X_3) \cap [10000, +\infty]$$

Analyse statique

```
x := 1
```

```
{X1 = [1, 1]}
```

```
while x < 10000 do
```

```
{X2 = [1, +∞]}
```

```
  x := x + 1;
```

```
{X3 = [2, +∞]}
```

```
done;
```

```
{X4 = [10000, +∞]}
```

$$X_1 = [1, 1]$$

$$X_2 = (X_1 \cup X_3) \cap [0, 9999]$$

$$X_3 = X_2 \oplus 1$$

$$X_4 = (X_1 \cup X_3) \cap [10000, +\infty]$$

Elargissement



Analyse statique

```
x := 1
```

```
{X1 = [1, 1]}
```

```
while x < 10000 do
```

```
  {X2 = [1, 9999]}
```

```
  x := x + 1;
```

```
  {X3 = [2, 10000]}
```

```
done;
```

```
{X4 = [10000, 10000]}
```

$$X_1 = [1, 1]$$

$$X_2 = (X_1 \cup X_3) \cap [0, 9999]$$

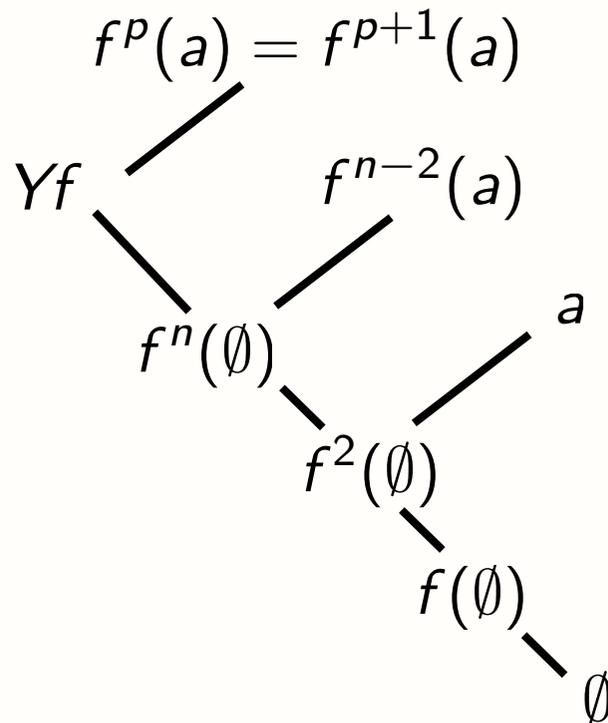
$$X_3 = X_2 \oplus 1$$

$$X_4 = (X_1 \cup X_3) \cap [10000, +\infty]$$

Point fixe

Analyse statique

- (D, \sqsubseteq) treillis complet, f fonction monotone
- f a un plus petit point-fixe (Knaster-Tarski)



Analyse statique d'alias

- les alias sont **inoffensifs** avec des variables **non modifiables** (en fait elles facilitent l'allocation mémoire automatique)
- les langages fonctionnels (Haskell, ML, Ocaml) privilégient l'usage des variables non modifiables.
- les alias sont **dangereux** avec des variables **modifiables** (ADA, C, Java)
- trouver les **alias** est souvent le premier pas d'un analyseur statique

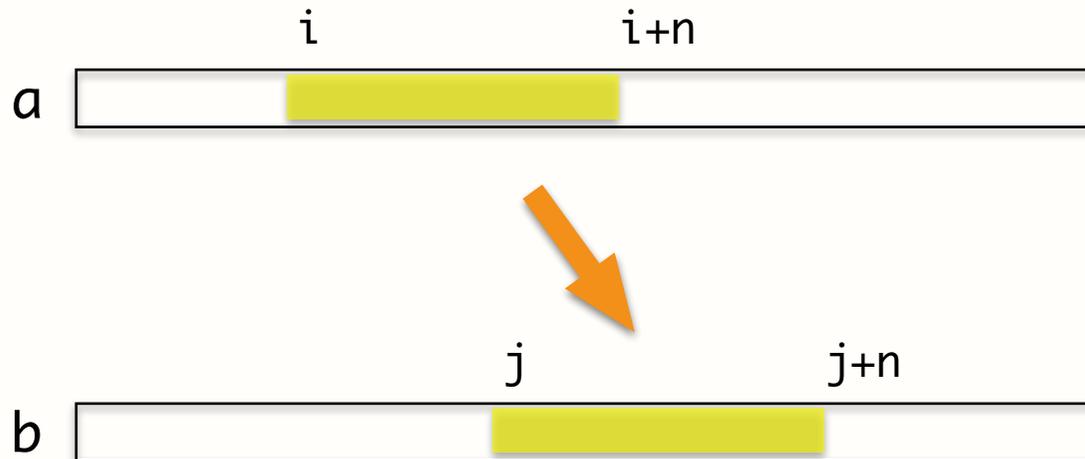
Analyse statique d'alias

- flow-sensitive methods (CFA) non tractable on large programs.
- **flow-insensitive** methods [Andersen 94] deal in a few seconds on several 1000 *locs*.
- **flow-insensitive** and **context-insensitive** methods are even more efficient [Steensgaard 96, Das 00, Heinze/Tardieu 02]
- other contextual methods [Deutsch 94]

Analyse statique d'alias

- le programme suivant est-il correct ?

```
def copy (a, i, b, j, n) :  
    for k in range (n) :  
        b[j+k] = a [i+k]
```



OCTOBRE 1997

A502 Suspense et Succès: rapport sur le vol A502

The 17 months of effort after the June 1996 failure paid off on 30th October 1997 when Ariane 5 fully completed its second qualification flight. There was, however, some suspense during the flight when engineers realized in real time that the launcher was being submitted to an **excessive roll** after separation of the solid boosters and up to the end of the cryogenic stage flight

“Le logiciel a marché à 120%”

Après ...

- analyse du code des **ARD**, **ATV** et autres satellites [deutsch]
- **Polyspace**: start-up company [deutsch, pilaud]
- règles de programmation du CNES [deutsch, gonthier]
- expertise du code de **Columbus** [jjl, gonthier, blanchet, muller]
- règles de programmation de l'ESA [gonthier, jjl]

- à l'ENS, l'analyseur Astrée fait bien mieux pour le **A380**
- toujours de beaux jours pour l'analyse statique de programmes
- ... et la vérification de programmes (embarqués ou pas)

- notre projet Moscova a été mieux évalué:
1994 = projet "pizza"  1998 = projet "exemplaire"

Conclusion

- Ariane 501 a démontré l'importance des bogues **logiciel** en vraie grandeur
- On peut utiliser des méthodes **élémentaires** des théories de la programmation et de la concurrence
- Analyse de programmes **existants**
- Application des résultats de la **recherche** (IABC)
- **Lot of fun** ...