Overview

- Introduction
  - Astrium Space Transportation Case study
    - SCADE modelling
    - Data handling
    - Numerical algorithm
    - Event driven
  - Alfa
  - Gnatprove
  - Gnattest
- Conclusion
Astrium products
Tools

- GNAT
  - gnatpro-7.1.0w-20120514-45-i686-pc-mingw32-bin.exe

- gnatprove
  - hilite-0.2w-20120514-i686-pc-mingw32-bin.exe

- GPS
  - gps-5.2.0w-20120514-i686-pc-mingw32.exe

- SCADE Suite
  - 6.3.1
Overview

- Introduction

- **Astrium Space Transportation Case study**
  - SCADE modelling
  - Data handling
  - Numerical algorithm
  - Event driven

- Alfa
- Gnatprove
- Gnattest
- Conclusion
Technical scope of the case study

MVM

TM/TC

GNC

Environment

Sensors

Navigation

Guidance

Control

Actuators

ECS

EPC

EAP

GNCSensors

Actuators

Control

Guidance

Navigation

Environment

TM

TC
Size of the Astrium case study

```
/HiLite/CASE_STUDY/TMTC/ADA/src
$ find . -name "*.ad[bs]" -exec cat {} \; | wc -l
54610

/HiLite/CASE_STUDY/TMTC/ADA/src
$ find . -name "*.ad[bs]" -exec grep "\;" {} \; | wc -l
12543

/HiLite/CASE_STUDY/TMTC/ADA/testing
$ find . -name "*.ad[bs]" -exec cat {} \; | wc -l
17738

/HiLite/CASE_STUDY/TMTC/ADA/testing
$ find . -name "*.ad[bs]" -exec grep "\;" {} \; | wc -l
5524
```
Overview

- Introduction

- Astrium Space Transportation Case study
  - SCADE modelling
    - Data handling
    - Numerical algorithm
    - Event driven

- Alfa
- Gnatprove
- Gnattest
- Conclusion
Solar wing deployment

- Acyclic events
- Redundancy (FDIR)
- Automaton oriented

(stdout)

The Flight Application Software powers thermal knives in order to deploy the solar wings

Software part modelled in SCADE
Mode automaton

Diagram showing the states and transitions of a mode automaton.
Activation conditions
begin
  if (Ctx.init_5) then
    SM_FSM_state_sel_1_1 := Kcg_Types.SSM_st_SURVIVAL;
    SM_AP_state_sel_1_1 := Kcg_Types.SSM_st_INIT_2;
    SC_t1_1_1 := P5_SGS_LIB_TYPES.IMPORTED_CONSTANTS.C_D_SC;
    last_RT_INHIBIT_TABLE_1_1 :=
      P5_SGS_LIB_TYPES.MAIN.C_D_SGS_RT_INHIBIT_TABLE;
    last_RT_INHIBIT_EQPT_1_1 := P5_SGS_LIB_TYPES.MAIN.C_D_RT_INHIBIT_EQPT;
    last_ALARM_ID_1_1 := Kcg_Types.NO_ALARM;
  else
    SM_FSM_state_sel_1_1 := Ctx.SM_FSM_state_nxt_1_1;
    SM_AP_state_sel_1_1 := Ctx.SM_AP_state_nxt_1_1;
    SC_t1_1_1 := Ctx.rem_SC;
    last_RT_INHIBIT_TABLE_1_1 := Ctx.RT_INHIBIT_TABLE_1_1;
    last_RT_INHIBIT_EQPT_1_1 := Ctx.RT_INHIBIT_EQPT_1_1;
    last_ALARM_ID_1_1 := Ctx.ALARM_ID_1_1;
  end if;
  case (SM_FSM_state_sel_1_1) is
    when Kcg_Types.SSM_st_SURVIVAL =>
      SM_SGS_SC_FULL_DEPLOYMENT_reset_act_1_1 :=
        (((SGS_CMD.CMD = Kcg_Types.SGS_F_DEPLOYMENT_N) or
          (SGS_CMD.CMD = Kcg_Types.SGS_F_FIXED_N)) or
         (SGS_CMD.CMD = Kcg_Types.SGS_F_ROTATING_3)) or
        (SGS_CMD.CMD = Kcg_Types.SGS_F_ROTATING_N)) or
       (SGS_CMD.CMD = Kcg_Types.SGS_F_STOWED_N);
    tmp_1_4_2 := (((SGS_CMD.CMD = Kcg_Types.SGS_F_SURV_INIT) or
      (SGS_CMD.CMD = Kcg_Types.SGS_F_SURV_ROTATING)) or
       (SGS_CMD.CMD = Kcg_Types.SGS_F_SURV_POS_ZERO)) or
      (SGS_CMD.CMD = Kcg_Types.SGS_F_SURV_FIXED);
    if (SM_SGS_SC_FULL_DEPLOYMENT_reset_act_1_1) then
      SM_FSM_state_act_1_1 := Kcg_Types.SSM_st NOMINAL;
      L17_2_1_1 := True;
    else

Overview

- Introduction
- Astrium Space Transportation Case study
  - SCADE modelling
  - Data handling
    - Numerical algorithm
    - Event driven
- Alfa
- Gnatprove
- Gnattest
- Conclusion
### Telemetry / Telecommand

**TMTCTMTC**

**ECSS-E-70-41A**

30 January 2003

---

**Space engineering**

Ground systems and operations — Telemetry and telecommand packet utilization

---

<table>
<thead>
<tr>
<th>Packet Header (48 Bits)</th>
<th>Packet Data Field (Variable)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Packet ID</td>
</tr>
<tr>
<td></td>
<td>Packet Sequence Control</td>
</tr>
<tr>
<td></td>
<td>Packet Length</td>
</tr>
<tr>
<td></td>
<td>Data Field Header (Optional) (see Note 1)</td>
</tr>
<tr>
<td></td>
<td>Application Data</td>
</tr>
<tr>
<td></td>
<td>Spare</td>
</tr>
<tr>
<td></td>
<td>Packet Error Control (see Note 2)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Version Number (=0)</th>
<th>Type (-1)</th>
<th>Data Field Header Flag</th>
<th>Application Process ID</th>
<th>Sequence Flags</th>
<th>Sequence Count</th>
<th>Data Field Header (Optional) (see Note 1)</th>
<th>Application Data</th>
<th>Spare</th>
<th>Packet Error Control (see Note 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>2</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Variable</td>
<td></td>
<td></td>
<td>Variable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Variable</td>
<td></td>
<td></td>
<td>Variable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Variable</td>
<td></td>
<td></td>
<td>Variable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16</td>
<td></td>
<td></td>
<td>16</td>
</tr>
</tbody>
</table>

---

**All the space you need**

10/05/2011 — p15
Structure of telemetry / telecommand packets

```plaintext
type T_Packet_ID is
  record
    Version_Number : T_Version_Number;
    Packet_Type     : T_Packet_Type;
    Data_Field_Header_Flag : T_Data_Field_Header_Flag;
    Application_Process_Id : Tmte.Entities.T_Application_Process_Id;
  end record;
for T_Packet_ID use
  record
    Version_Number at 0 range 0 .. 2;
    Packet_Type at 0 range 3 .. 3;
    Data_Field_Header_Flag at 0 range 4 .. 4;
    Application_Process_Id at 0 range 5 .. 15;
  end record;
for T_Packet_ID'Size use 16;

type T_Packet_Header is
  record
    Packet_ID : T_Packet_ID;
    Packet_Sequence_Control : T_Packet_Sequence_Control;
    Packet_Length : T_Packet_Length;
  end record;
for T_Packet_Header use
  record
    Packet_ID at 0 range 0 .. 15;
    Packet_Sequence_Control at 0 range 16 .. 31;
    Packet_Length at 0 range 32 .. 47;
  end record;
for T_Packet_Header'Size use 48;
```
Verification of telecommand packets

```literate
procedure Check_Telecommand
{
  Data_Field_Header : in T_Data_Field_Header;
  Tc_Error : out T_Tc_Error
}
is
  Sequence_Count : Tmtc.Packet.T_Sequence_Count;
begin
  if not Packet_Header.Packet_ID.Version_Number'Valid then
    Tc_Error := Invalid_Version_Number;
  elsif not Packet_Header.Packet_ID.Packet_Type'Valid then
    Tc_Error := Invalid.Packet_Type;
  elsif not ( Packet_Header.Packet_ID.Packet_Type = Tmtc.Packet.Tc_Packet ) then
    Tc_Error := Invalid.Packet_Type;
  elsif not Packet_Header.Packet_ID.Data_Field_Header_Flag'Valid then
    Tc_Error := Invalid_Data_Field_Header_Flag;
  elsif not ( Packet_Header.Packet_ID.Data_Field_Header_Flag = Tmtc.Packet.Field_Header_Present ) then
    Tc_Error := Invalid_Data_Field_Header_Flag;
  elsif not Packet_Header.Packet_ID.Application_Process_Id'Valid then
    Tc_Error := Invalid_Application_Process_Id;
  elsif not Packet_Header.Packet_Sequence_Control.Sequence_Flags'Valid then
    Tc_Error := Invalid.Sequence_Flags;
  elsif not Packet_Header.Packet_Sequence_Control.Sequence_Count'Valid then
    Tc_Error := Invalid.Sequence.Count;
  elsif not Packet_Header.PacketLength'Valid then
    Tc_Error := Invalid.Packet.Length;
  elsif not
```
Definition of data bus

```plaintext
type T_SADE_MOTOR_MOVEMENT_COMMAND is -- 5 BC-RT
record
  MOTOR1_TURNING_SENS : M1.Bits.T_1Bit; -- MOTOR1_DATA 7 1
  MOTOR1_SPEED_VALUE  : M1.Bits.T_5Bit; -- MOTOR1_DATA 11 5
  MOTOR2_TURNING_SENS : M1.Bits.T_1Bit; -- MOTOR2_DATA 23 1
  MOTOR2_SPEED_VALUE  : M1.Bits.T_5Bit; -- MOTOR2_DATA 27 5
  MOTOR3_TURNING_SENS : M1.Bits.T_1Bit; -- MOTOR3_DATA 39 1
  MOTOR3_SPEED_VALUE  : M1.Bits.T_5Bit; -- MOTOR3_DATA 43 5
  MOTOR4_TURNING_SENS : M1.Bits.T_1Bit; -- MOTOR4_DATA 55 1
  MOTOR4_SPEED_VALUE  : M1.Bits.T_5Bit; -- MOTOR4_DATA 59 5
end record;
for T_SADE_MOTOR_MOVEMENT_COMMAND use
record
  MOTOR1_TURNING_SENS at 0 range 7 .. 7; -- MOTOR1_DATA 7 1
  MOTOR1_SPEED_VALUE at 1 range 3 .. 7; -- MOTOR1_DATA 11 5
  MOTOR2_TURNING_SENS at 2 range 7 .. 7; -- MOTOR2_DATA 23 1
  MOTOR2_SPEED_VALUE at 3 range 3 .. 7; -- MOTOR2_DATA 27 5
  MOTOR3_TURNING_SENS at 4 range 7 .. 7; -- MOTOR3_DATA 39 1
  MOTOR3_SPEED_VALUE at 5 range 3 .. 7; -- MOTOR3_DATA 43 5
  MOTOR4_TURNING_SENS at 6 range 7 .. 7; -- MOTOR4_DATA 55 1
  MOTOR4_SPEED_VALUE at 7 range 3 .. 7; -- MOTOR4_DATA 59 5
end record;
```
Access to the data bus

```
-- Access to the bus frame with a Boolean result
procedure Read_Data_Pool_Boolean

  ( Parameter_Id : in  T_Parameter_Id;
  Read_Value   : out Boolean;
  Is_Ok        : out Boolean

  )

with

  Post =>

    ( Is_Ok = ( Parameter_Id = Id_SGS_MOTOR_POS_ACTI_0_POS_ACTI or else
               Parameter_Id = Id_SGS_MOTOR_POS_ACTI_1_POS_ACTI or else
               Parameter_Id = Id_SGS_MOTOR_POS_ACTI_2_POS_ACTI or else
               Parameter_Id = Id_SGS_MOTOR_POS_ACTI_3_POS_ACTI )

    )

  and then

    ( if Parameter_Id = Id_SGS_MOTOR_POS_ACTI_0_POS_ACTI then
      Read_Value = DP.Engineering_Data.SGS_MOTOR_POS_ACTI(0).POS_ACTI )

    and then

    ( if Parameter_Id = Id_SGS_MOTOR_POS_ACTI_1_POS_ACTI then
      Read_Value = DP.Engineering_Data.SGS_MOTOR_POS_ACTI(1).POS_ACTI )

    and then

    ( if Parameter_Id = Id_SGS_MOTOR_POS_ACTI_2_POS_ACTI then
      Read_Value = DP.Engineering_Data.SGS_MOTOR_POS_ACTI(2).POS_ACTI )

    and then

    ( if Parameter_Id = Id_SGS_MOTOR_POS_ACTI_3_POS_ACTI then
      Read_Value = DP.Engineering_Data.SGS_MOTOR_POS_ACTI(3).POS_ACTI )
```
Monitoring list

type I_Monitored_Parameter
  (Check_Type : T_Check_Type := Limit_Check;
   Parameter_Data_Type : Tmtc.Data_Pool.T_Parameter_Data_Type
    := Tmtc.Data_Pool.Nat32_Type ) is
  record
    -- The identification of the on-board parameter to be monitored
    Parameter_Id : Tmtc.Data_Pool.T_Parameter_Id;
    -- Is the monitoring of the parameter enabled or disabled
    Is_Enabled : Boolean;
    -- The monitoring interval for the parameter
    Monitoring_Interval : T_Monitoring_Interval;
    -- A counter allowing deciding if it is time to monitor the parameter
    -- (depending of the value of Monitoring_Interval)
    Monitoring_Counter : Ml.T_Nat32;
    -- Depending if the nature of the check to be performed
    case Check_Type is
    when Limit_Check =>
      case Parameter_Data_Type is
      when Tmtc.Data_Pool.Boolean_Type =>
        null;
      when Tmtc.Data_Pool.Nat32_Type =>
        Low_Limit_Nat32_Value : Ml.T_Nat32;
        High_Limit_Nat32_Value : Ml.T_Nat32;
      when Tmtc.Data_Pool.Int32_Type =>
        Low_Limit_Int32_Value : Ml.T_Int32;
        High_Limit_Int32_Value : Ml.T_Int32;
      when Tmtc.Data_Pool.Nat64_Type =>
        Low_Limit_Nat64_Value : Ml.T_Nat64;
        High_Limit_Nat64_Value : Ml.T_Nat64;
      when Tmtc.Data_Pool.Int64_Type =>
        Low_Limit_Int64_Value : Ml.T_Int64;
        High_Limit_Int64_Value : Ml.T_Int64;
Overview

- Introduction

- Astrium Space Transportation Case study
  - SCADE modelling
  - Data handling
  - Numerical algorithm
  - Event driven

- Alfa
- Gnatprove
- Gnattest
- Conclusion
Algorithms

- Orientation of the ATV solar wings
  - Optimisation of energy

- From SPARK to Alfa
function Sin32 (X : T_Float32) return T_Float32

with
    Pre => ( X >= -C_2Pi32 ) and then
            ( X <= C_2Pi32 ),
    Post => ( Sin32'Result >= -1.0 ) and then
            ( Sin32'Result <= 1.0 ),
Test_Case => (Name => "-2Pi .. -3Pi/2",
               Mode => Nominal,
               Requires => X >= -C_2Pi32 and X < -C_3Halfpi32,
               Ensures => Sin32'Result >= 0.0 and Sin32'Result <= 1.0),
Test_Case => (Name => "-3Pi/2 .. -Pi",
               Mode => Nominal,
               Requires => X >= -C_3Halfpi32 and X < -C_Pi,
               Ensures => Sin32'Result >= 0.0 and Sin32'Result <= 1.0),
Test_Case => (Name => "-Pi .. -Pi/2",
               Mode => Nominal,
               Requires => X >= -C_Pi32 and X < -C_HalfPi,
               Ensures => Sin32'Result >= -1.0 and Sin32'Result <= 0.0),
Test_Case => (Name => "-Pi/2 .. 0",
               Mode => Nominal,
               Requires => X >= -C_HalfPi32 and X < 0.0.
From SPARK to Ada 2012

```
procedure S_CONV_AND_INCREMENT_ANGLE (INPUT_XSUN : in M1.T_FLOAT32;
                                      Angle_Radian : in M1.T_Angle_HalfPi;
                                      Angle_Degree : out M1.T_Angle_360)

  --# derives Angle_Degree from Angle_Radian,
  --#
  --#   INPUT_XSUN;
  --# pre ( Angle_Radian >= -M1.C_HalfPi32 ) and
  --#
  --#   ( Angle_Radian <= M1.C_HalfPi32 );
with Pre =>
  ( Angle_Radian >= -M1.C_HalfPi32 ) and then
  ( Angle_Radian <= M1.C_HalfPi32 );
```

```
function S_COMP_MOTORJ_THETA_POINT_COM_M (SADM_J : in T_SGS_SADM_ID;
                                          Theta_Point_Com : in T_MOTOR_THETA_POINT_Com)
  return T_MOTOR_THETA_POINT_Com

  --# return MOTJ_THETA_POINT_COM_M => ( ( MOTJ_THETA_POINT_COM_M >= -0.5 ) and
  --#
  --#   ( MOTJ_THETA_POINT_COM_M <= 0.5 ) );
with Post =>
  ( S_COMP_MOTORJ_THETA_POINT_COM_M!Result >= -0.5 ) and
  ( S_COMP_MOTORJ_THETA_POINT_COM_M!Result <= 0.5 );
```
Overview

- Introduction
- Astrium Space Transportation Case study
  - SCADE modelling
  - Data handling
  - Numerical algorithm
  - Event driven
- Alfa
- Gnatprove
- Gnattest
- Conclusion
5.2.8 Waits

a. It shall be possible to specify the following types of wait statement within the main body of an OBCP:
   1. wait until a given on-board time;
   2. wait for a given interval of time to elapse;
   3. wait until a given condition becomes true;
   4. wait until a given event occurs;
   5. wait until any event from a specified list of events occurs.

b. It shall be possible to specify a combination of conditions within a wait statement.

c. The conditions specified for an OBCP precondition or confirmation shall be any combination of the following:
   1. wait until a given absolute time;
   2. wait for a given interval of time to elapse;
   3. wait until a given condition becomes true;
   4. wait until a given event occurs;
   5. wait until any event from a specified list of events occurs;
   6. test whether a given condition is true.

d. It shall be possible to define timeout conditions for wait statements and the behaviour if the timeout is exceeded.
Features

- Events
  - Detection, timed windows

- Variables

- Expressions
  - +, -, comparisons…

- Automated procedure

- Onboard mission plan
  - Loop, conditions…

- (Fault Detection) Isolation and Recovery
Example of contract

Example:
✓ A list of event detection statuses
✓ Request to reset the detection status for Event

procedure Reset_Event_Status (Event : in T_Event) with Post =>

Post-condition
not Event_Status(Event).Detection and
(for all Other_Event in T_Event =>
( if Other_Event /= Event then
    Event_Status( Other_Event ) = Event_Status'Old( Other_Event ) ));

The detection status is unchanged

<table>
<thead>
<tr>
<th>Event1</th>
<th>Event</th>
<th>Event3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not detected</td>
<td>Not detected</td>
<td>Detected</td>
</tr>
</tbody>
</table>
Generic

1. generic
type T_Event_Id is <>;

2. package MvM_Events is

   -- The type defining the number of times an event can be detected
   subtype T_Number_Of_Detection is Ml.T_Nat64;

   type T_Type_Of_Monitoring is
   (No_Window,
    Time_Window,
    Protected_Window);

   type T_Type_Of_Time is
   (Absolute_Time,
    Relative_To_Event);

   -- The detection status of an event
   type T_Event_Status is
   record
     Is_Active : Ml.T_Boolean := False;
     Type_Of_Monitoring : T_Type_Of_Monitoring := No_Window;
     Type_Of_Time : T_Type_Of_Time := Absolute_Time;
   end record

   case Is_Active is
     when False =>
       null;
     when True =>
       -- The number of times the event has been detected
       -- in the time window
       (i.e., between Start_Of_Monitoring and End_Of_Monitoring)
Contract can be quite big
Overview

- Introduction
- Astrium Space Transportation Case study
  - SCADE modelling
  - Data handling
  - Numerical algorithm
  - Event driven
- Alfa
- Gnatprove
- Gnattest
- Conclusion
Features not yet supported

tmtc-types.ads:35:04: info: enumeration representation clause is not yet supported
...
tmtc-data_pool.adb:394:24: info: attribute is not yet supported
...
tmtc.ads:34:09: info: discriminant is not yet supported
...
ml-bits.adb:25004:44: info: type conversion not between scalar types is not yet supported
...
ml.adb:688:04: warning: in instantiation at a-ngelfu.adb:943
...
sgs-basic_types.ads:108:09: info: array type is not yet supported
...
Subprograms in Alfa : 92% (618/669)
   ... already supported : 88% (586/669)
   ... not yet supported : 5% (32/669)
Subprograms not in Alfa : 8% (51/669)

Subprograms not in Alfa due to (possibly more than one reason):
   exception : 6% (40/669)
   unchecked conversion : 1% (10/669)
   impure function : 0% (1/669)

Subprograms not yet supported due to (possibly more than one reason):
   attribute : 6% (39/669)
   conversion : 2% (11/669)
   multi dim array : 1% (4/669)
   discriminant : 0% (3/669)

Units with the largest number of subprograms in Alfa:
   ml-bits : 98% (388/396)
   ml : 78% (140/180)
   tmtc-data_pool : 96% (47/49)
   sgs-main : 100% (14/14)
   scade-ln1 : 100% (11/11)
   scade-p5_sgs_lib_types-im : 100% (3/3)
(...)

Units with the largest number of subprograms not in Alfa:
   ml : 22% (40/180)
   ml-bits : 2% (8/396)
   tmtc-data_pool : 4% (2/49)
   util-time : 100% (1/1)
Compatibility of the case study with Alfa

- Two main features
  - Generic: Implemented
  - Discriminant: To be implemented

- + some other minor features not yet supported
Visibility problematic

Visibility strategy and proof

```
with M1;
use type M1.T_Nat54;

package Mvm.Obit is

  subtype T_Numerical_Obit is M1.T_Nat54;

  type T_Obit is private;

  -- Return the current OBIT
  function Get(Obit : in T_Obit) return T_Numerical_Obit;

  -- Make the time running, i.e. increment the OBIT
  procedure Run_Time(Obit : in out T_Obit) with
    Post => Get(Obit) := Get(Obit'Old),
    Test_Case => (Name => "Test of Run_Time",
                   Mode => Nominal,
                   Requires => Get(Obit) < T_Numerical_Obit'Last,
                   Ensures => Get(Obit) = Get(Obit'Old) +1),
    Test_Case => (Name => "Test of Run_Time with overflow",
                   Mode => Nominal,
                   Requires => Get(Obit) = T_Numerical_Obit'Last,
                   Ensures => Get(Obit) = Get(Obit'Old));

private

  type T_Obit is
    record
      Hidden_Obit : T_Numerical_Obit := 0;
    end record;

  -- Return the current OBIT
  function Get(Obit : in T_Obit) return T_Numerical_Obit is
    (Obit.Hidden_Obit);
  end Mvm.Obit;
```

Reading right without writing right

Use in proof

Encapsulated private value
Imprecision of the localisation

Very efficient executable contracts
Overview

- Introduction
- Astrium Space Transportation Case study
  - SCADE modelling
  - Data handling
  - Numerical algorithm
  - Event driven
- Alfa
- Gnatprove
- Gnattest
- Conclusion
function Sign_Strict_Int32 (X : in T_Int32) return T_Sign_Int32
with
Post -> ( ( Sign_Strict_Int32'Result = 1 ) or
        ( Sign_Strict_Int32'Result = -1 ) ) and then
       (if X >= 0 then Sign_Strict_Int32'Result = 1) and then
       (if X < 0 then Sign_Strict_Int32'Result = -1),
Test_Case => (Name => "Negative input",
              Mode => Nominal,
              Requires => X < 0,
              Ensures => Sign_Strict_Int32'Result = -1),
Test_Case => (Name => "Zero input",
              Mode => Nominal,
              Requires => X = 0,
              Ensures => Sign_Strict_Int32'Result = 1),
Test_Case => (Name => "Positive input",
              Mode => Nominal,
              Requires => X > 0,
              Ensures => Sign_Strict_Int32'Result = 1):
end X1;
function Sign_Strict_Int32 (X : in T_Inf32) return T_Sign_Int32
is
  Result : T_Sign_Int32;
begin
  if (X > 0) then
    Result := 1;
  elsif |X < 0| then
    Result := -1;
  end if;
  return Result;
end Sign_Strict_Int32;
function Sign_Strict_Int32 (X : in T_Int32) return T_Sign_Int32 is
  Result : T_Sign_Int32;
begin
  if (X > 0) then
    Result := 1;
  elsif |X < 0| then
    Result := -1;
  end if;
  return Result;
end Sign_Strict_Int32;
Function Sign_Strict_Int32 (X : in T_Int32) return T_Sign_Int32
with
  Post -> | (Sign_Strict_Int32'Result = 1) or
           (Sign_Strict_Int32'Result = 0) and then
           (if X >= 0 then Sign_Strict_Int32'Result = 1) and then
           (if X < 0 then Sign_Strict_Int32'Result = -1),
  Test_Case => (Name => "Negative input",
               Node => Nominal,
               Ensures => Sign_Strict_Int32'Result = -1),
  Test_Case => (Name => "Zero input",
               Node => Nominal,
               Requires => X = 0,
               Ensures => Sign_Strict_Int32'Result = 1),
  Test_Case => (Name => "Positive input",
               Node => Nominal,
               Requires => X > 0,
               Ensures => Sign_Strict_Int32'Result = 1);
function Sign_Strict_Int32 (X : in T_Int32) return T_Sign_Int32
is
  Result : T_Sign_Int32;
begin
  if (X >= 0) then
    Result := 1;
  else
    Result := -1;
  end if;
  return Result;
end Sign_Strict_Int32;
function Sign_Strict_Int32 (X : in T_Int32) return T_Sign_Int32
is
    Result : T_Sign_Int32;
begin
    if (X >= 0) then
        Result := 1;
    else
        Result := -1;
    end if;
    return Result;
end Sign_Strict_Int32;
Use of gnatprove – approach

- Use on the complete case study
  - Not yet supported
    (e.g. discriminant)

→ Use on a restricted “SPARK” case study
Four dimension arrays not accepted
Adaptation of the case study
Subprograms in / not in Alfa
(restricted case study)

Subprograms in Alfa : 76% (124/164)
  ... already supported : 74% (122/164)
  ... not yet supported : 1% ( 2/164)
Subprograms not in Alfa : 24% ( 40/164)

Subprograms not in Alfa due to (possibly more than one reason):
  exception : 24% ( 40/164)

Subprograms not yet supported due to (possibly more than one reason):
  attribute : 11% ( 18/164)

Units with the largest number of subprograms in Alfa:
  ml : 73% (110/150)
  sgs : 100% (14/14)

Units with the largest number of subprograms not in Alfa:
  ml : 27% (40/150)
Application subprograms in Alfa

++ sgs__s_normalise_0_360 sgs.ads:636
++ sgs__s_normalise_minus_plus_180 sgs.ads:645
++ sgs__s_conv_and_increment_angle sgs.ads:659
++ sgs__s_comp_motorj_theta_point_com sgs.ads:672
++ sgs__s_comp_motorj_theta_point_com_m sgs.ads:694
++ sgs__s_store_theta_point_com_step sgs.ads:705
++ sgs__s_rotate_reed_switch_sequences sgs.ads:718
++ sgs__s_are_cms_switch_sts_available sgs.ads:729
++ sgs__s_compute_reed_switch_angle sgs.ads:749
++ sgs__s_rotate_data sgs.ads:795
++ sgs__s_sgs_b6_elab_sol_wing_cmd_angle sgs.ads:234
++ sgs__estimate_solar_wing_j_position sgs.ads:808
++ sgs__commanded_velocity_calculation sgs.ads:830
++ sgs__s_sgs_b1_comp_sol_wing_ang_veloc sgs.ads:259
Some library subprograms bodies not in Alfa

- ml__num32__expX a-ngelfu.ads:60 instantiated at ml.adb:121 (exception)
- ml__num32__arccosX a-ngelfu.ads:108 instantiated at ml.adb:121 (exception)
- ml__num32__arccos__2X a-ngelfu.ads:112 instantiated at ml.adb:121 (exception)
- ml__num32__arccoshX a-ngelfu.ads:168 instantiated at ml.adb:121 (exception)
- ml__num32__arccothX a-ngelfu.ads:177 instantiated at ml.adb:121 (exception)
- ml__num32__arcsinX a-ngelfu.ads:100 instantiated at ml.adb:121 (exception)
- ml__num32__arcsin__2X a-ngelfu.ads:104 instantiated at ml.adb:121 (exception)
- ml__num32__arctanX a-ngelfu.ads:116 instantiated at ml.adb:121 (exception) [attribute]
- ml__num32__arctan__2X a-ngelfu.ads:123 instantiated at ml.adb:121 (exception) [attribute]
- ml__num32__arctanhX a-ngelfu.ads:173 instantiated at ml.adb:121 (exception) [attribute]
- ml__num32__cotX a-ngelfu.ads:96 instantiated at ml.adb:121 (exception)
- ml__num32__cot__2X a-ngelfu.ads:98 instantiated at ml.adb:121 (exception) [attribute]
- ml__num32__cothX a-ngelfu.ads:160 instantiated at ml.adb:121 (exception)
- ml__num32__expX a-ngelfu.ads:56 instantiated at ml.adb:121 (exception) [attribute]
- ml__num32__exp_strictX a-ngelfu.adb:66 instantiated at ml.adb:121 (exception) [attribute]
* ml__num32__local_atanX a-ngelfu.adb:70 instantiated at ml.adb:121 [attribute]
- ml__num32__logX a-ngelfu.ads:48 instantiated at ml.adb:121 (exception)
- ml__num32__log__2X a-ngelfu.ads:52 instantiated at ml.adb:121 (exception)
- ml__num32__sin__2X a-ngelfu.ads:73 instantiated at ml.adb:121 (exception) [attribute]
- ml__num32__sqrtX a-ngelfu.ads:42 instantiated at ml.adb:121 (exception)
- ml__num32__tan__2X a-ngelfu.ads:92 instantiated at ml.adb:121 (exception) [attribute]
- ml__num64__expX a-ngelfu.ads:60 instantiated at ml.adb:688 (exception)
- ml__num64__arccosX a-ngelfu.ads:108 instantiated at ml.adb:688 (exception)
+ ml__num64__arccos__2X a-ngelfu.ads:112 instantiated at ml.adb:688 (exception)
10/05/2011 p51

10) Reference SADM switch 1 status of the SADM: it is available if it is TRUE

14) Reference SADM switch 2 status of the SADM: it is available if it is TRUE

is

-- Returned value
REPORT_IS_OK : BOOLEAN;
--------------------- CODE ---------------------

begin

-- Return the logical "AND" between all data:
-- 1) CMS validity flag: it is available if it is OK
REPORT_IS_OK := (CMS_VALIDITY_FLAG, CMS_CHU1 VALIDITY_FLAG = OK) and then

-- 2) Canonic wing reed switch 1 status of the SADM: it is available if it is I
(WING_SWITCH_CAMO_STS_RPT_OK(SADM J, SGS_SWITCH 1)) and then

-- 3) Canonic wing reed switch 2 status of the SADM: it is available if it is I
(WING_SWITCH_CAMO_STS_RPT_OK(SADM J, SGS_SWITCH 2)) and then

Phase 3 of 3: generation and proof of VCs ...

Exception raise in Session.idle_handler: anomaly: Out of memory
Session.idle_handler stopped

<table>
<thead>
<tr>
<th>Nom de l'image</th>
<th>Nom de l'utilisateur</th>
<th>Processeur</th>
<th>Utilisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>gnawhy3.exe</td>
<td>M023206</td>
<td>50</td>
<td>2 050 844 Ko</td>
</tr>
<tr>
<td>EngineServer.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>130 552 Ko</td>
</tr>
<tr>
<td>gpt.exe</td>
<td>M023206</td>
<td>00</td>
<td>113 260 Ko</td>
</tr>
<tr>
<td>explorer.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>102 608 Ko</td>
</tr>
<tr>
<td>Acrobat.exe</td>
<td>M023206</td>
<td>00</td>
<td>91 056 Ko</td>
</tr>
<tr>
<td>Msihosts.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>53 360 Ko</td>
</tr>
<tr>
<td>services.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>50 284 Ko</td>
</tr>
<tr>
<td>spoolsv.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>42 316 Ko</td>
</tr>
<tr>
<td>svchost.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>39 440 Ko</td>
</tr>
<tr>
<td>svchost.exe</td>
<td>SERVICE RESEAU</td>
<td>00</td>
<td>34 692 Ko</td>
</tr>
<tr>
<td>PPNVCOM.EXE</td>
<td>M023206</td>
<td>00</td>
<td>32 616 Ko</td>
</tr>
<tr>
<td>svchost.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>32 088 Ko</td>
</tr>
<tr>
<td>WINWORD.EXE</td>
<td>M023206</td>
<td>00</td>
<td>31 700 Ko</td>
</tr>
<tr>
<td>CmExe.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>29 672 Ko</td>
</tr>
<tr>
<td>svchost.exe</td>
<td>SERVICE LOCAL</td>
<td>00</td>
<td>29 336 Ko</td>
</tr>
<tr>
<td>POWERPNT.EXE</td>
<td>M023206</td>
<td>00</td>
<td>25 944 Ko</td>
</tr>
<tr>
<td>svchost.exe</td>
<td>SERVICE RESEAU</td>
<td>00</td>
<td>21 308 Ko</td>
</tr>
<tr>
<td>PolySpaceInOneClick.exe</td>
<td>M023206</td>
<td>00</td>
<td>15 656 Ko</td>
</tr>
<tr>
<td>OUTLOOK.EXE</td>
<td>M023206</td>
<td>00</td>
<td>13 912 Ko</td>
</tr>
<tr>
<td>FrameworkService</td>
<td>SYSTEM</td>
<td>00</td>
<td>11 416 Ko</td>
</tr>
<tr>
<td>xemacc.exe</td>
<td>M023206</td>
<td>00</td>
<td>9 888 Ko</td>
</tr>
<tr>
<td>Acrobat.exe</td>
<td>M023206</td>
<td>00</td>
<td>9 708 Ko</td>
</tr>
<tr>
<td>EXCEL.EXE</td>
<td>M023206</td>
<td>00</td>
<td>9 140 Ko</td>
</tr>
<tr>
<td>winlogon.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>8 292 Ko</td>
</tr>
<tr>
<td>atask.exe</td>
<td>M023206</td>
<td>00</td>
<td>7 744 Ko</td>
</tr>
<tr>
<td>SBSRSY.EXE</td>
<td>SYSTEM</td>
<td>00</td>
<td>7 600 Ko</td>
</tr>
<tr>
<td>winprvse.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>7 284 Ko</td>
</tr>
<tr>
<td>gptbuild.exe</td>
<td>M023206</td>
<td>00</td>
<td>6 856 Ko</td>
</tr>
<tr>
<td>dlmhost.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>5 696 Ko</td>
</tr>
<tr>
<td>winprvse.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>5 928 Ko</td>
</tr>
<tr>
<td>gnawhy.exe</td>
<td>M023206</td>
<td>00</td>
<td>5 572 Ko</td>
</tr>
<tr>
<td>iss_ecc.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>5 476 Ko</td>
</tr>
<tr>
<td>smss.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>5 032 Ko</td>
</tr>
<tr>
<td>msiexec.exe</td>
<td>M023206</td>
<td>00</td>
<td>4 928 Ko</td>
</tr>
<tr>
<td>Sbtlrun.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>4 888 Ko</td>
</tr>
<tr>
<td>bash.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>4 476 Ko</td>
</tr>
<tr>
<td>info.exe</td>
<td>M023206</td>
<td>00</td>
<td>4 116 Ko</td>
</tr>
</tbody>
</table>

Processus : 67  UC utilisée : 72%  Charge dédiée : 3347 Mo /
Overview

- Introduction
- Astrium Space Transportation Case study
  - SCADE modelling
  - Data handling
  - Numerical algorithm
  - Event driven
- Alfa
- Gnatprove
- Gnattest
- Conclusion
function Sign_Strict_Int32 (X : in T_Int32) return T_Sign_Int32 with
Post => ( ( Sign_Strict_Int32'Result = 1 ) or
( Sign_Strict_Int32'Result = -1 ) ) and then
(if X >= 0 then Sign_Strict_Int32'Result = 1) and then
(if X < 0 then Sign_Strict_Int32'Result = -1),
Test_Case => (Name => "Negative input",
Mode => Nominal,
Requires => X < 0,
Ensures => Sign_Strict_Int32'Result = -1),
Test_Case => (Name => "Zero input",
Mode => Nominal,
Requires => X = 0,
Ensures => Sign_Strict_Int32'Result = -1),
Test_Case => (Name => "Positive input",
Mode => Nominal,
Requires => X > 0,
Ensures => Sign_Strict_Int32'Result = 1);


```ada
10 return T_Sign_Int32
20 \  X \leq 0 \rightarrow \text{Sign}_\text{Strict}_\text{Int32}(\text{Result} = -1)
30 \  X > 0 \rightarrow \text{Sign}_\text{Strict}_\text{Int32}(\text{Result} = 1)
40 \end_of_MI;
```
function Sign_Strict_Int32 (X : in T_Int32) return T_Sign_Int32 with
  Post => \( \text{(Sign\_Strict\_Int32}'\text{Result} = 1) \text{ or} \ \text{(Sign\_Strict\_Int32}'\text{Result} = -1) \) and then
  (if \( X \geq 0 \) then Sign\_Strict\_Int32}'\text{Result} = 1) and then
  (if \( X < 0 \) then Sign\_Strict\_Int32}'\text{Result} = -1),
  Test_Case => (Name => "Negative input",
    Mode => Nominal,
    Requires => X < 0,
    Ensures => Sign\_Strict\_Int32}'\text{Result} = -1),
  Test_Case => (Name => "Zero input",
    Mode => Nominal,
    Requires => X = 0,
    Ensures => Sign\_Strict\_Int32}'\text{Result} = -1),
  Test_Case => (Name => "Positive input",
    Mode => Nominal,
    Requires => X > 0,
    Ensures => Sign\_Strict\_Int32}'\text{Result} = 1);
with Gnattest_Generated;

separate (M1.Test_Data.Tests)

procedure Test_Sign_Strict_Int32_9F4b45_0052bd (T : in out Test) is
  pragma Unreferenced (T);
  function Sign_Strict_Int32 (X : in T_Int32) return T_Sign_Int32 renames
    Wrap_Test_Sign_Strict_Int32_9F4b45_0052bd;
begin
  AllUnit.Assertions.Assert
    (Gnattest_Generated.Default_Assert_Value,
     "Test not implemented.");
end Test_Sign_Strict_Int32_9F4b45_0052bd;
with Gnat тест_Generated;

procedure Test_Sign_Strict_32_9f4b45_0052bd (T : in out Test) is
  pragma Unreferenced (T);
begin
  All UIT.Assertions.Assert
    (Gnat тест_Generated.Default_Assert_Value,
     "Test not implemented.");
end Test_Sign_Strict_32_9f4b45_0052bd;
10/05/2011 p61

1 -- function Sign_Strict_Int32 (X : in T_Int32) return T_Sign_Int32
2  -- Test Case "Zero input"
3
4 with Gnatest_Generated;
5
6 separate (M1.Test_Data.Tests)
7 procedure Test_Sign_Strict_Int32_9f4b45_0052bd (T : in out Test) is
8    pragma Unreferenced (T);
9  function Sign_Strict_Int32 (X : in T_Int32) return T_Sign_Int32 renames Wrap_Test_Sign_Strict_Int32_9f4b45_0052bd;
10 begin
11    Assert( Sign_Strict_Int32(X => 0) = 1, "Zero input" );
12 end Test_Sign_Strict_Int32_9f4b45_0052bd;

[2012-05-03 14:30:20] process terminated successfully (elapsed time: 00.45s)
1  -- function Sign_Strict_Int32 (X : in T_Int32) return T_Sign_Int32
2  -- Test Case "Zero input"
3  with Gnatest_Generated;
4
5  separate (M1.Test_Data.Tests)
6  procedure Test_Sign_Strict_Int32_9f4b45_0052bd (T : in out Test) is
7      pragma Unreferenced (T);
8      function Sign_Strict_Int32 (X : in T_Int32) return T_Sign_Int32 renames Wrap_Test_Sign_Strict_Int32_9f4b45_0052bd;
9  begin
10     Assert ( Sign_Strict_Int32(X => 0) = 1, "Zero input" );
11  end Test_Sign_Strict_Int32_9f4b45_0052bd;
12
13 Test_Sign_Strict_Int32_9f4b45_0052bd
14
15 FAIL M1.Test_Data.Tests : Test_Sign_Strict_Int32_9f4b45_0f8e1e
16     Test not implemented.
17     at ml-test_data-tests-test_sign_strict_int32_9f4b45_0f8e1e.adb:11
18 FAIL M1.Test_Data.Tests : Test_Sign_Strict_Int32_9f4b45_4f40
19     Test not implemented.
20     at ml-test_data-tests-test_sign_strict_int32_9f4b45_4f40.adb:11
21
22 Total Tests Run:   3
23 Successful Tests:  1
24 Failed Assertions: 2
25 Unexpected Errors: 0
26 [2012-05-03 14:38:31] process terminated successfully (elapsed time: 0.86s)
Positive points

- **Easiness**
  - Difficulty to manually write a test harness from scratch
  - Verification that the test case is satisfied
Points to be improved

- One file per test case
  - Huge number of files
    (one file is generally defined per package)

- Naming of files
File name not linked to the test case
Addition of a new test case
Previous file not impacted
Suppression of a test case
The project contains no scenario variables

Project

Tmc
.
..\OBJ

Test Case

Test Case

end M1:

end M1:

[/image]
Overview

- Introduction
- Astrium Space Transportation Case study
  - SCADE modelling
  - Data handling
  - Numerical algorithm
  - Event driven
- Alfa
- Gnatprove
- Gnattest
- Conclusion
Conclusion

A lot of progress on all the subjects

- **Alfa**
  - A lot of features already implemented
  - A few are missing for the Astrium case study

- **gnatprove**
  - Performance issue

- **gnatattest**
  - Very practical for small projects
  - Process issue for (normally) big projects

- **Case study**
  - Good process
  - Assessment of each new version of the tools
Always a great support from AdaCore