TOTAL PAGES : 34

HERSCHEL / PLANCK

MOIS – Rules to support modularity and commonality

H-P-1-ASP-TN-0575

Product Code : 470000

Rédigé par/*Written by*	**Responsabilité-Service-Société *Responsibility-Office-Company***	**Date**	**Signature**
F. Chatte	Ground Segment interface and operation manager		
Vérifié par/*Verified by*			
B. Dubois	EGSE		
JP. Hayet			
Approbation/*Approved*			
C. Masse	Product Assurance Manager		
J-J. JUILLET	Project Manager		

Data management : G. SERRA

Entité Emettrice : Alcatel Space - Cannes
(détentrice de l'original) :

HERSCHEL/PLANCK	DISTRIBUTION RECORD		
DOCUMENT NUMBER : H-P-1-ASP-TN-0575	Issue / Rev. : 01/00D1 Date: 22-September-2003		
EXTERNAL DISTRIBUTION		**INTERNAL DISTRIBUTION**	
ESA	X	HP team	X
ASTRIUM	X		
ALENIA	X		
CONTRAVES			
TICRA			
TECNOLOGICA			
GMV	X		
RHEA	X		
		Clt Documentation	Orig.

ENREGISTREMENT DES EVOLUTIONS / *CHANGE RECORDS*

ISSUE	DATE	§ : DESCRIPTION DES EVOLUTIONS *§ : CHANGE RECORD*	REDACTEUR *AUTHOR*
01/00D1	22/09/03	First issue Over-closure of AI#3517-07 "ASP to distribute for comments table of contents of the document to be written by ASP".	F. Chatte

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The objective of this document is to give the rules to code the AIT test sequences using MOIS in order to ensure the modularity of the test sequences and the commonality between the different level of tests and also with the operation. Those rules apply for any test sequence which can be used at system level and as such is subject to be re-used for operation.

It is not realistic to think that a high level test sequence can be converted in an high level operation procedure mainly because :

- the simulation environment could be not representative,
- a high level test sequence includes a lot of call to other test sequences which generates in fact a quite big expended test sequence which reduce the possibility of commonality with operation.

It is more realistic to make small modular procedures which can be combined in different way to produce else high level AIT test sequences either high level operation procedures.

This document presents mainly how to generate modular procedures in a way compatible with the HPSDB high level data model (element / subsystem / model) and compatible with the specific MOIS implementation (multiple data bases, instantiation, ...).

Chapter 2 lists the applicable and reference documents.

Chapter 3 makes a short presentation of the use on MOIS on Herschel / Planck and specific functions.

Chapter 4 describes how to get the required modularity and the associated naming convention.

Chapter 5 specifies the way to interface the test sequences.

Chapter 6 presents the interfaces with HPSDB.

Chapter 7 presents the different test sequences exchanges on the project and the associated facility.

Chapter 8 presents the test sequence configuration management.

Chapter 9 presents the MOIS AIT customisation,

Chapter 10 presents if any the MOIS limitations,

Chapter 11 presents the operation procedures generation,

Annexes 1 and 2 provides additional AIT examples and rules.

2. APPLICABLE AND REFERENCE DOCUMENT

2.1 Applicable documents

AD1	MOIS – delta specification	H-P-1-ASP-SP-0390
AD2	HPSDB – Specification	H-P-1-ASP-SP-0082
AD3	Naming convention	H-P-1-ASP-SP-0141
AD4	HPSDB – data collection plan	H-P-1-ASP-PL-0455

2.2 Reference document

RD1	MOIS help V5.7	
RD2	HPSDB Guide	H-P-1-ASP-TN-231

2.3 Acronyms

ACMS	Attitude Control and Measurement Subsystem
AD	Applicable Document
AIT	Assembly Integration Test
ASP	Alcatel SPace
AVM	Avionics Model
CDMU	Control and Data Management Unit
CQM	Cryogenic Qualification Model
DB	Data Base
ESOC	Europeen Space Operation Center
FOG	
HPSDB	Herschel / Planck System Data Base
MOC	Mission Operation Center
MOIS	Mission Operations Information System
MTL	Mission Time Line
NMCVT	NaMing ConVenTion
OBDH	On Board Data Handling
PFM	Proto-Flight Model
PLM	PayLoad Module
RD	Reference Document
RF	Radio Frequency
RW	Reaction Wheel
S/C	SpaceCraft
SCOS	Spacecraft Control and Operation System
STR	Star TacKer
SVM	SerVice Module
TBC	To Be confirmed
TBD	To Be defined
TBW	To Be Write
TOPE	Test and Operation Procedure Environment
TC	TeleCommand
TM	TeleMetry
TS	Test Sequence

UDC dynamic User Defined Constant

2.4 Definitions

"Box object"

A "box object" is one "element" or one "subsystem" or one "model". All the "items" (parameter, TM packet, TC packet, constants, ...) contain in a "box object" can only refer to other items inside the same "box object", they cannot refer to items not belonging to the same "box object" except for "generic item".

"Generic item"

A "generic item" is an "item" which can be addressed by any other "item" of HPSDB (for instance : ON/OFF calibration curve). All "generic items" are grouped inside a unique "generic" "box object".

"Instantiation"

When an "element" is associated to a subsystem (for instance STR associated in ACMS subsystem), the identifiers of the items of this "element" are automatically instantiated with the subsystem identifier and the position according to AD3. For instance the parameter M012 of element STR will be instantiated in AM012030 in the nominal STR which is inside A subsystem and in position 030, and it will be instantiated in AM012031 in the redundant STR which is inside A subsystem and in position 031.

"Item"

An item can be : a curve, a parameter, a TC packet, A TM packet, a parameter, ... An item is unique inside a "box object".

"Level"

HPSDB allows to define "box object" at four levels :

. "Element level" (for instance definition of a STR)
. "Subsystem level" (for instance definition of ACMS subsystem)
. "Model level" (for instance definition of Planck PFM)
. "Generic level" (for instance ON/OFF calibration curve)

. 1 - The maximum of items are entered by the prime at "generic level",
. 2 - The maximum of items are entered by the element contractor at "element level",
. 3 - The maximum of items are entered by the subsystem contractor at "subsystem level",
. 4 – The remaining data are entered by the model contractor at "model level".

"Position"

"Position" gives the position of an element inside a "subsystem". This "position" is unique and can be allocated to zero or one and only one "element". The "position" is used inside the "subsystem" to instantiate the "item" identifiers in order to guarantee there is no duplicated identifiers. Note : there is no position of a subsystem inside a model, the positions are in fact allocated for model (this is due to SCOS limitations).

"Pseudo element"

A "pseudo element" is a virtual (it does not exist) "element" inside a "subsystem" which contain all the "items" which are derived from "items" of different "elements" inside the "subsystem" (for instance a "derived parameter" calculated from a "parameter" of the nominal STR and from a "parameter" of the redundant STR). A "pseudo element" inside a "subsystem" has also a "pseudo position" (refer to AD3).

"Pseudo subsystem"

A "pseudo subsystem" is a virtual (it does not exist) "subsystem" inside a "model" which contain all the "items" which are derived from "items" of different "subsystems" inside the "model" (for instance a "derived parameter" calculated from a "parameter" of ACMS subsystem and from a "parameter" of power subsystem). A "pseudo subsystem" inside a "model" has also a "pseudo position" (refer to AD3).

3. MOIS PRESENTATION

MOIS has been developed initially to support the generation of operation procedure. It has also be adapted to be able to generate AIT test sequence in any language. Refer to Figure 3-1 MOIS overview

On Herschel / Planck several adaptation has been required in order to :

- Support TOPE language generation (refer to Figure 3-2 TOPE interface),
- HPSDB interface including multi-databases interface(refer to Figure 3-3 HPSDB interface 1/3 (Overview) , Figure 3-4 HPSDB Interfaces 2/2 (Scenario at T0) and Figure 3-5 HPSDB Interfaces 3/3 (Scenario at T1)),
- Merging of library (refer to Figure 3-6 Merging),
- Instantiation (refer to Figure 3-7 Instantiation),
- Cross reference (refer to Figure 3-8 Cross references),
- Reverse generation (refer to Figure 3-9 Reverse generation),
- SCOS 2000 improvements (refer to Figure 3-10 SCOS changes).

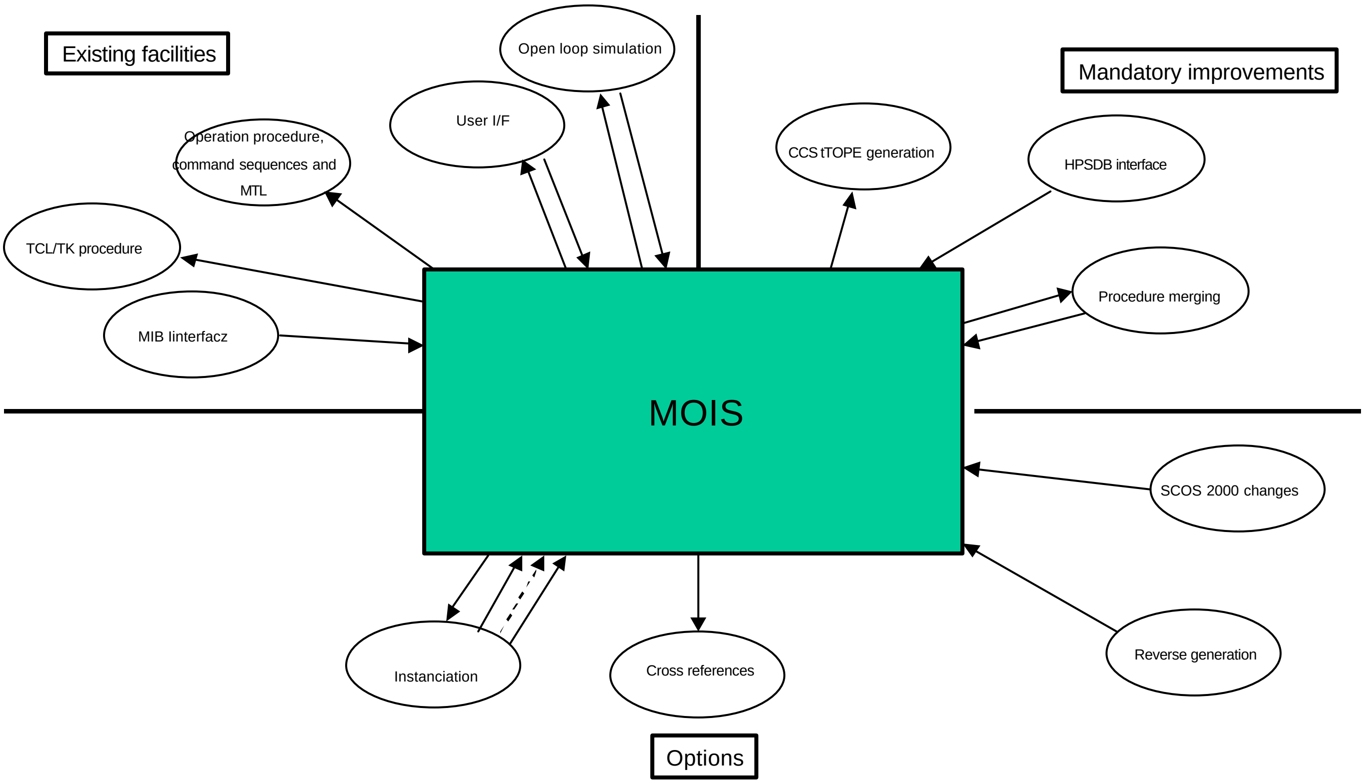
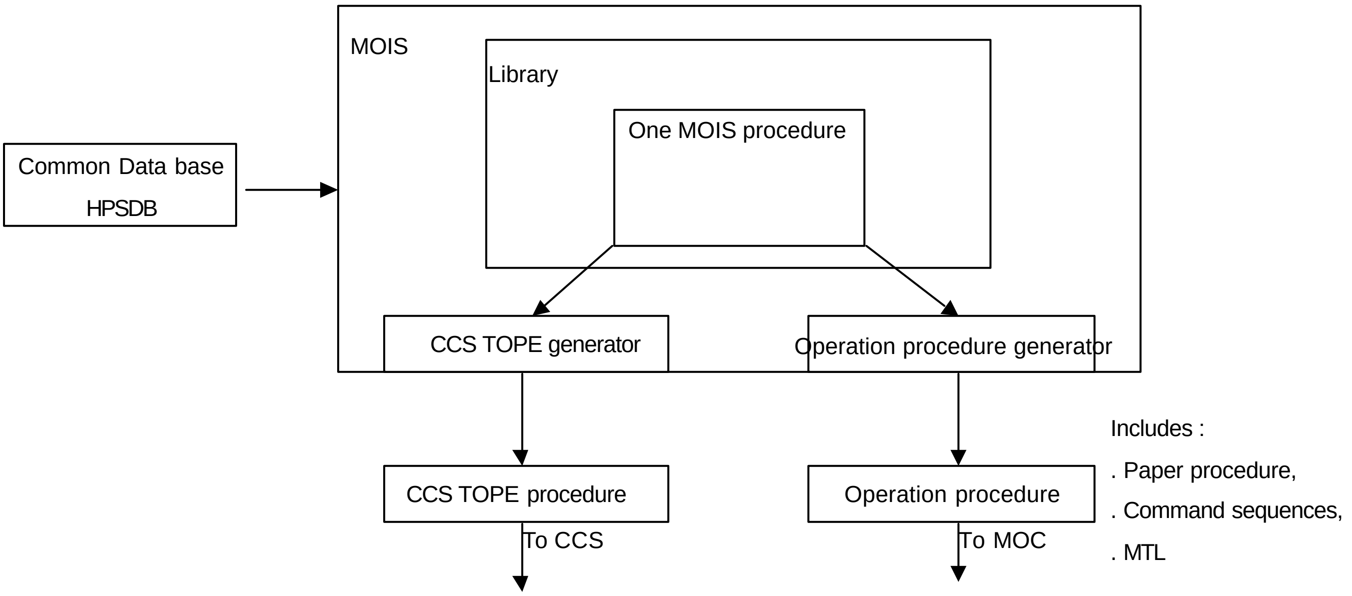
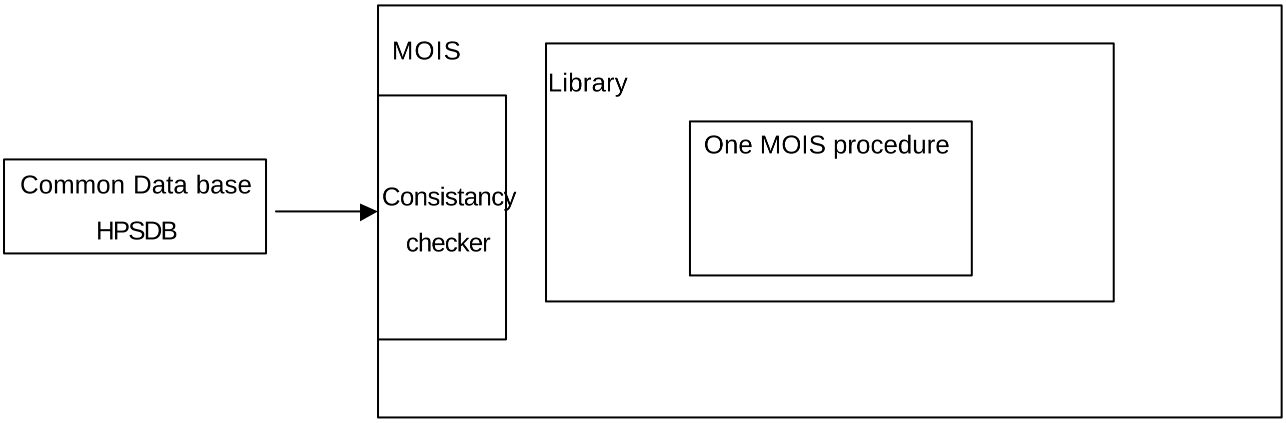


Figure 3-1 MOIS overview



As soon the CCS TOPE procedure has been validated in AIT, the operation procedure is declared validated.

Figure 3-2 TOPE interface



The MOIS interface with HPSDB shall be compatible with the three HPSDB interface levels :

. Element (for element test or / and instantiation)

. Subsystem (for subsystem level test)

. Model (for module and system test and operations)

And shall follow the naming convention.

Figure 3-3 HPSDB interface 1/3 (Overview)

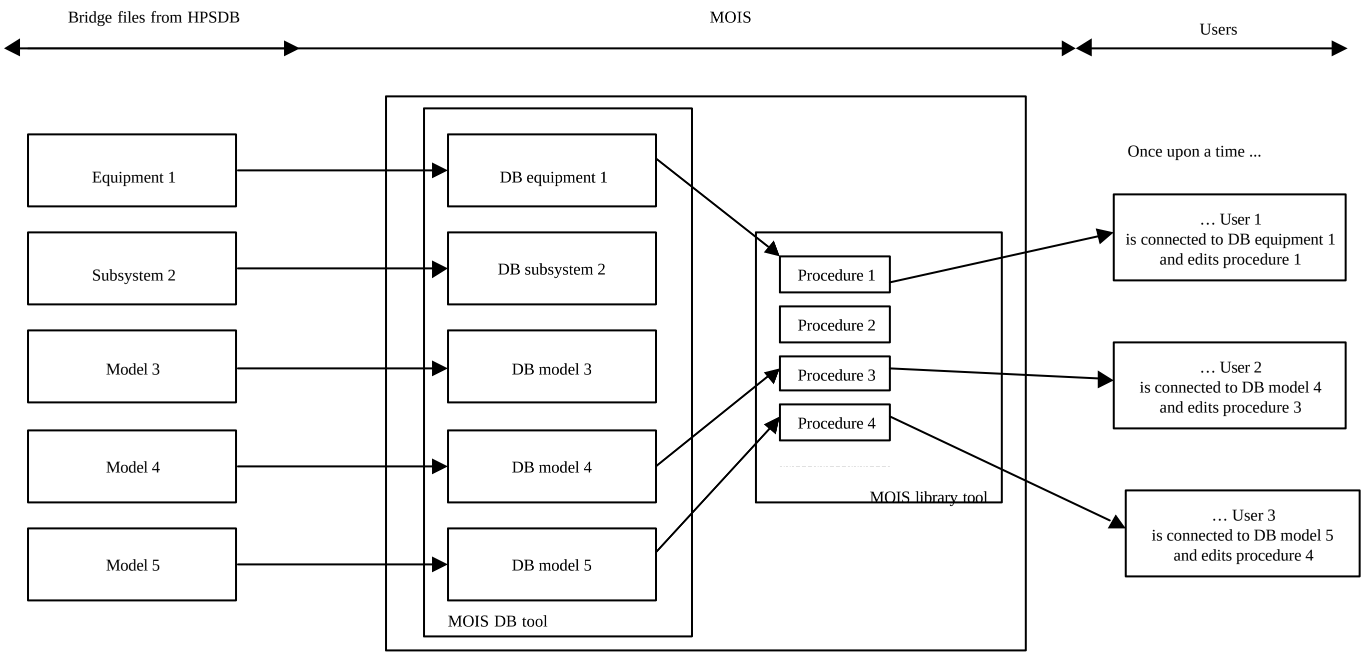


Figure 3-4 HPSDB Interfaces 2/2 (Scenario at T0)

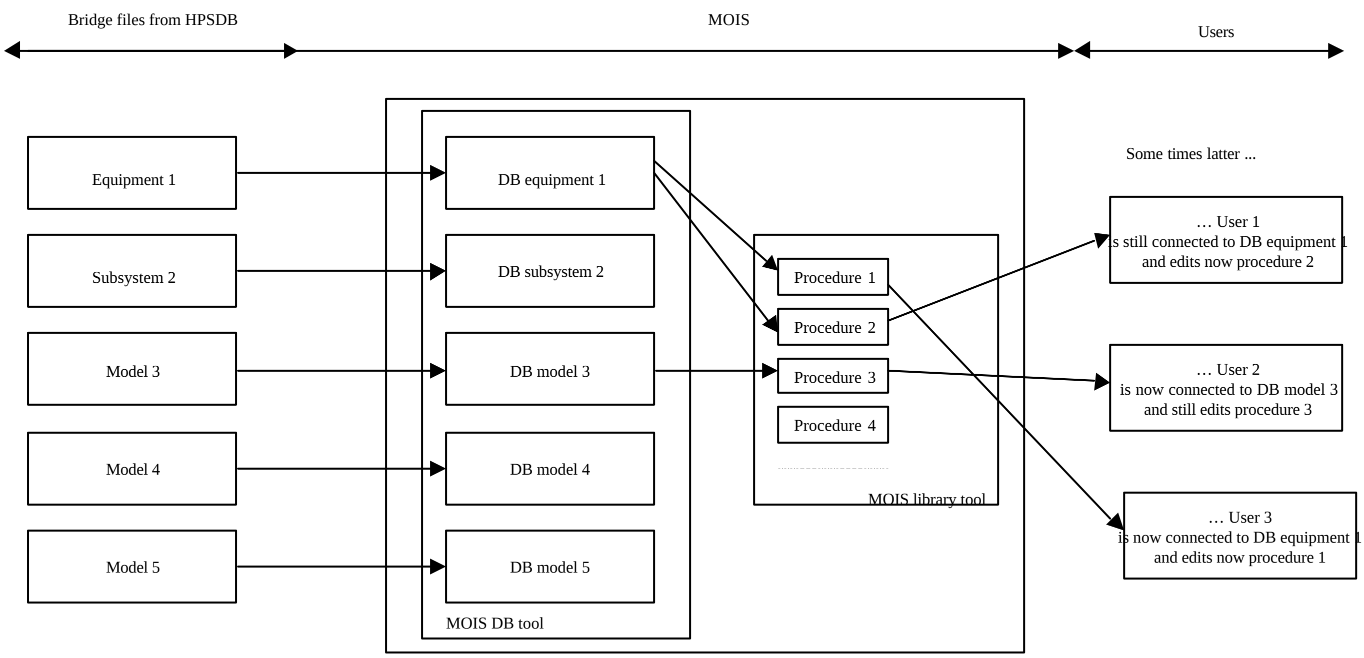
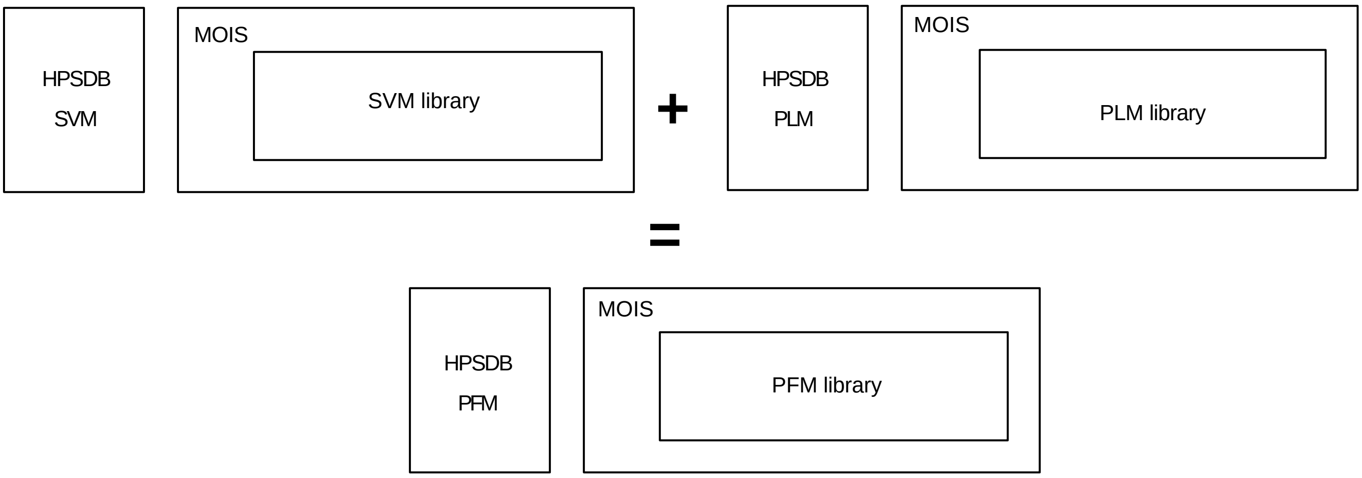


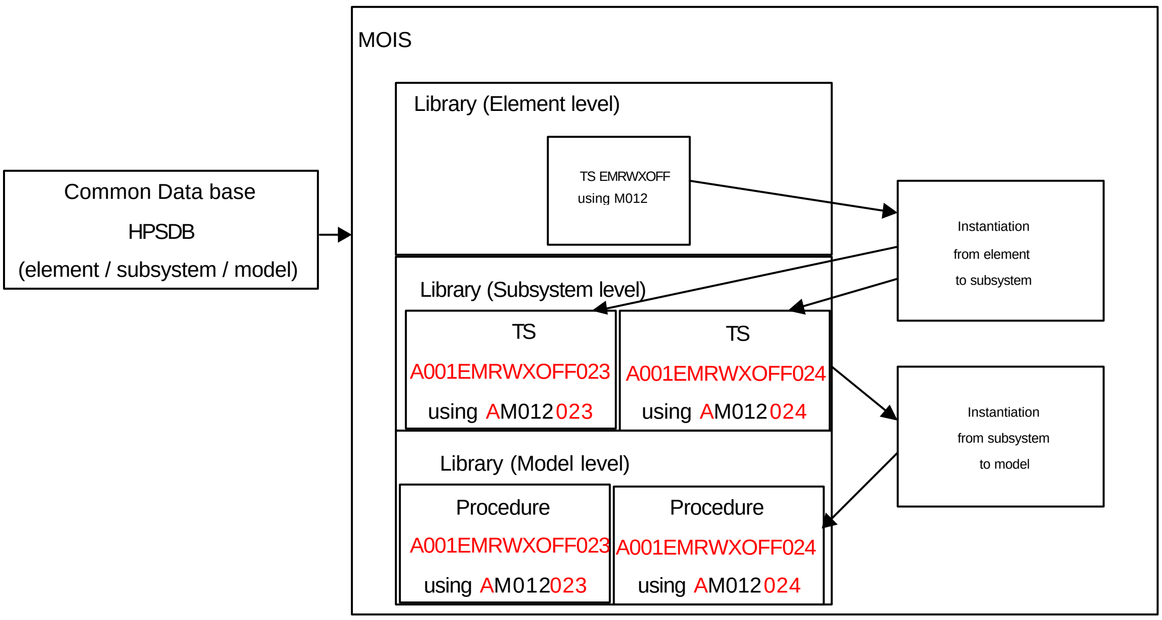
Figure 3-5 HPSDB Interfaces 3/3 (Scenario at T1)



The SVM library and PLM library are merge to build the PFM library

for both Herschel and Planck

Figure 3-6 Merging

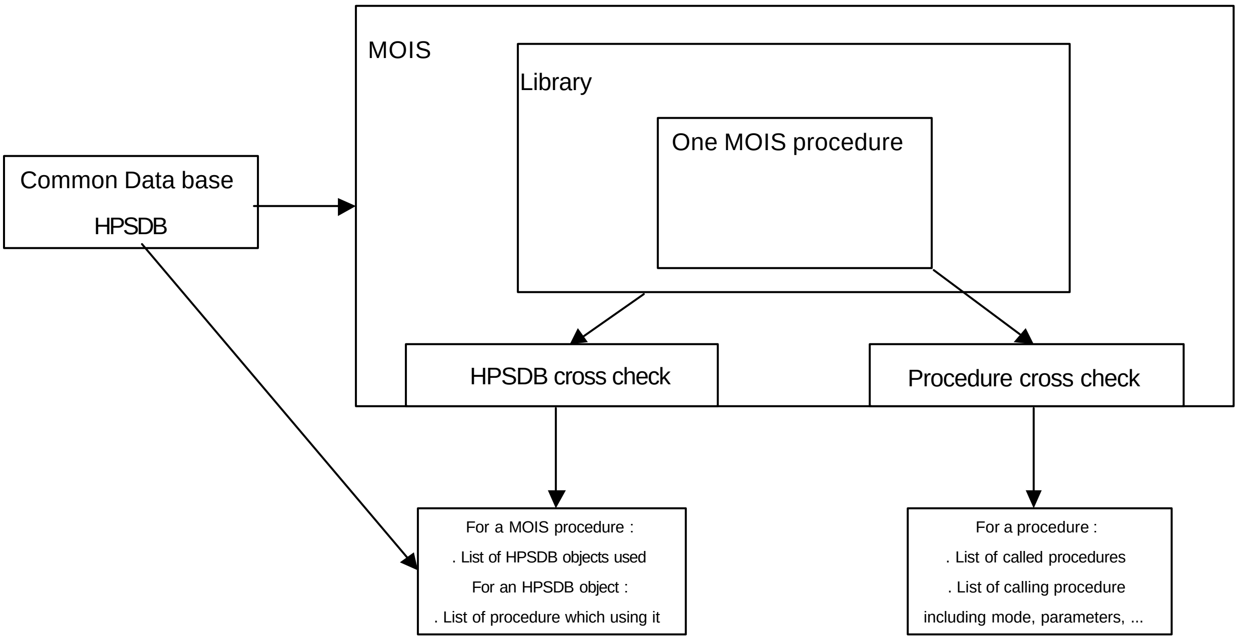


A procedure defined once at element level can be used to generate the different procedure need at subsystem or model levels

for nominal and redundant elements and for the different subsystems and models (Herschel / Planck, …).

This forces to generate modular procedure (element / subsystem / model)

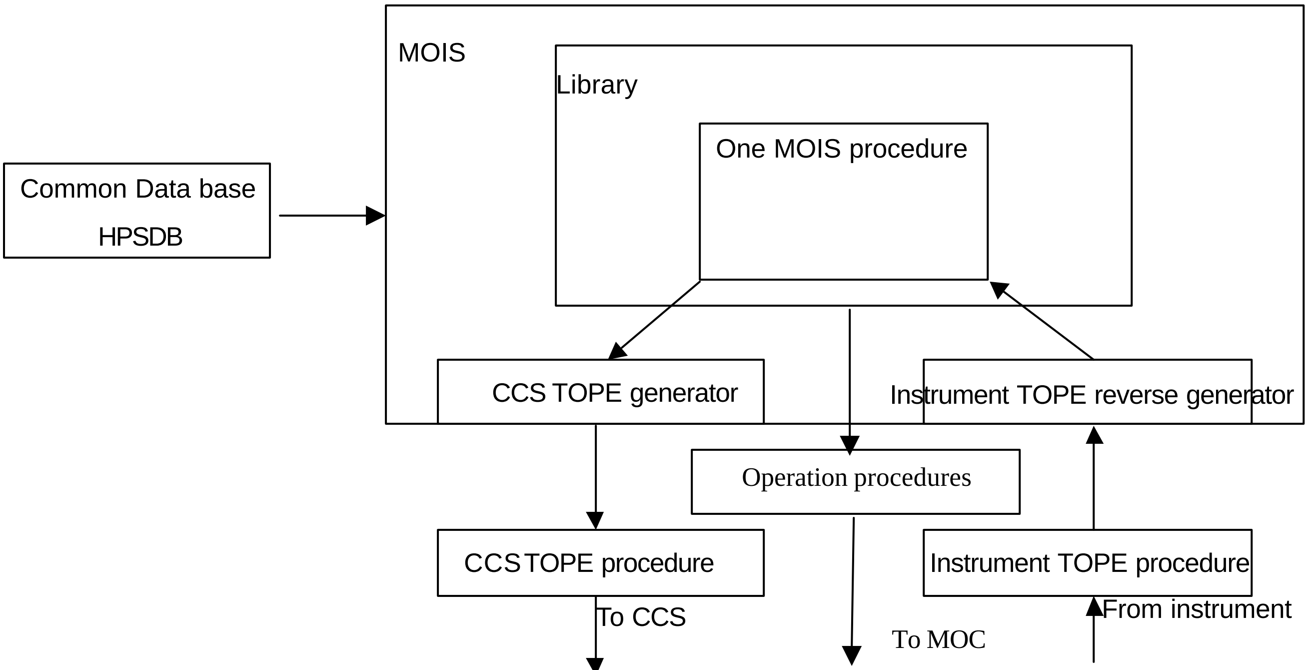
Figure 3-7 Instantiation



This facility helps the user to analyse the impact of one change in the data base or in a procedure

by providing all the cross references

Figure 3-8 Cross references



The instrument TOPE procedures from subsystem level test are loaded inside MOIS,

It is then possible to generate CCS TOPE procedures for module and system level test,

It is also possible to generate operation procedures.

Figure 3-9 Reverse generation

MOIS Kernel to be changed to support :

. Calibration curve identifier,

. Range set identifier

. Command verification stage identifier

as HPSDB does

Figure 3-10 SCOS changes

4. TEST SEQUENCES ORGANISATION

This chapter provides both a naming convention to apply to all the AIT test procedures in order to ensure that no duplicated name can be generated on different MOIS servers (except for common procedures) and to ensure that the name can be "decoded" to understand the objective of the test sequence. It presents also what should be the contents and interfaces of each test sequence in order to get modular test sequences. To achieve such objectives the chapter is based on the different levels defined in HPSDB, and on the ALCATEL AIT experience.

4.1 Test sequence levels

(Refer to Figure 4-1 Element/subsystem / Model levels - Links and inheritage)

As for HPSDB, four levels are foreseen :

- Generic test sequences,
- Element test sequences,
- Subsystem test sequences,
- Model test sequences.

4.1.1 HPSDB levels and items.

HPSDB allows to manage "box objects" and "items".

A "box object" can be an element (Star tracker for instance), a subsystem (ACMS subsystem for instance) or a model (Herschel PFM for instance). In addition HPSDB support a unique "generic box object". Each "box object" contain lists of items.

The items inside a "box object" can be linked together (for instance a parameter can refer to a curve) or can be linked to generic items which are grouped inside the "generic box object" (for instance a parameter can refer to the generic ON/OFF calibration curve). A generic item can refer to any other generic item but it cannot refer to item belonging to other "box object" than the "generic box object". The items supported by HPSDB are the following :

- TM packet,
- TM packet standard,
- TM packet PSICD,
- TM packet SCOS archiving,
- TM structure,
- TM packet group,
- TC packet,
- TC packet header,
- TC structure,
- TC packet group,
- Command sequence,
- Command verification stage,
- 1553 status word,
- 1553 command word,
- 1553 message,

- 1553 acquisition command link,
- 1553 structure,
- 1553 message group,
- OBDH interrogation,
- OBDH acquisition command link,
- OBDH interrogation group,
- Parameters,
- Parameter group,
- Parameter set,
- Parameter value set,
- Parameter range set,
- Alphanumeric display,
- Graphic display,
- Scrolling display,
- Variable TM packet display,
- Constants.

Only a subset of these items are addressed in TOPE statements (Warning : the TM packet is not accessible only the TM packet SCOS archiving is).

An element “box object” contains all the items associated to that “box object”.

A subsystem “box object” is considered to be a collection of “element box objects” and specific items which are grouped in a “subsystem pseudo box object”. The items of “subsystem pseudo box object” can refer to “element box object “ items for “element box object” which are included inside the subsystem (for instance a derived parameter can be calculated from a nominal STR parameter and a redundant STR parameter). Each “element box object” inside a subsystem is associated with a unique position which allow the instantiation of all the element identifiers (this permits to differentiate for instance the nominal and redundant identifiers).

A model “box object” is considered to be a collection of “subsystem box objects” and specific items which are grouped in a “model pseudo box object”. The items of “model pseudo box object” can refer to “subsystem box object “ items for “subsystem box object” which are included inside the model (for instance a derived parameter can be calculated from a STR parameter and a CDMU parameter). Due to SCOS limitation, there is no position allocated per subsystem inside a model, the position given for each element inside the subsystem are also valid at model level.

In addition HPSDB support for each “box object” a “theoretical” definition of the “box object” from which several “real” instances can be derived. At test sequence level this difference between “theoretical” and “real” items is irrelevant. For instance the same test can be run using only “theoretical” part of the model under test or using also the “real” part. The differences between the two result of the same test will be due to the fact in the first configuration only the “theoretical” curves are considered, in the second configuration the “real” curves, if any, will overwrite the theoretical ones.

4.1.2 Generic test sequences

A generic test sequence is a test sequence which includes only references to generic items of HPSDB. It shall not include any reference to element or subsystem or model HPSDB items.

In addition a generic test sequence can include any test sequence which does not refer to any HPSDB item.

A generic test sequence can be called by any other test sequences or by user.

A generic test sequence can call only an other generic test sequence.

4.1.3 Element test sequences

An "element test sequence" is a "test sequence" which is attached to an "element box object" (star tracker for instance) and which includes only references :

- to "element box object items" to which the "test sequence" is attached to,
- to "generic items".

It shall not include any reference :

- to other "element box object item",
- to "subsystem box object item",
- to "model box object item".

An "element test sequence" can be called only by a "subsystem test sequence" or by the user.

An "element test sequence" can call only "generic test sequence".

4.1.4 Subsystem test sequences

A "subsystem test sequence" is a "test sequence" which is attached to a "subsystem box object" (RF subsystem for instance) and which includes only references :

- to "subsystem box object items" to which the "test sequence" is attached to,
- to "element instantiated items" which are part of the subsystem,
- to "generic items".

It shall not include any reference :

- to other "element box object item" than those referenced in the subsystem,
- to other "subsystem box object item",
- to "model box object item".

A "subsystem test sequence" can be called only by model test sequence or by the user.

A "subsystem test sequence" can call "element test sequence" and " generic test sequence".

4.1.5 Model test sequences

A "model test sequence" is a "test sequence" which is attached to a "model box object" (Herschel PFM for instance) and which includes only references :

- to “model box object items” to which the “test sequence” is attached to,
- to “subsystem items” which are part of the model,
- to “generic items”.

It shall not include any reference :

- to other “subsystem box object item” than those referenced in the model,
- to other “model box object item”

A “model test sequence” can be called only by user.

A “model test sequence” can call “element test sequence”, “subsystem test sequence” and “generic test sequence”.

4.2 Test sequence types

Refer to (Figure 4-2 Test sequences types and interfaces)

Four types of test sequences are defined for each level :

- Procedure test sequences,
- Configuration test sequences,
- Measurement test sequences,
- Tool test sequences.

4.2.1 Procedures test sequences

The procedures test sequences are test sequences which are aimed to execute a test at any level. In addition, contrary to other test sequences types, the procedure test sequences have visibility on all the lower level test sequences which allows to re-use them at higher level. Dedicated procedure test sequences are reserved to execute “cross checks”.

A procedure test sequence can call inside the same box object (same level) :

- An other procedure test sequence (can be useful at PFM level to call SVM procedures and PLM procedures),
- A configuration test sequence,
- A measurement test sequence,
- A tool test sequence.

A procedure test sequence can call at any lower level :

- A configuration test sequence,
- A procedure test sequence,
- A measurement test sequence,
- A tool test sequence.

A procedure test sequence can be called by :

- A procedure test sequence of any higher level,
- A procedure test sequence of the same level,
- The user (to execute a test).

4.2.2 Configurations test sequences

The configuration test sequences are test sequences which are aimed to configure a box object. They have no visibility to lower level test sequences.

A configuration test sequence can call inside the same box object (same level) :
- A tool test sequence,

A configuration test sequence can be called by :
- A procedure test sequence of any higher level,
- A procedure test sequence of the same level (inside the same box object),
- The user (to configure a box object manually).

4.2.3 Measurement test sequences

The measurement test sequences are test sequences which are aimed to perform acquisition and checks. They have no visibility to lower level test sequences.

A measurement test sequence can call inside the same box object (same level) :
- A tool test sequence,

A measurement test sequence can be called by :
- A procedure test sequence of any higher level,
- A procedure test sequence of the same level (inside the same box object),
- The user (to perform manual checks).

4.2.4 Tool test sequences

The tool test sequences are test sequences which are aimed to perform some activity common to all other procedures. They have no visibility to lower level test sequences.

A tool test sequence can call inside the same box object (same level) :
- An other tool test sequence.

A tool test sequence can be called by :
- A procedure test sequence of any higher level,
- A procedure test sequence of the same level (inside the same box object),
- A configuration test sequence of the same level (inside the same box object),
- A measurement test sequence of the same level (inside the same box object),
- A tool test sequence of the same level (inside the same box object),
- The user (for fun).

4.2.5 Summary of calls

> ^	Level	Element				Subsystem				Model				Generic			
Level	Type	P	C	M	T	P	C	M	T	P	C	M	T	P	C	M	T
EL.	P	X	X	X	X									X	X	X	X
	C				X												X
	M				X												X
	T				X												X
S/S	P	X	X	X	X	X	X	X	X					X	X	X	X
	C								X								X
	M								X								X
	T								X								X
Mod.	P	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	C												X				X
	M												X				X
	T												X				X
Gen.	P													X	X	X	X
	C																X
	M																X
	T																X

Table 1 - Possible calls between TS

4.3 Naming convention

The name of a test sequence will be the following :

<Subsystem identifier><>Level><Type><Mnemonic><Position>

with :

- <Subsystem identifier> =<Subsystem type><Subsystem number>
  With
  - <Subsystem type> is one character (refer to NMCVT-4081a-C & NMCVT-7500-C)
  - <Subsystem number> is three numerical digits (refer to NMCVT-4081e-C)

- <Level> is one character
  - <Level> = “M” for model level
  - <Level> = “S” for subsystem level
  - <Level> = “E” for element level
  - <Level> = “G” for generic level

- <Type> is one character
  - <Type> = “P” or “X” for procedure test sequence (“X” for cross checks)
  - <Type> = “C” for configuration test sequence
  - <Type> = “M” for measurement test procedure
  - <Type> = “T” for Tool test procedure

- <Mnemonic> is a free 6 characters field
- <Position> is three numerical digits (refer to NMCVT-4081c-C & NMCVT-7520-C)

Examples :

A001ECRW1OFF123
Element procedure of subsystem A001 to switch OFF the Reaction wheel mounted in position 123

S002SMTEMP01528
Subsystem measurement test sequence for SPIRE. Measurement of TEMP01. Note : as this procedure is a subsystem procedure the position is set to a pseudo position of subsystem S.

K256STTOOL01315
Subsystem tool for K (Cryo in case of Herschel, FOG in case of Planck) subsystem.

Z099MMCALCUL990
Model measurement tests sequence. Note : as this test sequence is a model test sequence the subsystem is forced to the pseudo subsystem type ("Z") and the position is set to a system pseudo position ("990").

4.4 Instantiation

MOIS allows to instantiate test sequences. This facility allows for instance to generate from a common source associated to a "box object" (element star tracker for instance) the corresponding test sequences for higher level (Nominal and redundant star tracker at subsystem level).

4.4.1 Instantiation at HPSDB level

The instantiation applies between element and subsystem on one hand and between subsystem and model on the other hand.

The instantiated box object inherits of all the attribute of the source box object. The inherited attributes can be over-written.

The item identifiers are instantiated according to AD3. Note : the identifiers are the same between subsystem and model.

4.4.2 Instantiation at MOIS level

MOIS allows to define test procedure at element level. In this case the procedure is automatically checked against a subset of HPSDB containing only the target element. All the reference to HPSDB items is done using the element identifier of the item according to AD3. The MOIS test sequence name for an element test sequence has the following format (Note : at element level the subsystem identifier is irrelevant, no assumption shall be done on the subsystem which can include this element, and the position is also irrelevant as far as the position refer to the position of the element inside the subsystem) :

<level><type><mnemonic>

- <Level> is one character
  - <Level> = "M" for model level
  - <Level> = "S" for subsystem level
  - <Level> = "E" for element level

- <Type> is one character
  - <Type> = "P" for procedure test sequence
  - <Type> = "C" for configuration test sequence
  - <Type> = "M" for measurement test procedure
  - <Type> = "T" for Tool test procedure

- <Mnemonic> is a free 6 characters field

Examples :

ECRW1OFF
Element test sequence to switch OFF a Reaction wheel.

To instantiate an element procedure, the user shall provide the subsystem identifier (<subsystem identifier>) and the position (<Position>). MOIS then automatically instantiates all the HPSDB objects identifiers embedded inside the element procedure according to AD3. For instance an element parameter M012 referenced in the element procedure ECRW1OFF is instantiated in AM012023 in the instantiated subsystem procedure A001ECRW1OFF023 (refer to first example in 4.3).

As for HPSDB, there is no items identifiers instantiation between subsystem and model level, as consequence there is no instantiation of test sequences between subsystem and model identifiers.

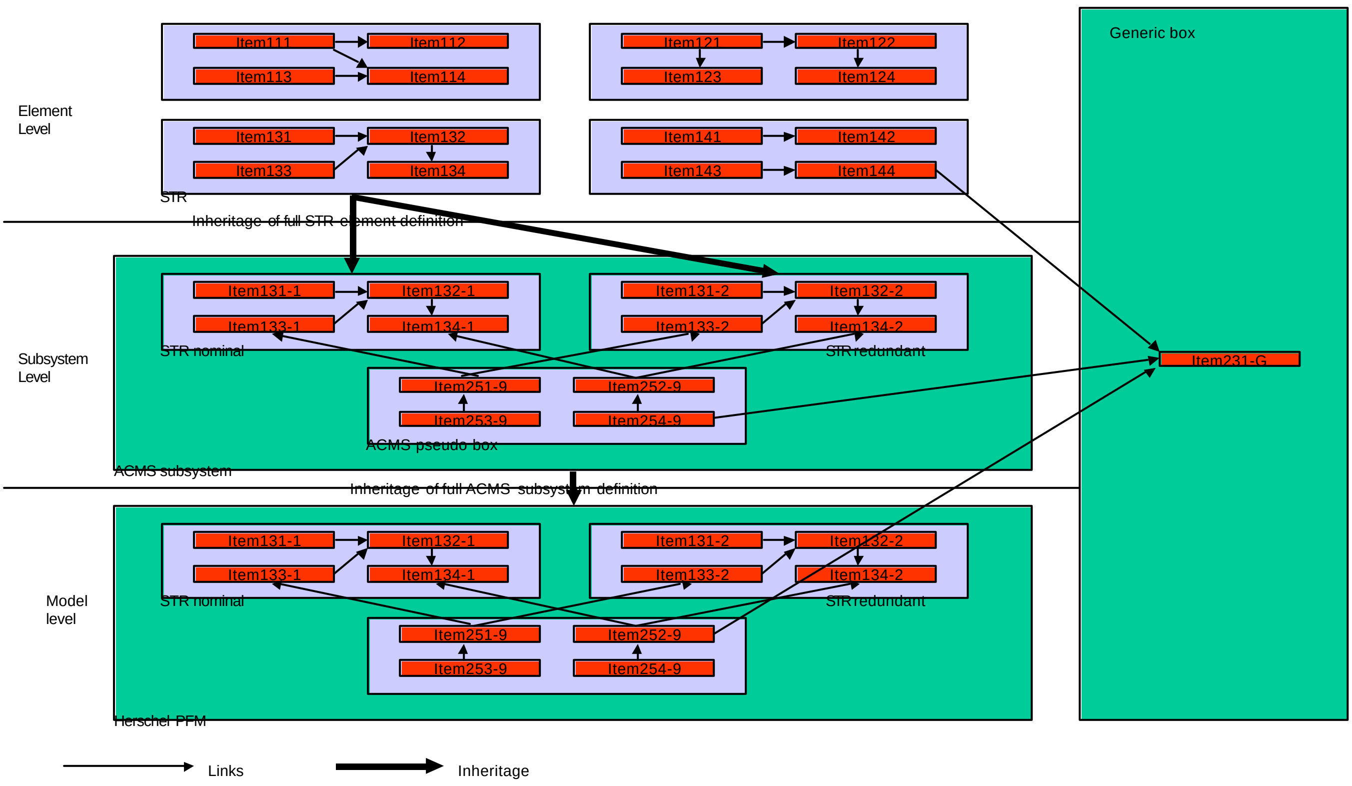


Figure 4-1 Element/subsystem / Model levels - Links and inheritage

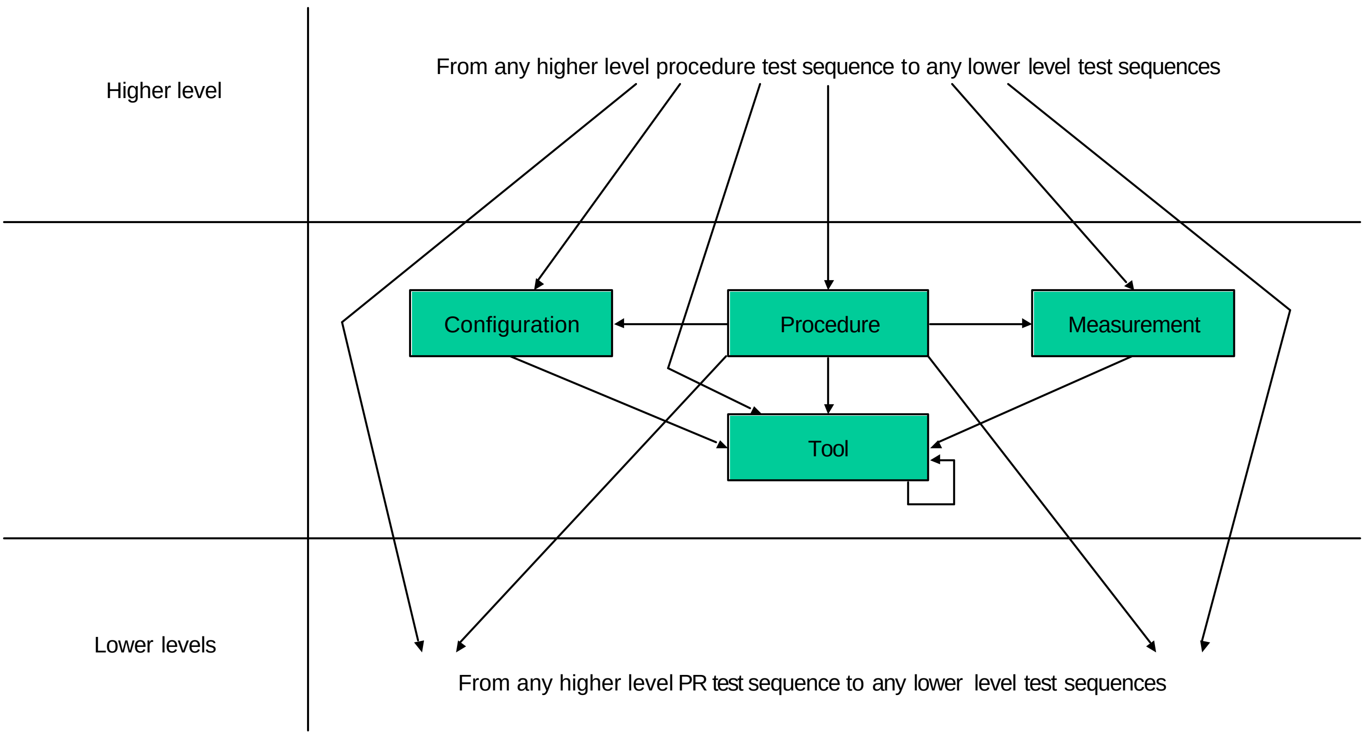


Figure 4-2 Test sequences types and interfaces

5. TEST SEQUENCE INTERFACES

In order to keep trace of all the interfaces between the different test sequences, it is required to use the user dynamic constant (UDC) defined in HPSDB and supported by SCOS. Any other facility (global variable, ...) to exchange data between test sequence shall not be used.

Due to SCOS limitation (unique SCOS packet to support dynamics UDC) allocation have been defined in AD3 (NMCVT-7530-C).

Note : due the limited number of UDC, the length of each UDC has to be optimised.

6. HPSDB INTERFACES

MOIS interfaces with HPSDB via bridge files which are common with CCS ones.

HPSDB allows to extract any box object from any level.

For instance :

- It is possible to extract only the data of a star tracker (element level),
- It is possible to extract only the data of a ACMS subsystem (subsystem level),
- It is possible to extract only the data of the AVM (model level)

Each MOIS servers is able to support several databases extracted from HPSDB. For instance the Alenia MOIS server will support as a minimum : AVM, Herschel SVM and Planck SVM, in addition it can support one data base for each element which can be instantiated.

The element test sequence shall be checked against an extraction of the corresponding element from HPSDB. They cannot be checked against an subsystem or model extraction.

The subsystem test sequence shall be checked against an extraction of the corresponding subsystem from HPSDB. However they can be checked against a model extraction including the referenced subsystem as far as there is no instantiation between subsystem and model level.

A model test sequence shall be checked against an extraction of the corresponding model from HPSDB.

In order to prevent any potential errors, a test sequence resulting of an instantiation shall be checked against the target subsystem or model data base.

The bridge file can be generated from a mirror site, in this case the subcontractor is in charge to manage the interface between the MOIS server and HPSDB mirror site.

The bridge file can be generated from central site, in this case the interface shall be managed according to AD4.

7. TEST SEQUENCES EXCHANGES

Common test sequence can be written. The way to exchange them is TBD (most probably via some MOIS facility : export / import, ...).

8. TEST SEQUENCE CONFIGURATION CONTROL

TBW.

Notes :

- both MOIS and CCS allow to control them.
- It is not excluded that some test sequences will be written outside MOIS (in this case CCS configuration control seems the most appropriate).
- How is done the configuration management on CCS when a test sequence is re-loaded from MOIS inside the CCS ?

9. MOIS CUSTOMISATION FOR AIT

A standard configuration (template) to generate AIT test sequence and associated documentation has to be prepared by ASP.

10. MOIS LIMITATION

This chapter list the limitation, if any, introduced by MOIS against TOPE.

TBW

Note :

- Asynchronous test sequence.

11. FROM AIT TO OPERATION

The initial objective of MOIS is to allow the “smooth transition” between AIT and Operation and to support ESA OIRD requirement “FUMC-7” (“The procedures presented in the SUM shall be validated at spacecraft system and subsystem level testing”).

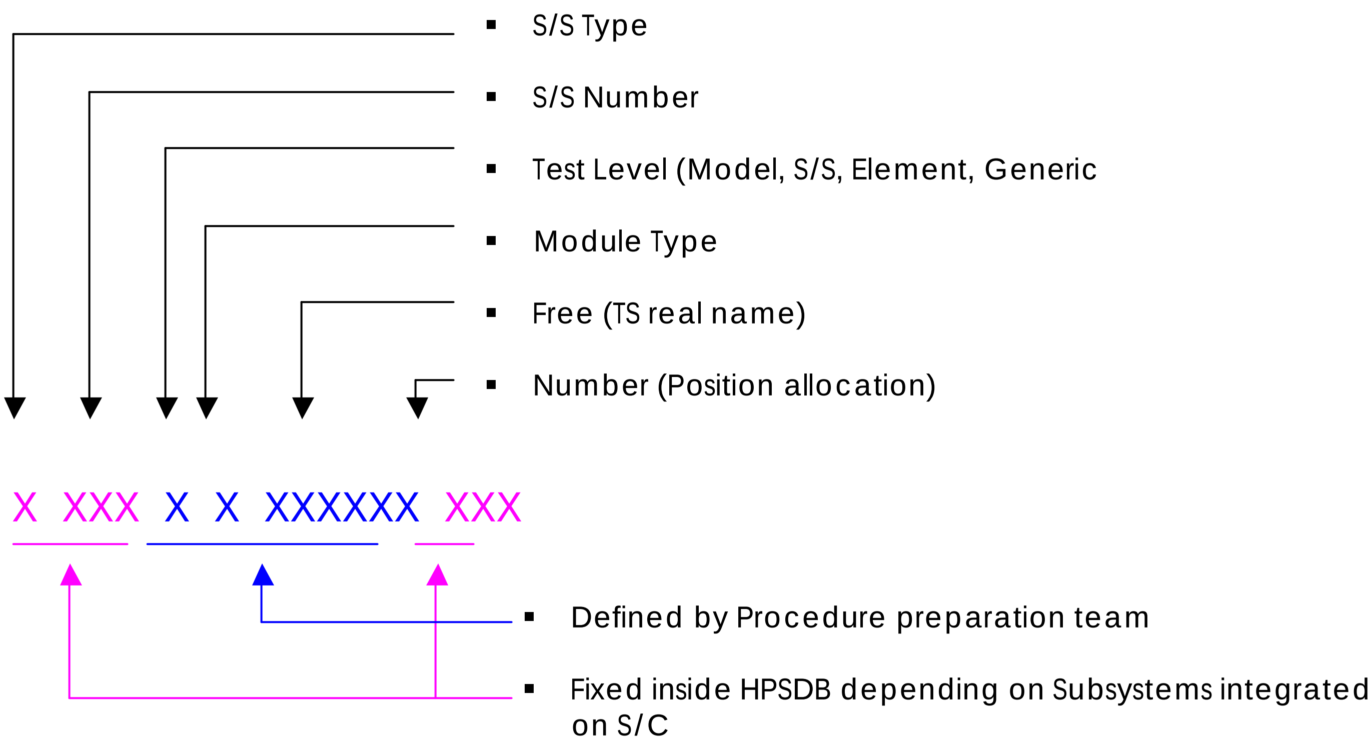
As far as the operation does not intend to use a language, the MOIS procedure shall be converted in a set of command sequences and associated paper procedure.

Detail to be written after MOIS meeting with ESOC planned on 01/10/03.

Some ideas :

- In order that the generation of operation procedures is transparent to AIT users, MOIS will (TBC) generate automatically the command sequences.
- The naming convention of command sequences has to be reviewed.
- How are generated the formal parameters ?
- …..

12. ANNEX 1 : TEST SEQUENCES NAMING EXAMPLES FOR AIT



DPU integration Test Sequence:
H240ECDPUXXX250 “Test Sequence for Switching ON/OFF DPU N”
In fact, the name creation is EDCPUXXX, its instantiation is H240ECDPUXXX250.

Planck HFI S/S Number = 240 (TBC in HPSDB)
DPU N position is 250 and DPU R is 251 (TBC in HPSDB)

Planck S/C Power ON:
J290MPSCON290 “Test Sequence for Switching ON Planck CQM”

Planck S/C Number = 290 (TBC HPSDB)
Planck CQM Position =290 (TBC HPSDB)
Planck AVM would be 291, PFM 292.

13. ANNEX 2 : TEST SEQUENCES AIT RULES

1 Test Sequence does 1 Action
The purpose is to create many basic objects that will be called by a main or Upper Level Test Sequence like in a Tree Structure.

The purpose is:
- to share information between all the Test Sequences,
- to structure and level the Test Sequences,
- to ease Test Sequences update,
- to use instantiation better,
- to avoid information duplication.

The procedures (high level or system test) will be structured in many S/S or equipment Test sequences Calls.

Example 1:

The SubSystem procedure for the whole AOCS Power ON, will call some “Son” Test Sequences:
Element RW Power ON Test Sequence,
Element STR Power ON Test Sequence,
Etc …
(in this case the subsystem procedure is called a “Father” or “Mother” Test Sequence)

In case of RW Power ON TC update:
-only one Test Sequence update is required,
-on the other hand, as many updates as Number of Tests Sequences containing RW power on TC.

Example 2:

A complete procedure with many sub Test Sequences. TBW