

SYSTEM DIGITAL MOCK-UP

“A new Approach for the Design and Development of Helicopter Avionic Systems”

Eric Guillon, Nicolas Brisset, Nicolas Damiani, Mathieu Goasdoué

Aircraft System and Avionics

EUROCOPTER

Aéroport International Marseille Provence
13725 Marignane cedex, FRANCE

Abstract

The development and troubleshooting of helicopter avionic systems is becoming more and more complex because of the huge amount of data exchanged between the various equipments that make up these systems. This makes it very complex to specify, validate and test avionic systems, either on the ground (bench) or in flight.

To help design systems, EUROCOPTER is currently developing a new tool called SDMU (**S**ystem **D**igital **M**ock-**U**p) to be used in the future concurrently to the system specification process, in order to reduce both development time and technical risks.

The SDMU would consist of a real-time simulation that will enable representing the interfaces between the equipments. The level of fidelity of these simulations allows to evaluate, to modify and to pre-validate the avionic interfaces in a full virtual environment without any physical equipment in the loop.

After presenting the principles of SDMU, this paper highlights the projected benefits of SDMU and describes how the simulation is built within the system specification process.

The projected benefits of this new tool are illustrated by an example of the forecast use of SDMU in the system specification process and the in-depth investigation methods is illustrated by a forecast example related to temperature management within the hydraulic system.

Abbreviations used in this paper:

AMC	Aircraft Management Computer
A/P	Auto Pilot
AFCS	Automatic Flight Control System
ARINC	Aeronautical Radio INC
ASIC	Application Specific Integrated Circuit
DMU	Digital Mock Up
HMI	Human Machine Interface
IMA	Integrated Modular Avionics
I/O	Input/Output
IRS	Interface Requirement Specification
FIFO	First-In First-Out
LVDT	Linear Variable Differential Transformer
MFD	Multi Function Display
OAT	Outside Air Temperature
RISE	Real time Interactive Simulation Environment
SDI	Source Destination Identifier
SDMU	System Digital Mock Up
VMS	Vehicle Management System
XML	eXtended Mark-up Language

1. SDMU FORECAST PRINCIPLES

1.1. Projected SDMU concept

In the same way as EUROCOPTER has developed a Digital Mock-Up for describing the geometry of helicopters, a new project is currently launched in order to develop a Digital Mock-up of avionic systems named SDMU (**S**ystem **D**igital **M**ock **U**p)

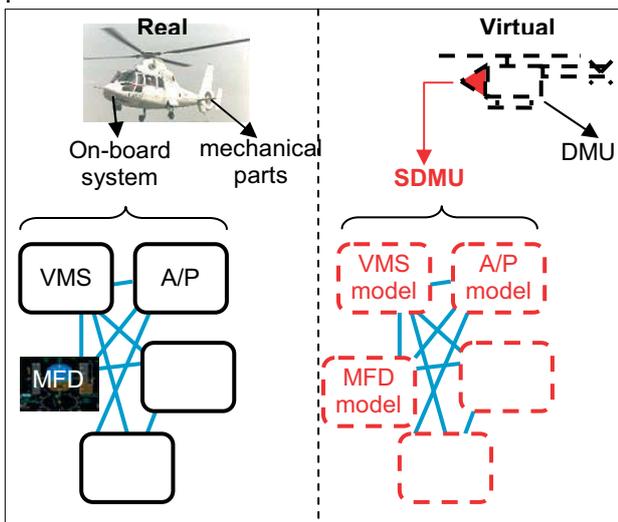


Fig.1: Real world and Virtual world

One of the main difficulties in avionic system development is to produce a complete and consistent interface definition. The main goal of SDMU will be to simulate interfaces between system components in order to fix equipment specifications as early as possible to reduce the lead time of integration tests on rigs and flight tests. To reach this goal it will be necessary to have:

- A database that describes the interfaces of every equipment including all input and output signals
- A flexible simulation platform on which to run all the models and which includes a set of tools intended to analyse the behaviour of the system.

The models forecasted to be used for simulating the system would be:

- o Models of the physical behaviour of components included in the system simulation (hydraulic system, electrical generation system, flight dynamics, ...)
- o Sensor or actuator models that translate engineering data from the physical world to system format parameters.
- o On-board software models (VMS, AFCS, etc).

1.2. New Centralized Interfaces database

Interface data from previous architectures were described in a database which mainly included data exchanged by ARINC 429 or 1553B data busses.

Architectures for modern helicopters now use integrated modular avionics (IMA) based on ARINC 653 protocol. This protocol includes, in addition to inter-equipment exchanges, inter-partition exchanges inside the same equipment. (See Annex A).

As these new types of ARINC653 exchanges were not supported by previous generation tools, EUROCOPTER decided to program definition of a new interface database that would use a more appropriate format, based on XML descriptions, which would be able to handle ARINC 653 data and also discrete as well as analog data.

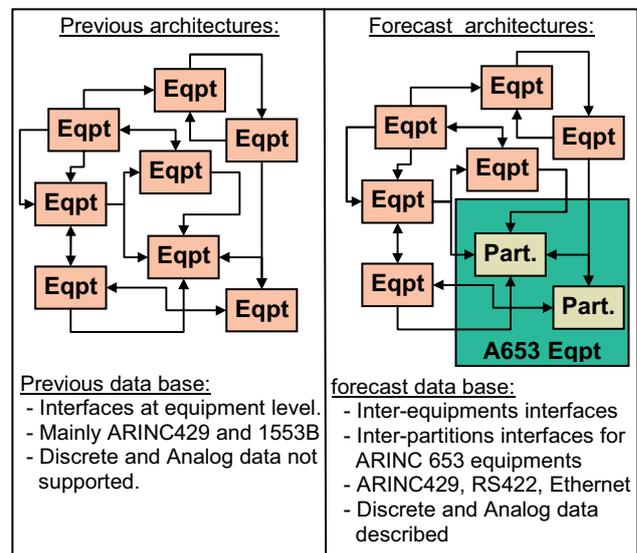


Fig.2 Forecast centralized database

The purpose of this forecast interface database is:

- To describe the architecture of the avionic system along with communications between all equipment (internal or external) comprising the system,
- To provide a detailed description of each equipment and partitions' inputs and outputs (Extension of data structure description from the previous data description tool),

This forecast Interface Data base description could then be used:

- To generate Interface Requirements Specification (IRS) documents,
- To automatically generate interface data in a neutral format which would be interpretable by the test benches' software in order to configure bench and flight test tools.

- To “feed” the forecast SDMU simulation: Interface description files would be automatically analysed by a translation process (SDMU parser) which would generate all the source code files (C/C++) which in turn would provide services and functions required to simulate data exchanges within the system.

1.3. Forecast scheduling

SDMU tool is to be mainly based on the R.I.S.E framework [1] and would include additional translation and verification tools, with which it would be possible to check the interface consistency, and provide a means of in-depth investigation.

Each equipment model would be managed in real time at the appropriate frequency to be as representative as possible of the embedded software.

Due to RISE’s multi-threading management capability, the forecast simulation representativity can be increased. Cross-checks between partitions can thus be simulated including ‘wait’ mechanisms between processes during execution.

2. FORECAST BENEFITS OF SDMU

2.1. Variety of models

Due to the different services available within the RISE framework, a wide variety of models would be used in the forecast SDMU:

- Models designed by providers :

In order to validate the interface and behaviour of a specific subsystem, equipment manufacturers are asked to provide representative models which are integrated in the forecast SDMU simulation. (Engine model, ancillary board models, etc).

- Embedded software:

On-board software under EUROCOPTER responsibility is developed using SCADE® and/or VAPS® tools, with the capability of automatically generating the corresponding C-code files which would be integrated into the simulation as in the actual avionic equipment. This would ensure that the model used for the forecast SDMU simulation be as close as possible to real embedded software. (Vehicle Management System (VMS) software, or for Automatic Flight Control System (AFCS) software)

- Physical models to be developed by EUROCOPTER:

These models would be written in C/C++ and would simulate behaviour of the helicopter physical systems (hydraulic, fuel, electrical

generation, etc) and could be used to study interactions between these systems and the avionic system under development.

These models would also be used to simulate failures and check the system behaviour in degraded cases.

2.2. Flexibility

SDMU simulation would be able to run on a Linux PC (desktop or laptop) and would not require any specific and possibly expensive hardware (such a test bench).

It would be able to simulate a partial or complete system which would then to be used in the pre-design phase as well as in the verification and validation phases.

2.3. Data exchanges

The major innovation of the forecast SDMU would be to simulate in real time the data flow between and within ARINC653 equipments as a result of an automatic translation process relying on the centralised Interface Database.

In order to run a simulation that would be as close as possible to the real system, one of the forecast SDMU principles would consist of representing all the data produced by the different equipments in their real/physical format without any simplification. Discrete and Analog data would be represented as electrical values, ARINC429 data as FIFO buffers containing the labels words, each of them containing the data coded with the true bit sequence. This is called “**Raw data format**”

Nevertheless, most of models that would simulate the physical behaviour of the helicopter and its systems, would operate using “**Engineering data format**”. This format would be translated into in the appropriate and expected “Raw data format” by the sensor models that use forecast SDMU generated conversion functions to ensure the communication with the avionic system.

Both Raw and Engineering data formats would be mixed for simulation, allowing a step-by-step approach all along the specification phase.

This ensures the highest representativity from a data exchange point of view, compared to exchanges restricted to engineering data.

In addition, for dual channel equipment*, two channels by equipment would be simulated which can represent the cross checks between channels of the same equipment (Crosstalk) or between different equipments (Data Exchanges).

* System software is duplicated into two channels running in the same equipment and this equipment being also duplicated within the system

2.4. Investigation means

Using the RISE framework, high performance investigation tools would be available for SDMU simulations:

- Visualisation tool (dumper) to display and/or modify engineering or raw data which would be decoded in real time and are accessible as data generated by the equipment as well as data from the physical link.
- Graphical capability for data variation display
- Record / Replay capability
- "ARINC429 disturber" tool for displaying and/or modifying data at label value level or at bit level (data would be encoded in real time into labels). (See figure 7). This possibility will be extended to every type of data managed by SDMU (Analog, Discrete, RS422,etc)

3. BUILDING THE SIMULATION

3.1. Interface description

The first step required to build SDMU simulation would be Interface analysis. Due to processes included in the forecast SDMU tool set, all the interfaces would be analysed to check the consistency of data exchanges, then a set of files would be produced for all the signals exchanged through the system and all the functions needed to manage these signals. (See Annex B Step #1)

The centralized database would be mainly composed of XML files. For each type of equipment two main files would be mandatory:

- A file describing all the outputs generated by each type of equipment including how data would be formatted. For example, we will find in this file, all data to define correspondence between physical data and electrical data.
- A file describing all the signals sent by a given type of equipment and all the occurrences of this equipment. This file would also describes, for each signal, which equipment or partition will be a consumer of this signal.

In addition, for ARINC 653 equipments two more files would be mandatory:

- A file describing all ports sent or received by each partition of each equipment.
- A file describing all channels sent and received by each partition of each equipment.

All extant numerical structures (ARINC429 or RS422 or Data Structure) would be described in a specific file that includes framing parameters

Finally, a global file would be used to enumerate all

equipment occurrences for the assessed subsystem.

3.2. Interface translation

The second step would consist of using a specific process to translate all the interfaces described in the database into files that are usable by the SDMU simulation (libraries, headers ...). (See Annex B Step #2)

All files describing the interface of the assessed subsystem would be automatically analysed by the SDMU translator tool to generate the following components:

- A C/C++ structure defining a shared memory that includes, for each equipment, the data produced for each module, and for each partition.

- A C/C++ structure defining a shared memory that includes, for each signal sent, the details of each data included in the signal.

These two shared memories would describe, using different formats, all data exchanged in the system using the real aircraft format.

The SDMU translator process (SDMU parser) would also generates a set of communication functions specific to ARINC 653 protocol to create, write and read ports described in the database for the equipment under investigation:

```
CREATE_SAMPLING_PORT,
READ_SAMPLING_MESSAGE,
WRITE_SAMPLING_PORT ...
```

In addition, the forecast SDMU translator process would generate all the functions required to encode and decode data from engineering format to raw format (and vice-versa):

```
analog_set_value, analog_get_value,
discrete_set_value, discrete_get_value,
a429_set_label, a429_get_label,
a429_set_sdi, a429_get_sdi,
a429_set_ssm, a429_get_ssm ...
```

and a complete set of functions used to decode and encode all labels described in the centralised data base:

```
Lxxx_yy_get_zzzz et Lxxx_yy_set_zzzz
where
xxx represents label number,
yy represents SDI and
zzzz represents encoded field.
```

3.3. Sub-systems translation

Today, almost all of EUROCOPTER's sub-system specifications are entirely defined using SCADE[®] tool. The forecast tool would integrate a C-code generator to translate functional description sheets into C-code files. These generated files would be then embedded in SDMU simulation to be managed

as other simulation models are.
(See Annex B Step #3)

To simulate a partition included in dual equipment, the related model would be duplicated into four channels. Thus the simulation user would be able to verify the exchanges between channels within the same equipment (Cross-Talk) or between equipments (Data Exchange).

In the same way, the HMI partition used to display flight and vehicle data to the crew is defined using SCADE® for the logical part and VAPS® for the graphical part. These two tools include a C-code generator to obtain a graphical model communicating with other models through ARINC429 exchanges. (See Annex B Step #4)

3.4. Simulation Environment

In order to emulate the assessed sub-systems, a set of models are to be developed to “feed” the simulation with representative data from the physical world. There would be, for example, a hydraulic system model, a fuel system model, an electrical generation model or a flight dynamics model.

All these models would produce or receive data in engineering format (physical data) which will have to be translated into interface value like frequencies or voltages ...(raw data)

To perform this translation, a set of sensor models or/and actuator models will be developed. To encode or decode values from Engineering data format to Raw data format (or vice-versa) these models would use functions and services provided by the forecast SDMU translator (cf §3.2).

They would be thus automatically updated when a new simulation is to be integrated after delivery of a new version of the centralised interface data base or a software upgrade. (See Annex B Step #5)

In addition, in order to help users to easily interpret exchanges performed during simulation, specific interfaces would be developed:

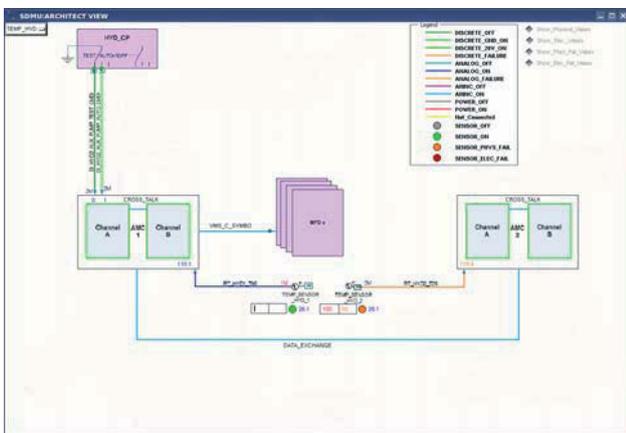


Fig. 3: Architect's view for Hydraulic function

This graphical interface would allow, for each assessed function (one dedicated page per function), the display of Analog and Discrete values. It will also be used to indicate ARINC429 line activity and to mute it if necessary (failure simulation).

In the same way, for discrete and analog data managed by sensor models, the simulation user will be able to modify the physical values as well as electrical values to simulate failure or degraded modes through dedicated control panel and architecture views.

Due to the tools available within RISE framework, SDMU users would be able to easily display the value of any data included in shared memories. He will also be able to easily suspend the execution of a particular model and modify its outputs in order to analyse the effect on the other models

4. EXAMPLES OF USAGE

4.1. Function verification and validation

The onboard VMS (Vehicle Management System) is in charge of all helicopter parameters management. A prototype of dedicated SDMU simulation for VMS (Vehicle Management System) is currently being developed during all the phases of the development of the VMS software. This forecast SDMU simulation would be used for:

- System architecture validation
- Functional requirement validation
- Embedded software verification
- Dynamical system behaviour analysis

The possibility to translate data automatically from a centralised interface and to automatically translate specification from SCADE® and VAPS® to the simulation model, means that SDMU would be included in EUROCOPTER validation and verification processes.

At each major stage of the development process, a release of the sub-system software would be made. If necessary, a new set of interface files would be updated. Both elements would be delivered for simulation integration and a new release of the HMI partition would be integrated if this is needed to ensure data exchange consistency.

Due to the automatic process that would be used, the time between specification and interface delivery and simulation availability will be very short (several hours). This would allow multiple iterations between the specification and the simulation, thus greatly reducing development time.

The updated simulation would be then delivered to users (system architects and software developers) who use it to assess interface consistency and sub-systems behaviour.

This kind of process can of course help to detect different types of problems at different levels:

- By testing the consistencies between the interface description and SCADE® model.
- By testing the consistency between data exchanged from an equipment to another in terms of validity.
- By testing the correct display of data sent to HMI partition

These tests performed up stream of the validation process would reduce the number of problems detected later on test rigs or during flight tests, which has a direct impact on the cost and timescale of the system development.

When architects, in charge of specification, would decide that it is mature enough to be presented to all people involved in this development, a specific meeting would be hold so as to validate the specification. During this meeting, in presence of all concerned (Project team, Airworthiness and Safety specialists, rig test and, flight test personnel), all the elements of the specification would be presented for validation and the SDMU simulation would be used to illustrate every part of the functionality to be validated.

4.2. In depth Investigation

Using the RISE Framework and its associated tools, SDMU users will easily access all data described in the various shared memories, and also the internal data of each model. Thus users would be easily and deeply investigate the software, for example to tune the assessed function.

When models are compiled, data mapping would be used to generate the files used by SDMU to enable access to internal data values. This tool, named “scheduled dumper” (see fig.5), would be used in real time to display the internal values for each running model.

To illustrate the possibilities of the forecast SDMU in terms of investigation, we will focus on temperature treatment within the hydraulic system. The different steps of this treatment are summarised in figure #4.

Outside Air Temperature would be managed by a specific environment model.

This temperature would be then used by hydraulic model to generate physical oil temperature within hydraulic system #2 (named Temperature29).

Then sensor model would translate this engineering data to raw data (voltage included in RT_HYD_29 signal) would be using a function generated by the forecast SDMU parser.

AMC2 would receive this signal and performs all functions concerning this data:

- Check with the second equipment channel (Cross Talk checks).
- Check with the second AMC (Data Exchange checks).
- Perform all controls needed to display corresponding information to the crew.

Then AMC1 would be able to generate ARINC429 data to be sent to the MFD using the VMS_C_SYMBO ARINC line.

These ARINC429 labels would be read by the MFD and automatically decoded into engineering data (HYDT29) for displaying information to the crew.

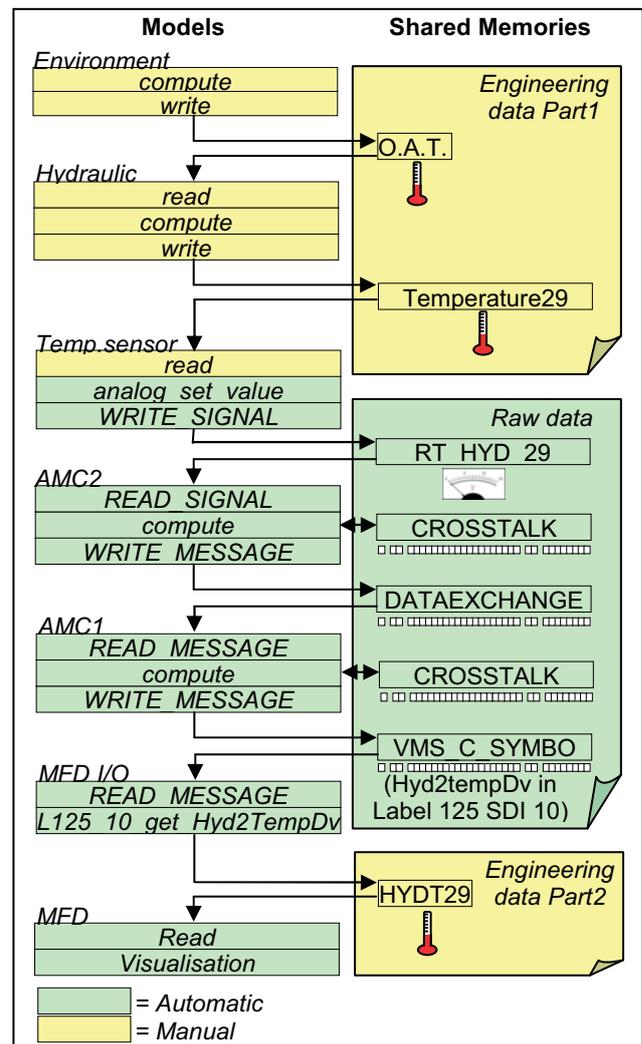


Fig. 4 : Data treatment example: Temperature from Hydraulic system #2 (T29)

SDMU users would be able to investigate at each step of the process, for example using the scheduled dumper for displaying, recording or modifying data from shared memories (Engineering or Raw).

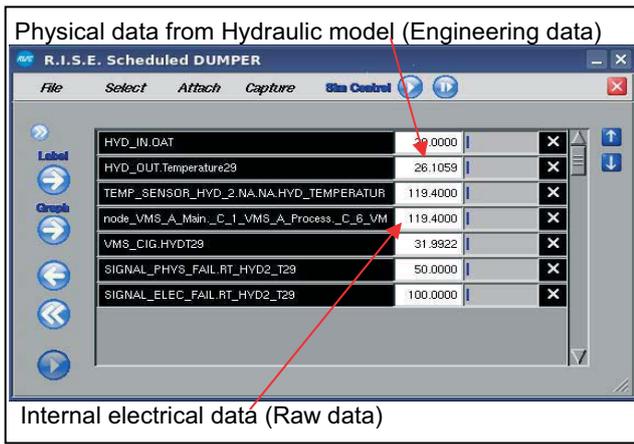


Fig.5 : Scheduled Dumper

SDMU users would also use the Architect view and its input fields to modify Engineering or Raw data from sensors. In the following example, a physical value is set to 50°C instead of 26.1°C. The electrical value would be automatically updated to the corresponding value (119.4 Ohms)

Thus system designers would easily assess the consequences on the hydraulic function and test, for example, the field of validity of the temperature management function.



Fig. 6: Architect's view detail

Using the "ARINC429 Disturber", SDMU users would also directly modify ARINC429 label content. In the following example, data sent from the AMC to the MFD is set to 31.990140°C instead of its nominal setting. The result would be seen using the Scheduled Dumper. The displayed temperature (HYDT29) would be 32°C.

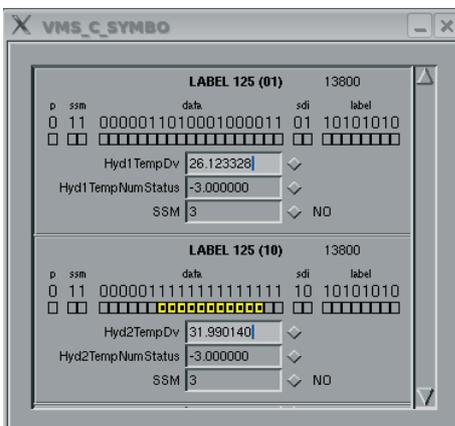


Fig. 7 : „ARINC 429 Disturber“

5. CONCLUSION

The main goal of SDMU is to validate the system interface definition through simulation.

Its testing during research phase has already demonstrated that it can also be very useful to assess sub-system specifications before integrating the onboard code.

Others uses of SDMU would be:

- Tuning of bench test & ground test procedures.
- Training of Maintenance and Customers Support people.
- Reuse for training applications (Control Panels, simulation models, simulation architectures).

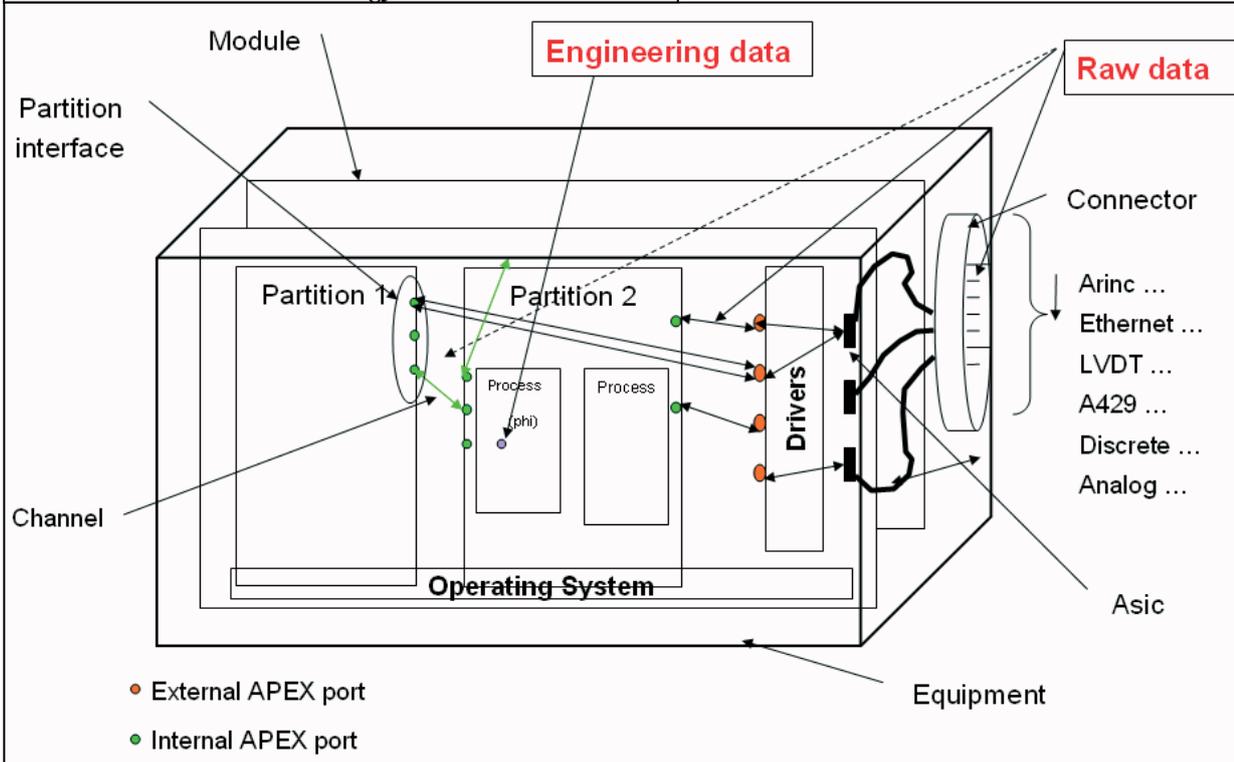
A further development would be to enable the user to independently update their models (provided there is no interface modification) in order to further reduce development time.

SDMU could soon be used for EUROCOPTER system development and would then strongly participates to risk mitigation and to cost and lead time reduction.

References:

[1] N.Damiani, R.Marhic and Y.Brun – "NH90 FORMAL SPECIFICATION & ENGINEERING PILOTED SIMULATION" 30th European Rotorcraft Forum – Marseille, France 14-16 September 2004.

Annex A: ARINC653 terminology used in EUROCOPTER



Annex B: SDMU simulation building process

