modelling and automatic **code** generation

Where are we? Where are we going?

Maxime Perrotin and Celia Yabar
Software Engineering division – ESTEC TEC SWE
Control Systems division
ESTEC TEC-ECN

Presented at ADCSS'2014

Back in 2000…

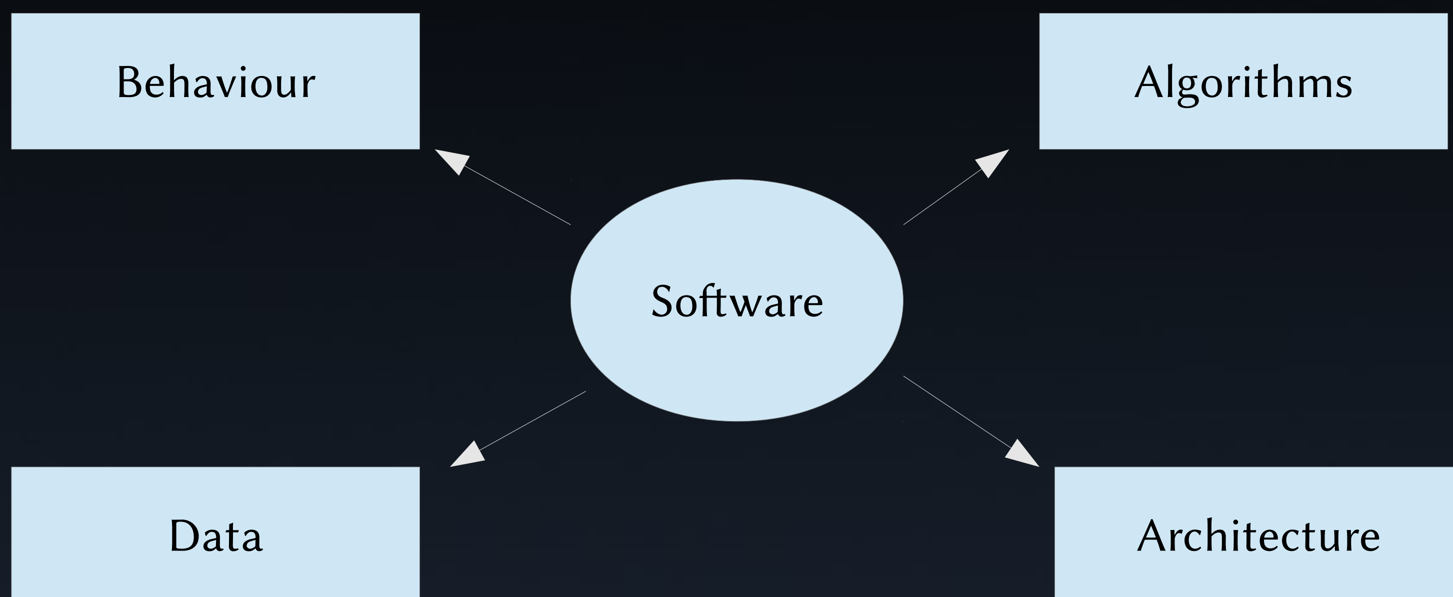
- The space industry had recently moved from Assembly to Ada
- The first satellite specification and software based on modelling and automatic code generation was in preparation (Smart-1, launched in 2002)
- Tools were already advanced – and promising for the future
  - ObjectGeode
  - Tau
  - Statemate
  - StP
  - ObjecTime
  - SCADE
  - Matlab/Simulink
- Where are we **now** ?

Use models but why ?

- In general bugs can come from
  - Bad specifications
  - Inefficient design
  - Bad programmers
  - Bad programming langages
  - Bad process
  - ...Or a little bit of all the points above

The main challenge

- Improve the production cycle of software



Software modelling

- GNC/AOCS engineers gave a strong impulse
- Matlab is good for designing control laws
- From simulation to code generation : a single step ?
- Already flying on several missions
- Several companies are ready to adopt the approach on a large scale
- ESA has an internal working group on the topic

Algorithms

Modelling with Matlab/Simulink

- Cannot be done blindly – a process must support the approach
  - Use cases : simulators, flight software, non-critical ground software, research on algorithms
  - Who does what and when ? Software and GNC people must work hand in hand !
  - Simulink blocks or Embedded Matlab ?
  - How does it fit with ECSS standards ?
- Which code generator : RTW/EC or **Qgen** ?
- Several successful case studies (USACDF, Vega…) done by GNC teams at ESA with the support of the software department

Working group on ESA Standard for Modeling with the MATLAB and Simulink Product Family

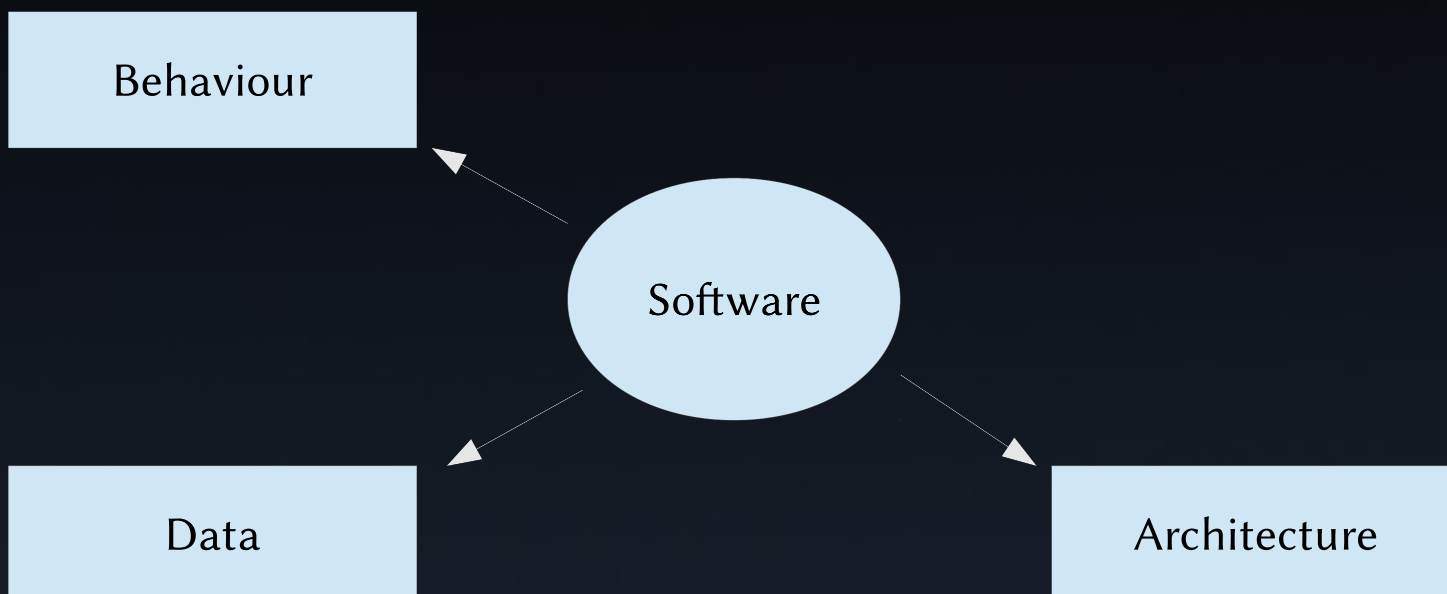
- Purpose: to define an official ESA standard to apply when creating models and code using the MATLAB and Simulink product family
- The objective is to help producing code that is correct, readable, sharable/reuseable, and maintainable
- The first outcome of this working group is a White Paper.

Working group activities

- Study the state of the art of the standards and guidelines already available
- Review past ESA projects using Mathworks products and identify potential areas of improvement in the current design approach which could be solved by following standards and guidelines.
- Define the types of standards and guidelines needed for both Matlab and Simulink
- Study the existing tools to automatically check the standards and guidelines such as the Model Advisor or M-lint
- Prepare the table of content of the ESA Standard for Modeling with the MATLAB and Simulink product family

Simulink does not cover everything (yet?)

- Do we have tools to improve the rest of the onboard software ? ... and do they really help ?



Data Modelling

- Model TM and TC structures and ensure consistency at system-level
- Automatically generate
  - C and Ada native data types
  - Data encoders and decoders
  - Interface Control Documents
  - Automatic test cases
  - Database entries

Data

Example

```
-------------------------------
-- General Telecommand structure
-------------------------------

T-telecommand ::= SEQUENCE
{
    packet-header       TC-packetHeader,
    data-field-header   T-tc-dataFieldHeader,
    application-data    T-tc-applicationData,
    crc                 T-uint16
}
```

```
------------------------------
-- Telecommand application data
------------------------------

-- List of all available TCs categorized by their respective pus(-sub)types
-- Definition of actual payload data is done in respective Types below
-- In the ACN-file this type is used to automatically assign the pustype and subtype fields
-- in encoding and determine the packet type from pustype and subtype in decoding
-- Types defined as T-NULL have no actual payload data besides the fields
-- for pustype and subtype.
T-tc-applicationData ::= CHOICE
{
  tc-3-27-update-hk-period   TC-UPDATE-HK-PERIOD,
  tc-6-2-load-memory         TC-LOAD-MEMORY,
  tc-6-5-dump-memory         TC-DUMP-MEMORY,
  tc-6-9-check-memory        TC-CHECK-MEMORY,
  tc-6-129-transfer-image    TC-TRANSFER-IMAGE,
  tc-210-3-reset-dpu         T-NULL,              -- T-NULL is for TCs which don't have any applicationData
  tc-210-4-enable-watchdog   T-NULL,                     -- but only service type and subtype. Still they have to
  tc-210-5-disable-watchdog  T-NULL,              -- Appear in the list of valid commands. T-NULL ensures that 0 bits will be encoded
  tc-210-6-boot-iasw         TC-BOOT-IASW,
  tc-197-2-report-boot       T-NULL
}
```

Add a few directives to instruct tools to respect the PUS standard

```
-- Table which maps the pusType and subtype to the corresponding
-- packet payload data
T-tc-applicationData<T-uint8:pusType, T-uint8:pusSubType> []
{
    tc-3-27-update-hk-period   [present-when pusType==  3 pusSubType== 27 ],
    tc-6-2-load-memory         [present-when pusType==  6 pusSubType==  2 ],
    tc-6-5-dump-memory         [present-when pusType==  6 pusSubType==  5 ],
    tc-6-9-check-memory        [present-when pusType==  6 pusSubType==  9 ],
    tc-6-129-transfer-image    [present-when pusType==  6 pusSubType==129 ],
    tc-210-3-reset-dpu         [present-when pusType==210 pusSubType==  3 ],
    tc-210-4-enable-watchdog   [present-when pusType==210 pusSubType==  4 ],
    tc-210-5-disable-watchdog  [present-when pusType==210 pusSubType==  5 ],
    tc-210-6-boot-iasw         [present-when pusType==210 pusSubType==  6 ],
    tc-197-2-report-boot       [present-when pusType==197 pusSubType==  2 ]
}
```

Get this ICD for free...

T-tc-applicationData(CHOICE) ASN.1 ACN	min = 0 bytes	max = 1010 bytes

==============================
Telecommand application data
==============================
List of all available TCs categorized by their respective pus(-sub)types
Definition of actual payload data is done in respective Types below
In the ACN-file this type is used to automatically assign the pustype and subtype fields
in encoding and determine the packet type from pustype and subtype in decoding
Types defined as T-NULL have no actual payload data besides the fields
for pustype and subtype.

No	ACN Parameters what is this?	Type
1	pusType	T-uint8
2	pusSubType	T-uint8

No	Field	Comment	Present	Type	Constraint	Min Length (bits)	Max Length (bits)
1	tc-3-27-update-hk-period		pusType=3 AND pusSubType=27	TC-UPDATE-HK-PERIOD	N.A.	32	32
2	tc-6-2-load-memory		pusType=6 AND pusSubType=2	TC-LOAD-MEMORY	N.A.	112	8080
3	tc-6-5-dump-memory		pusType=6 AND pusSubType=5	TC-DUMP-MEMORY	N.A.	80	80
4	tc-6-9-check-memory		pusType=6 AND pusSubType=9	TC-CHECK-MEMORY	N.A.	72	72
5	tc-6-129-transfer-image		pusType=6 AND pusSubType=129	TC-TRANSFER-IMAGE	N.A.	80	80
6	tc-210-3-reset-dpu		pusType=210 AND pusSubType=3	T-NULL	N.A.	0	0
7	tc-210-4-enable-watchdog		pusType=210 AND pusSubType=4	T-NULL	N.A.	0	0
8	tc-210-5-disable-watchdog		pusType=210 AND pusSubType=5	T-NULL	N.A.	0	0
9	tc-210-6-boot-iasw		pusType=210 AND pusSubType=6	TC-BOOT-IASW	N.A.	80	80
10	tc-197-2-report-boot		pusType=197 AND pusSubType=2	T-NULL	N.A.	0	0

And this code...and much more

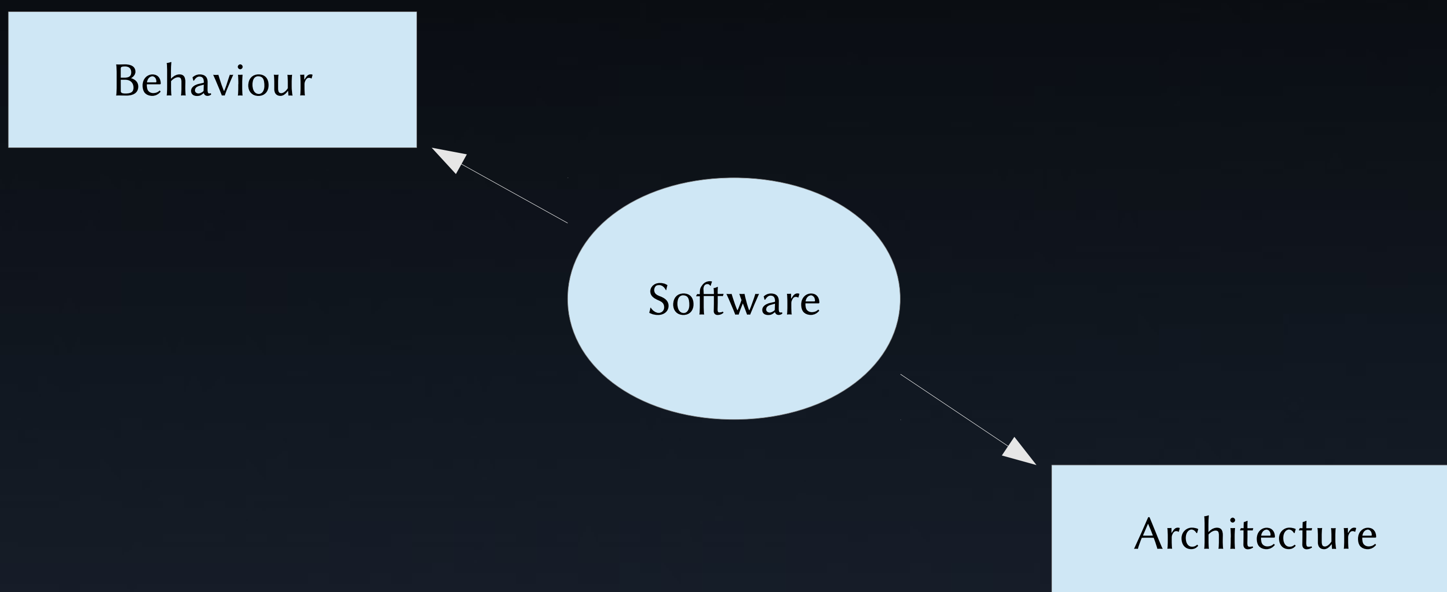
```
typedef struct {
    enum {
        T_tc_applicationData_NONE,
        tc_3_27_update_hk_period_PRESENT,
        tc_6_2_load_memory_PRESENT,
        tc_6_5_dump_memory_PRESENT,
        tc_6_9_check_memory_PRESENT,
        tc_6_129_transfer_image_PRESENT,
        tc_210_3_reset_dpu_PRESENT,
        tc_210_4_enable_watchdog_PRESENT,
        tc_210_5_disable_watchdog_PRESENT,
        tc_210_6_boot_iasw_PRESENT,
        tc_197_2_report_boot_PRESENT
    } kind;
    union {
        TC_UPDATE_HK_PERIOD tc_3_27_update_hk_period;
        TC_LOAD_MEMORY tc_6_2_load_memory;
        TC_DUMP_MEMORY tc_6_5_dump_memory;
        TC_CHECK_MEMORY tc_6_9_check_memory;
        TC_TRANSFER_IMAGE tc_6_129_transfer_image;
        T_NULL tc_210_3_reset_dpu;
        T_NULL tc_210_4_enable_watchdog;
        T_NULL tc_210_5_disable_watchdog;
        TC_BOOT_IASW tc_210_6_boot_iasw;
        T_NULL tc_197_2_report_boot;
    } u;
} T_tc_applicationData;

#define T_tc_applicationData_REQUIRED_BYTES_FOR_ENCODING       1007
#define T_tc_applicationData_REQUIRED_BITS_FOR_ENCODING        8049
#define T_tc_applicationData_REQUIRED_BYTES_FOR_ACN_ENCODING   1010
#define T_tc_applicationData_REQUIRED_BITS_FOR_ACN_ENCODING    8080
#define T_tc_applicationData_REQUIRED_BYTES_FOR_XER_ENCODING   2272

void T_tc_applicationData_Initialize(T_tc_applicationData* pVal);
flag T_tc_applicationData_IsConstraintValid(const T_tc_applicationData* val, int* pErrCode);
flag T_tc_applicationData_ACN_Encode(const T_tc_applicationData* val, BitStream* pBitStrm, int* pErrCode, flag bCheckConstraints);
flag T_tc_applicationData_ACN_Decode(T_tc_applicationData* pVal, BitStream* pBitStrm, int* pErrCode, T_uint8 pusType, T_uint8 pusSubType);
#ifndef ERR_T_tc_applicationData_unknown_choice_index
#define ERR_T_tc_applicationData_unknown_choice_index          1037  /**/
#endif
```

Architecture and Behaviour

- Define the software interface and deployment
- Describe the dynamics



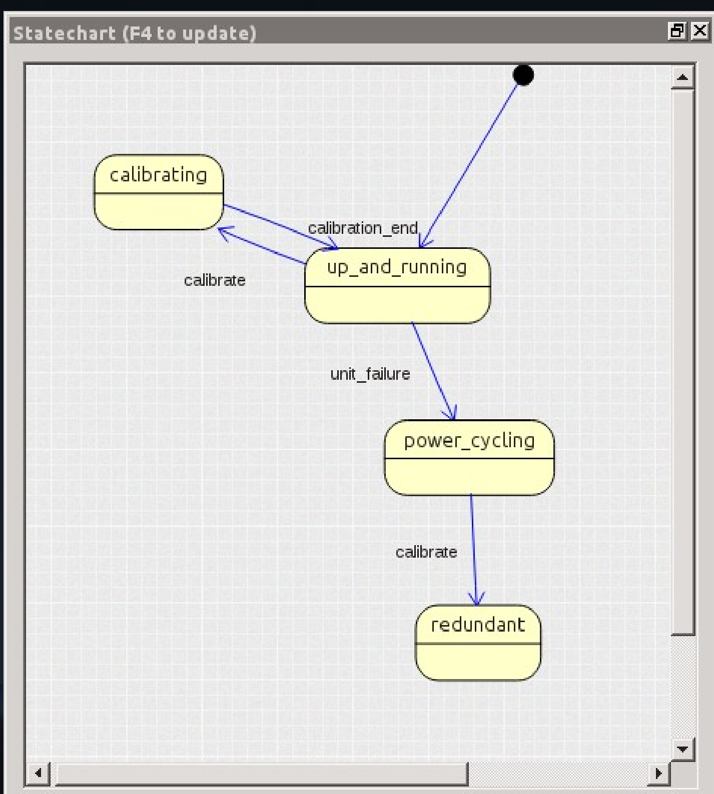
Challenge and solutions

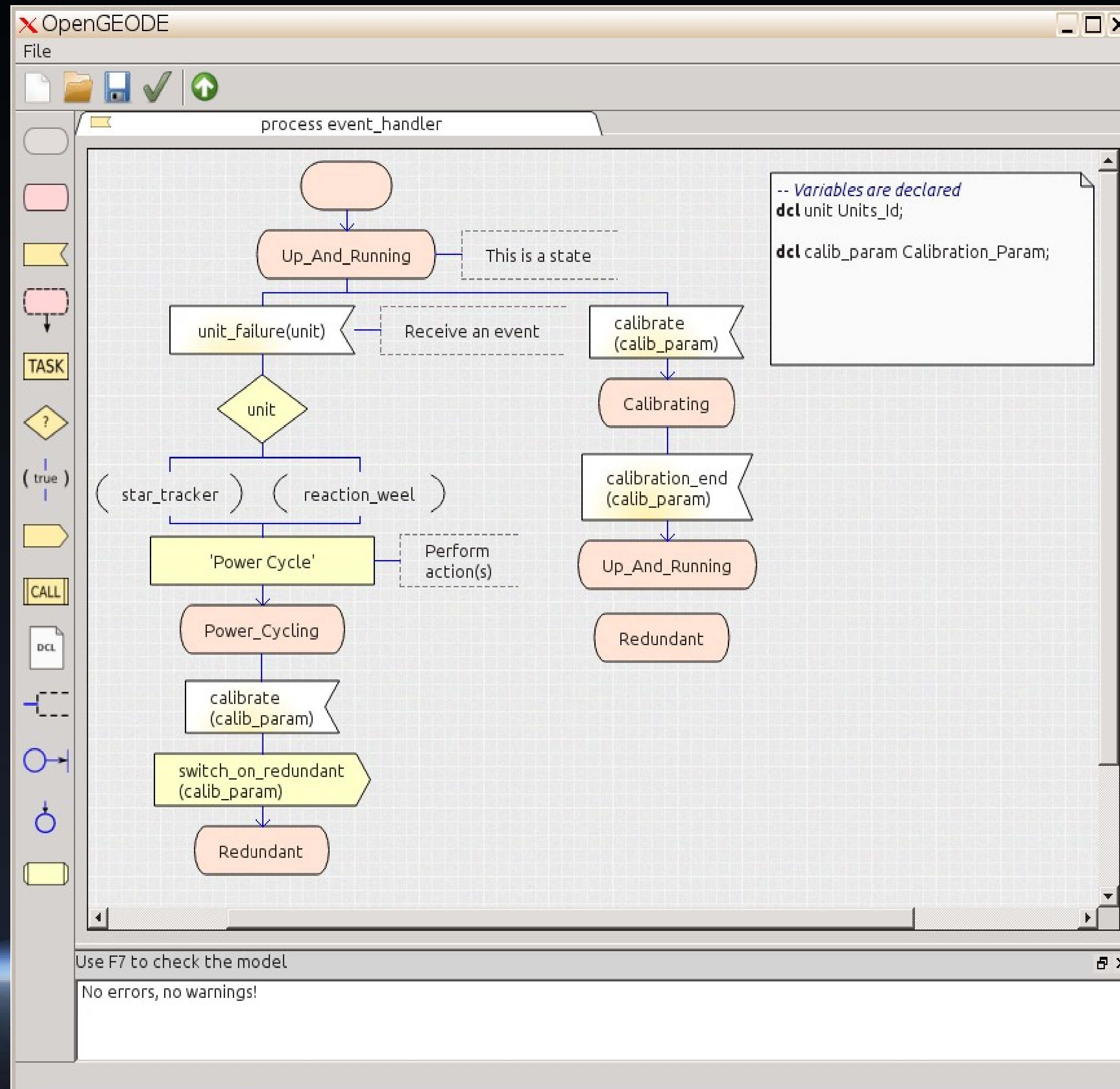
- Flight software embeds state machines everywhere
- But programming langages aren't aware of that

- ⟶ Use domain-specific langages such as **SDL**
- ⟶ Model interfaces and dynamic interactions (SAVOIR/**CORDET** approach, implemented in **TASTE**)
- ⟶ Simulate behaviour the same way as AOCS/GNC engineers simulate physics and control laws, using tools
- ⟶ Generate the code and get bug-free software !

Today's tools

- OpenGEODE
- RTDS
- PnP FW Profile





Code generators

They are mature, easy to customize, powerful

- Target safe code, without heap usage, without external dependencies
- Generated code is simple, readable, binaries are tiny and speedy
- A lot of progress was made in the past 5 years

- Target langages : C, (Spark) Ada, and LLVM

Conclusions

- ESA and the space industry are active and experienced in modelling and autocoding
- GNC teams are strongly pushing Software people to improve their development processes – and it works !
- We are willing to go further and improve the tools
- But there are still some challenges and we miss pilot projects to progress on real cases