HCSS use cases for ILT technical note

Issue 2.8

05 November 2004

K. Galloway

1 Introduction 1
1.1 Purpose and scope of document 1
1.2 Use case description 1
1.3 Use case categorization 1
1.4 Use case format 2
1.5 Actors 3
1.6 References 3
2 ILT test campaign 5
2.1 Overview 5
UCF-700 [Summary]: Perform ILT test campaign 6
3 ILT test preparation 8
3.1 Overview 8
UCF-703 [Summary]: Perform ILT test preparation 10
UCF-707 [User]: Configure the HCSS 12
UCF-756 [User]: Ingest MIB into HCSS 16
UCF-752 [Summary]: Define observing mode, building block or procedure 19
UCF-757 [User]: Define test procedure template 22
4 ILT instrument tests 25
4.1 Overview 25
UCF-701 [Summary]: Execute ILT tests 28
UCF-708 [Summary]: Start the EGSE 30
UCF-709 [Summary]: Start the ILT/ IST HCSS 32
UCF-711 [User]: Run ILT test procedure 34
UCF-758 [User]: Ingest near real-time telemetry during ILT/ IST 38
UCF-601 [User]: Perform RTA 43
UCF-747 [User]: Perform QLA 46
UCF-759 [User]: Ingest telecommand history during ILT phase 49
UCF-763 [User]: Ingest OOL history during ILT phase 52
5 ILT off-line analysis 55
5.1 Overview 55
UCF-706 [Summary]: ILT/ IST off-line analysis 56
6 ILT test telemetry replay 58
6.1 Overview 58
UCF-702 [Summary]: Replay ILT/ IST test telemetry 60
UCF-705 [User]: Run telemetry playback selector 62

1 Introduction

1.1 Purpose and scope of document

In [**AD-1**] various categorizations of the complete set of use cases are presented. One of these categorizations is mission phase. This technical note collects together, and presents a more detailed overview of, the use cases which have been identified for implementation in the HCSS to support the ILT mission phase.

This technical note considers the use cases which will be performed by the ILT HCSS users proper [**AD-2**]. It does not consider those use cases relating to the software engineering users of the system (see [**AD-1**] for a list of these use cases).

Additionally, the operation of external systems (the EGSE-ILT system for example) is only described where it will impact upon the HCSS. To give an example, test control [UCF-711] can use test procedure templates [UCF-757] retrieved from the HCSS for subsequent execution by SCOS 2000. It will also permit the sending of individual telecommand mnemonics to SCOS 2000. The use cases in this technical note do not address the sending of these individual telecommand mnemonics as it will not involve the use of the HCSS.

1.2 Use case description

Use cases provide a means of describing functional requirements in terms of the tasks that various actors wish to perform by using the system under development. Note that these actors may be user roles (for example instrument tester) or external systems (for example EGSE-ILT), anything which interacts with the system under development (the HCSS). The use cases describe how the system will be used, rather than the services it provides. They are at a level of abstraction which avoids implementation issues or the details of user interfaces.

The approach to writing use cases adopted by the CSDT is based on that of Cockburn [**RD-2**] which treats ‘business modelling” in terms of high-level/ summary use cases. The summary-level use cases provide a context for the user-level use cases by showing how these user-level use cases fit together into “business processes”. These summary-level use cases (business processes) typically involve many actors performing their own lower level goals. For example, in order to perform an instrument test [see UCF-701] the instrument engineer must first create a test procedure template. The instrument tester subsequently uses the test procedure template in order to perform the test.

1.3 Use case categorization

For each instrument model the ILT mission phase will be split into campaigns. Within this ILT test campaign concept there are 4 activities which will involve the use of the HCSS. These are:

- ILT preparation
- ILT execution
- ILT off-line analysis
- ILT test telemetry replay

The concept of test campaign and the above 4 sub categories provide a useful mechanism for dividing the ILT use cases and they are further expanded upon in the subsequent sections of this technical note.

1.4 Use case format

Each use case description contained in this technical note has the following format:

Name: The name of the use case is the primary actor's goal (see [**AD-2**]).

Level: Identifies the level of the use case:
Summary-level: a high-level use case.
User-level: implements a user-level goal.
See [**RD-2**] for a detailed description of summary and user-level use cases.

Scope: The scope of the use case. The system with which the external actor interacts.

Version: Version number of the use case definition.

Status: Draft, reviewed, approved.

Brief description: A short description of what the use case does.

Phase: The mission phases for which the use case is applicable.

Actors: Identifies the actors involved in the use case (see [**AD-2**]).

Triggers: The condition which causes the use case to be invoked.

Preconditions: Conditions which must be true before the use case can be used (for example the actor must be logged on).

Minimal guarantees: Conditions guaranteed upon completion (normal or failure) of the use case, provided that the preconditions are true.

Success guarantees: Additional conditions guaranteed upon successful completion of the use case.

Stakeholders and interests: Lists the stakeholders who have an interest in this use case and states what those interests are.

Main success scenario: A sequence of numbered steps that represent the basic scenario leading to a successful outcome. Each step starts with the name of the actor that performs the step. Note that the system with which the actor interacts is also here understood to be an actor. For example: If the astronomer actor (AST) wishes to submit a proposal using the HCSS then the sequence of steps could be:
AST: Submit a science proposal
HCSS: Validate the science proposal
HCSS: Save the science proposal

HCSS: Notify AST that science proposal was successfully saved

Extensions: Describes alternative flows, resulting from variations on the main scenario and exceptional flows which may lead to failure of recovery action. Extensions are numbered to correspond with the steps of the main scenario to which they refer. For example:

- 2a extends step 2 of the main scenario.
- 2a1 first substep of 2a
- 2an nth substep of 2a
- 2-4a may be used for steps 2,3 or 4.
- 2-4a1 first substep of 2-4a
- 2-4an nth substep of 2-4a
- *a may be used to extend any step.
- *a1 first substep of *a
- *an nth substep of *a

Frequency of occurrence: How often the use case happens - in total, rather than for an individual instance of an actor.

Reference: Cross references to requirements in the HCSS URD [**AD-3**] and the FGS IRD [**AD-5**].

Open issues: Issues which need to be resolved before the use case can be finalised.

Comments: Any additional comments.

1.5 Actors

In [**AD-2**] the following actors are identified as “HCSS users proper” during the ILT mission phase:

- Instrument engineer
- Instrument tester
- EGSE-ILT

In some of the use cases presented in this technical note other actors are identified. This is simply because these use cases are applicable to many mission phases, not only ILT. These additional actors are associated with these other mission phases and can be ignored within the context of this technical note.

1.6 References

1.6.1 Applicable documents

- **AD-1**: Herschel Common Science System: Use Case Definitions, Issue 2.1, FIRST/FSC/DOC/0158, 11 April 2003
- **AD-2**: Herschel Common Science System: Actor Descriptions, Issue 2.1, FIRST/FSC/DOC/0157, 11 April 2003
- **AD-3**: HCSS User Requirements Document, Issue 2.0, FIRST/FSC/DOC/0115, 31 August 2001
- **AD-4**: Herschel Common Science System: Open Issues, Issue 2.6, FIRST/FSC/DOC/0159, 29 October 2003

- **AD-5**: Herschel Ground Segment Interface Requirements Document, Issue 2.1, FIRST/FSC/DOC/0117, 12 December 2003
- **AD-6**: HCSS CUS use cases technical note, Issue 0.1, FSCDT/TN-023, 15 April 2004

1.6.2 Reference documents

- **RD-1**: Herschel Operations Scenario Document, Issue 1.2, FIRST/FSC/DOC/0114, 17 March 2002
- **RD-2**: Writing Effective Use Cases, Alistair Cockburn, Addison-Wesley
- **RD-3**: HCSS Glossary of Terms, Issue 1.1, FIRST/FSC/DOC/0120, 15 March 2001
- **RD-4:** HCSS Software Project Management Plan (incorporating FSC Development Roadmap), Issue 4.0, FIRST/FSC/DOC/0116, 14 July 2003

2 ILT test campaign

2.1 Overview

This section contains one use case, UCF-700. UCF-700 is a (cloud-level [**RD-2**]) summary use case showing the top level steps which will be involved in:

- preparing for an ILT test campaign.
- the execution of the tests identified in an ILT test campaign.
- the following up of the results of an ILT test campaign.

It is from this use case that all of the other use cases identified in this technical note descend.

For each instrument model the ILT mission phase will be split into test campaigns. As a rule, each test campaign will start with a test readiness review at which time:

- all test procedures and test scripts must be ready
- the EGSE configuration will be frozen.
- the HCSS configuration will be frozen.

Following a successful review the tests constituting the test campaign will be executed and the data produced stored within the HCSS archive. The end of a test campaign will be marked by a formal post test campaign review.

Use case UCF-700 is seen to consist of the following steps:

- Preparation for the test campaign. This involves the preparation of all items which will be needed in order for the tests to be executed. This will include MIBs, test procedures, QLA procedures and scripts, observation definitions etc. Further details are provided in Section 3.
- A review will precede the actual performing of the tests identified for the test campaign. This test readiness review will ensure that everything is in place for the tests to be performed.
- The actual execution of the tests identified to be performed in the test campaign. Further details are provided in Section 4.
- Analysis of the data produced by the tests. This analysis will be performed off-line in order not to interfere which any test(s) which will be ongoing. Further details are provided in Section 5.
- A post test campaign review which will determine the success or otherwise of the test campaign.

An important extension is also shown in UCF-700. While the EGSE and HCSS are present and all the necessary data is in place (MIB etc.) it will be possible to replay the telemetry of a previous test from the HCSS archive. Further details are provided in Section 6.

UCF-700 [Summary]: Perform ILT test campaign

Level: **summary**
Scope: **EGSE-ILT and HCSS**
Version: **1.0**
Status: **issue**

Brief description:

For each instrument model the ILT will be split into test campaigns. This summary level use case describes the preparation for a test campaign, the execution of the tests identified in the test campaign and the subsequent analysis of the results.

Phase:

Instrument level testing:	Y
Integrated system testing:	N
Ground segment testing:	N
LEOP:	N
Commissioning:	N
Calibration/ PV:	N
Science demonstration:	N
Routine operations:	N
Post operations:	N

Actors:

IE: Instrument engineer (primary actor)
IT: Instrument tester (secondary actor)
ICC: Instrument control centre personnel (secondary actor)

Triggers:

Testing of an instrument is to be performed. Normal work during ILT.

Preconditions:

None.

Minimal postconditions:

None.

Success postconditions:

The test campaign is completed successfully.

Stakeholders and interests:

TBW.

Main success scenario:

1 IE: Perform ILT preparation [UCF-703]
2 ICC: Perform test campaign readiness review (procedural)
3 IT: Execute ILT tests [UCF-701]
4 IE: Perform ILT off-line analysis [UCF-706]
Step 4 can be performed in parallel to step 3 (see comments)
5 ICC: Perform formal post test campaign review (procedural)

Extensions:

2a: ICC: Fail test campaign readiness review
3a: IT: Replay ILT test telemetry [UCF-702]
4a: ICC: Fail formal post test campaign review

Frequency of occurence:

Often during ILT

References:

None.

Open issues:

- None.

Comments:

- There can be several tests associated with a campaign. Off-line analysis of the results of a previous test can be performed while a new test is being run.

3 ILT test preparation

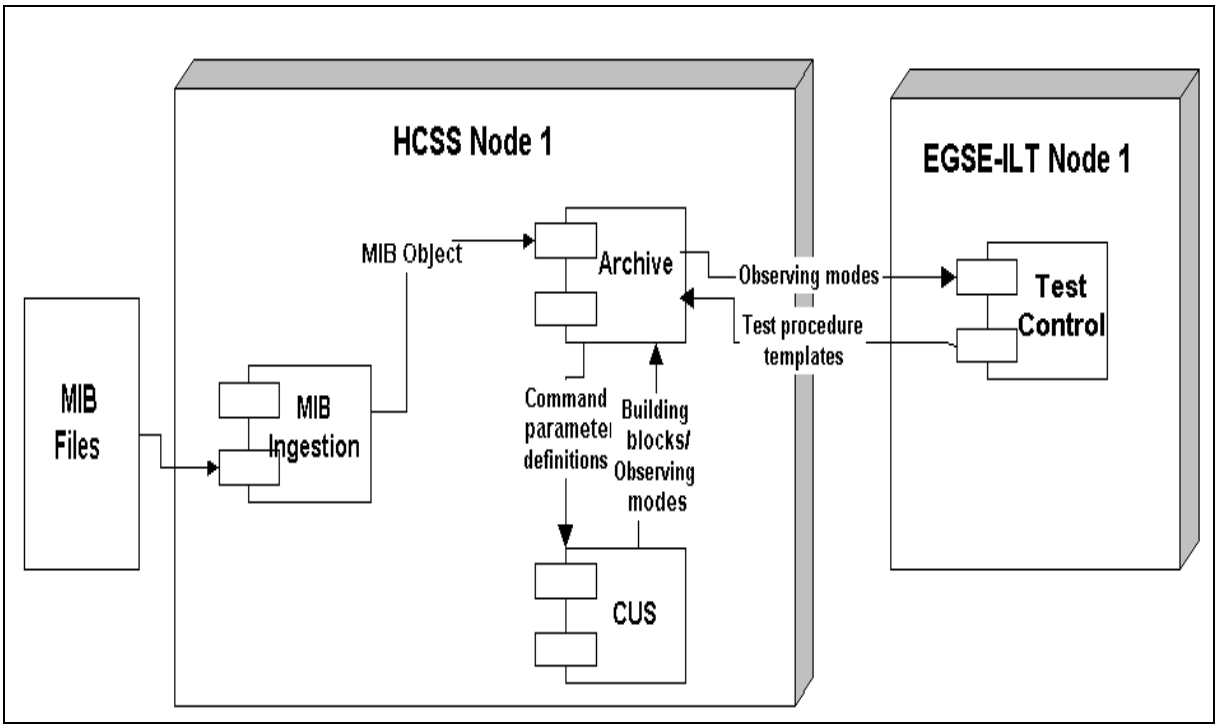
3.1 Overview

This section presents those use cases which detail the steps which will be involved in ILT test preparation.

ILT test preparation will involve the preparation of all items (MIBs, test procedures, QLA procedures and scripts, observation definitions, etc.) which will be required in order to enable the tests identified in a test campaign (see Section 2) to be performed. A test readiness review (see UCF-700) will review these items before deciding whether the test campaign can proceed.

Figure 1 shows the HCSS processes and the EGSE processes (those which interact with the HCSS, see Section 1.1) which will be involved in ILT test preparation.

Figure 1: ILT test preparation process diagram



An instrument specific mission information base (MIB) consists of a collection of tables. A MIB will be made available to the HCSS by an ICC as a collection of files (the “MIB files” shown in Figure 1), one file for each MIB table. These MIB tables/ files will contain the telecommand mnemonic and the telemetry packet definitions for an instrument. Figure 1 shows that the MIB files will be ingested into the HCSS archive for subsequent use by HCSS clients.

The common uplink system (CUS) will subsequently use the telecommand (command) parameter definitions in an ingested MIB object in order to create building blocks and observing modes which will be themselves stored in the HCSS archive.

When a new MIB is to be used that contains changed command definitions the existing observing modes and building blocks must be validated against this MIB and corrected/ updated as necessary. Note that if the updated MIB does not contain modified command definitions then the observing modes and building blocks will remain valid.

Test control is a process belonging to the EGSE-ILT system (external to the HCSS). It will be used to create test procedure templates. These test procedure templates will contain references to the previously defined building blocks and observing modes. Test control will use HCSS APIs to retrieve the building blocks and observing modes from the HCSS archive and to store the created test procedure templates.

At the end of ILT test preparation everything which is required in order to perform the instrument tests defined for the test campaign will be stored in the HCSS archive.

UCF-703 is the summary level use case showing the steps which will be involved in preparing for ILT tests. This use case is seen to consist of the following steps:

- Configure the HCSS [UCF-707]. The HCSS is configured through a mechanism of assigning values to a predefined set of properties. To enable HCSS clients to work these properties must have values assigned to them before the clients are started.
- Ingest MIB into HCSS [UCF-756]. Whenever a new MIB becomes available it will be ingested into the HCSS archive for subsequent access by other HCSS subsystems (for example, the CUS).
- Validate the existing building blocks and observing modes already existing in the HCSS archive against a new MIB [UCF-752] which contains modified command definitions. It will be necessary to maintain the integrity of the building blocks and observing modes in the archive when such a new MIB is introduced. **See also note below.**
- Create new building blocks and observing modes [UCF-752]. Describes the actual steps which will be involved in creating new building blocks and observing modes. **See also note below.**
- Create test procedure template [UCF-757]. This is a user level use case describing the steps which will be involved in defining test procedure templates. This use case has the EGSE-ILT as the primary system and requesting services from the HCSS.

Note: UCF-752 references many sub-use cases. These sub-use cases are not included in this technical note. [**AD-6**] contains the full set of use cases associated with UCF-752 and this document should be consulted if more detailed information is required.

UCF-703 [Summary]: Perform ILT test preparation

Level: **summary**
Scope: **HCSS**
Version: **2.2**
Status: **issue**

Brief description:

This summary level use case describes how the instrument engineer actor prepares for ILT testing. The actor must ensure that the necessary building blocks and observing modes are defined within the HCSS archive. Using the EGSE-ILT system the instrument engineer uses these observing mode definitions to create test procedures templates. These test procedure templates are also stored in the HCSS archive using an HCSS API.

A MIB is ingested into the HCSS archive for use in the creation of building blocks and observing modes. Existing building blocks and observing modes must be validated against a new MIB if the new MIB contains modified command definitions.

Additionally, the use case identifies how a MIB is imported into the HCSS (using a MIB version control/ storage mechanism) and how a required MIB is subsequently exported out of the HCSS when it is required by an external system (the EGSE-ILT system).

Phase:

Instrument level testing:	Y
Integrated system testing:	N
Ground segment testing:	N
LEOP:	N
Commissioning:	N
Calibration/ PV:	N
Science demonstration:	N
Routine operations:	N
Post operations:	N

Actors:

IE: Instrument engineer (primary actor)

Triggers:

Testing of an instrument is to be performed [UCF-700]

Preconditions:

None.

Minimal postconditions:

None.

Success postconditions:

The HCSS archive contains everything necessary to support the ILT tests [UCF-700].

Stakeholders and interests:

TBW.

Main success scenario:

1 IE: Specify HCSS properties [UCF-707]
2 IE: Ingest MIB into HCSS [UCF-756]
3 IE: Import MIB (as a BLOB) into HCSS [UCF-761]
4 IE: Revalidate building blocks/ observing modes (see comments) [UCF-752]
5 IE: Define building block or observing mode [UCF-752]
6 IE: Define test procedure template (EGSE-ILT) [UCF-757]
7 IE: Export MIB (as a BLOB) out of HCSS [UCF-762]

Steps 1 to 7 can be optional, performed in parallel, taken in different orders or repeated.

Extensions:

None.

Frequency of occurence:

Often during ILT

References:

TBW

Open issues:

- None.

Comments:

- Note: Test control can use test procedure templates retrieved from the HCSS for subsequent execution by SCOS 2000. It also permits the sending of individual telecommand mnemonics to SCOS 2000. This use case relates specifically to the use of the test control interface between test control and the HCSS.
- The steps shown in the main success scenario can be optional, can be performed in parallel, can be taken in different orders, and can be repeated. For example:
  - Several MIBs may need to be held in the database.
  - A MIB can be exported at the same time as building blocks and observing modes are being defined.
  - A previously stored MIB can be exported while a new MIB is being imported/ ingested.
  - A MIB could be exported before building blocks/ observing modes are defined.
  - If a new MIB does not contain modified command definitions then it will not be necessary to perform building block/ observing mode revalidation.
- Export MIB (as a BLOB) out of HCSS for use in the EGSE environment. RTA analyses the housekeeping telemetry packets. It must have access to the MIB which was used in the generation of the telecommands which are to be sent to the instrument.
- UCF-752: This use case allows both the definition of new observing modes and building blocks and the validation of existing building blocks against an updated MIB.

UCF-707 [User]: Configure the HCSS

Level: **user**
Scope: **HCSS**
Version: **1.2**
Status: **issue**

Brief description:

The HCSS will use predefined properties to obtain needed input data such as database name and location. The HCSS property mechanism works by reading the property values from a sequence of property files specified by the HCSS user.

There will be those HCSS properties that have been pre-assigned values by the HCSS system administrators and those properties to which the HCSS users are allowed to assign values. This use case details the steps that the HCSS user will perform when assigning values to HCSS properties.

Phase:

Instrument level testing:	Y
Integrated system testing:	Y
Ground segment testing:	Y
LEOP:	Y
Commissioning:	Y
Calibration/ PV:	Y
Science demonstration:	Y
Routine operations:	Y
Post operations:	Y

Actors:

This use case is applicable to all the actors identified as users of the HCSS. The generic term USR (for user) is used in the “Main success scenario” and “Extensions” sections below.

Triggers:

Perform ILT test preparation [UCF-703].
Prepare for instrument specific IST tests [UCF-712].

Preconditions:

None.

Minimal postconditions:

None.

Success postconditions:

The assigned property values are successfully save to file.

Stakeholders and interests:

TBW.

Main success scenario:

1 USR: Start the HCSS configuration process

1.1 HCSS: Load HCSS property definitions and default values (see comments)
1.2 HCSS: Check that each property is defined only once
1.3 HCSS: Load users sequence of property files (see comments)
1.4 HCSS: Display property definitions and values from users sequence of prop. files

2 USR: Load a specific property file
2.1 USR: Request to load a specific property file
2.2 HCSS: Open specified property file and load property values
2.3 HCSS: Display property value against each property

USR: Repeat step 2 selecting alternative paths until loading of property values is complete

3 USR: Edit property value
3.1 USR: Select property value to edit and supply new value
3.2 HCSS: Display new value

USR: Repeat step 3 selecting alternative paths until editing is complete
USR: Repeat steps 2-3 selecting alternative paths until loading and editing is complete

4 USR: Save properties to a specified file
4.1 USR: Request to save properties to a specified file
4.2 HCSS: Validate value associated with each property
4.3 HCSS: Save properties to specified file

USR: Repeat step 4 selecting alternative paths until saving is complete
USR: Repeat steps 2-4 selecting alternative paths until loading, editing and saving is complete

5 USR: Exit
5.1 USR: Request to exit
5.2 HCSS: Check that no unsaved changes
5.3 HCSS: Terminate client

Extensions:

1.2a HCSS: Detect that a property has been defined more than once
1.2a1: HCSS: Issue a warning message to the user
1.2a2: HCSS: Prompt user to either continue or to abort

2.3a HCSS: A value is not defined for a property
2.3a1: HCSS: Display “NOT DEFINED” against the property

2a USR: Load prop values from user’s sequence of property files (see comments)
2a1 USR: Request to load property values from users sequence of property files
2a2 HCSS: Load property values from users sequence of property files
2a3 HCSS: Display property value against each property

2b USR: Request to create a new property file
2b1 HCSS: Display HCSS default value for each property

2c USR: Request to include the contents of a specified property file (see comments)
2c1 HCSS: Open specified property file and load property values
2c2 HCSS: Overwrite old property values with new property values and display

2d USR: Request to remove all loaded definitions
2d.1 HCSS: Set all property values to not defined.

3a USR: Insert new variable (see comments)
3a1 USR: Request to insert new variable
3a2 HCSS: Prompt user for variable name and value
3a3 USR: Supply variable name and value
3a4 HCSS: Display variable name and value with other variables

3b USR: Expand variable to its current value (see comments)
3b1 USR: Request to expand variable to its current value
3b2 HCSS: Display current values for variables
3c USR: Request to re-display variables
3d USR: Set all values that are not defined to their default values
3d1 USR: Request to set all values that are not defined to their default values
3d2 HCSS: Set all values that are not defined to their default values
3e USR: Display property name
3e1 USR: Request to show the name of the property
3e2 HCSS: Toggle between property name and property title.

3b2a HCSS: Variable is not defined
3b2a1 HCSS: Display error message

4a USR: Save properties to last file in users sequence of property files (see comments)
4a1 USR: Request to save properties to last file in users sequence of property files
4a2 HCSS: Validate value associated with each property
4a3 HCSS: Append properties to the end of last file in users sequence of property files
4b USR: Save only modified properties to a specified file
4b1 USR: Request to save only modified properties to a specified file
4b2 HCSS: Validate value associated with each property
4b3 HCSS: Save modified properties to specified file
4c USR: Save modified properties in the file they were read from (see comments)
4c1 USR: Request to save modified properties in the file they were read from
4c2 HCSS: Validate value associated with each property
4c3 HCSS: Save modified properties in the file they were read from

4.2a-4b2a HCSS: Validation fails
4.2a1-4b2a2 HCSS: Log error message and abandon save
4.3a HCSS: Could not save properties to specified file
4.3a1 HCSS: Log error message and abandon save
4b3a HCSS: Could not save properties to specified file
4b3a1 HCSS: Log error message and abandon save

4c2a HCSS: Properties were not read from a file (default values being used)

Frequency of occurence:

It is anticipated that each user of the HCSS will perform this step once in order to set up his basic environment. Subsequent updates are anticipated to be infrequent.

References:

Open issues:

- Do we need to mention required property views or is it an implementation detail.

Comments:

- HCSS property definitions and default values: The HCSS developers define the properties that are to be used and specify possible values and default value for each of these properties. These property definitions are not to be confused with the property files created by

the HCSS users. The property files contain the values assign by the the HCSS user to these properties.

- User's sequence of property files: The HCSS users must define a sequence of property files that he wants the HCSS to read when loading property values.
- It is possible that within this sequence of property files a property has been assigned a value more than once. In these circumstances the HCSS uses the last assigned value.
- Save properties to last file in the users default sequence of property files: The values assigned to the properties are appended to the end of the last file in the users default sequence of files. This ensures that when the HCSS loads these property values it will take these values.
- Save properties to file they were read from: The values assigned to the properties over-write the old values.
- Include contents of property file: Replace property values with those contained in the specified property file.
- Variables: Instead of assigning a value to a property a variable may be specified. This is useful where the same value needs to be assigned to many properties. The user can toggle between viewing the variable against the properties or the current value of the variable.

UCF-756 [User]: Ingest MIB into HCSS

Level: **user**
Scope: **HCSS**
Version: **2.7**
Status: **issue**

Brief description:

This use case describes how a mission information base (MIB) will be ingested into the HCSS.

The ICCs will update the telecommand mnemonic (command) definitions and/ or the telemetry packet definitions in the MIB using ICC functionality. It will then be necessary to import the new MIB into the HCSS for use by other HCSS subsystems.

The MIB to be ingested will consist of a number of tables, each contained in a separate file. The ingestion process will create a new MIB object containing the data described in these tables.

Phase:

Instrument level testing:	Y
Integrated system testing:	N
Ground segment testing:	N
LEOP:	N
Commissioning:	N
Calibration/ PV:	N
Science demonstration:	N
Routine operations:	N
Post operations:	N

Actors:

IE: Instrument engineer

Triggers:

ILT: A new MIB is available for ingestion into the HCSS [UCF-703].

Preconditions:

The IE has access to the MIB import functionality (ILT)

Minimal postconditions:

The newly imported MIB is the most recent of a collection of versionable MIB entities.

Success postconditions:

The MIB, in its imported form, can be used by the appropriate HCSS modules as required. The MIB can be re-exported to the exchange format required by the MIB editor.

Stakeholders and interests:

ICCs: The required MIB is available within the HCSS in a timely manner.

Main success scenario:

1 IE: Start MIB ingestion

1.1 IE: Configure MIB ingestion
1.1.1 IE: Identify MIB location
1.1.2 IE: Identify database in which MIB is to be stored
1.2 IE: Invoke MIB ingestion process
2 HCSS: Start MIB ingestion process
2.1 HCSS: Validate MIB (see comments)
2.1.1 HCSS: Validate MIB structure.
Repeat step 2.1.1 until all MIB structures validated
2.1.2 HCSS: Validate field type/ range
Repeat step 2.1.2 until all field type/ range validation performed
2.1.3 HCSS: Validate table cross-references
Repeat step 2.1.3 until all table cross references performed
2.2 HCSS: Ingest MIB
2.2.1 HCSS: Create MIB object
2.2.2 HCSS: Establish connection with identified database
2.2.3 HCSS: Get instrument model associated with the MIB from the database
2.2.4 HCSS: Associate MIB object with instrument model (see comments)
2.2.5 HCSS: Store MIB object in the database

Extensions:

2.1.1a HCSS: Structural validation fails.
HCSS: Log error/ warning information
2.1.2a HCSS: Type/ range validation fails.
HCSS: Log error/ warning information
2.1.3a HCSS: Cross-reference checks fail.
HCSS: Log error/ warning information
2.2.1a HCSS: MIB object creation fails.
HCSS: Log error information and terminate ingestion process
2.2.2a HCSS: Connection to database fails
HCSS: Log error information and terminate ingestion process
2.2.3a HCSS: Unable to locate instrument model in database
HCSS: Log error information and terminate ingestion process
2.2.5a HCSS: MIB storage fails.
HCSS: Log error information and terminate ingestion process

Frequency of occurence:
Frequently during ILT.

References:

SCOS-2000 database import ICD (S2K-MCS-ICD-0001-TOS-GCI).
HCSS MIB clarification and tailoring document (HSC/DOC/0300)

HCSS-UR-3.1-0840 (partial)

Open issues:

None.

Comments:

- The MIB to be ingested is assumed to be in ASCII form, one file per MIB table, and to have a structure in accordance with the SCOS-2000 database import ICD (S2K-MCS-ICD-0001-TOS-GCI).
- Every instrument model will have its own MIB. Due to the ageing of the instrument and improvements in instrument understanding there will be several versions of the MIB during the life time of an instrument model.
- Several MIBs will be applicable at a given point in time, one for each instrument model [**AD-4:** Open issue 2.12.8].
- Configure MIB ingestion: The MIB structure definition files are held locally, not in the database. These files are used by the MIB ingestion process when performing MIB validation.
- Validate MIB: The MIB ingestion process validates the whole MIB and then terminates if errors have been detected. That is, the “Ingest MIB” step is not performed when errors are detected.
- Validate MIB: Cross reference checks are performed. The actual checks performed are detailed in the HCSS MIB clarification and tailoring document (HSC/DOC/0300)
- Validate MIB structure: This ensures that the number of fields in each record of each table file is correct, and that the entries for each field are valid. The MIB structure definition files are used by the MIB ingestion process for this purpose.
- This use case plays a role in UCF-005 and UCF-006 because it is required that the deployment of a new MIB is properly coordinated with related updates of the system.
- Only a subset of a MIB is expected to be imported. The exact details of which parts of the MIB are ingested are defined in the HCSS MIB clarification and tailoring document (HSC/DOC/0300) [**AD-4:** Open issue 2.12.8].
- Associate MIB object with instrument model: Only when the MIB has been validated against the CUS definitions (see UCF-752) will the MIB become “usable”.

UCF-752 [Summary]: Define observing mode, building block or procedure

Level: **user**
Scope: **HCSS**
Version: **3.11**
Status: **issue**

Brief description:

This use case describes how an instrument engineer will use the HCSS to specify an observing mode, building block or procedure.

In the following description, *definition* refers to an observing mode, building block or procedure.

It is already identified that the HCSS functionality which will be used by these actors to define a mode/ block/ procedure is the common uplink subsystem (CUS). The term CUS therefore appears in details of the steps.

Phase:

Instrument level testing:	Y		
Integrated system testing:	Y	Call for key project proposals	Y
Ground segment testing:	Y	Call for guaranteed time proposals	Y
LEOP:	Y	Call 1 for open time proposals	Y
Commissioning:	Y		
Calibration/ PV:	Y		
Science demonstration:	Y		
Routine operations:	Y		
Post operations:	N		

Actors:

IE: Instrument Engineer (primary actor)

Triggers:

ILT: Testing of an instrument is to be performed [UCF-700] [UCF-703].
IST: Integrated system testing is to be performed [UCF-710][UCF-712]
An observing mode, building block or procedure needs to be defined
An observing mode, building block or procedure needs to be modified

Preconditions:

Existing observing modes, building blocks and procedures have been validated [UCF-704].

Minimal postconditions:

The creation of a new observing mode/ building block/ procedure does not invalidate already existing definitions.

Success postconditions:

The defined observing mode/ building block/ procedure is available to the instrument engineer.

Stakeholders and interests:

TBW

Main success scenario:

1 IE: Start CUS editor
- 1.1 IE: Request to start CUS editor
- 1.2 HCSS: Get instrument whose definitions are to be updated (see comments).
- 1.2.1 HCSS: Prompt user to identify instrument
- 1.2.2 IE: Identify instrument
- 1.2.3 HCSS: Get identified instrument
- 1.3 HCSS: Get instrument model from the database
- 1.4 HCSS: Display user interface
- 1.5 HCSS: Populate CUS registry with definitions in the archive
- 1.6 HCSS: Populate CUS registry with inst. command definitions (see comments)

2 IE: Import definitions from file [UCF-752A]

IE repeats step 2, selecting alternative files as necessary, until all needed definitions loaded

3 IE: Edit (new or previously created) definitions [UCF-752B]

IE repeats step 3, selecting alternative paths as necessary, until definition editing completed

4 IE: Save definitions [UCF-752H]

IE repeats steps 2-4, selecting alternative paths as necessary, until CUS session finished

5 IE: Stop CUS editor
- 5.1 IE: Request to stop CUS editor
- 5.2 HCSS: Check that definition has been saved

HCSS repeats step 5.2 for all definitions
- 5.3 HCSS: Terminate CUS editor

Extensions:

1.2a: HCSS: Get user predefined instrument (see comments)

1.2.2a: IE: Select to cancel use of CUS editor

1.2.2a1: HCSS: Terminate CUS editor

1.3a HCSS: Instrument model could not be found

1.3a1 HCSS: Signal error message and terminate

3a IE: Delete definition (see comments) [UCF-752C]
3b IE: Rename definition [UCF-752D]
3c IE: Validate definition [UCF-752E]
3d IE: Calculate duration of definition [UCF-752F]
3e IE: Generate command sequence from definition [UCF-752G]
3f IE: Create SPIRE command list building block [UCF-752I]
3g IE: Select updated MIB [UCF-752J]
3h IE: View definition history [UCF-752K]
3i IE: Calculate noise associated with definition [UCF-752L]
3j IE: Calculate data rate associated with definition [UCF-752M]
3k IE: Use calibration tables [UCF-752N]
3l IE: View differences between versions of a definition [UCF-752O]

5.2a HCSS: Unsaved definition detected

5.2a.1 HCSS: Detect unsaved definition

5.2a.2 HCSS: Prompt IE to save definition, reject definition or cancel CUS stop request

Frequency of occurence:

Frequently during ILT reducing in frequency from IST through routine phase.

References:

HCSS-UR-3.1-0840 (partial)
HCSS-UR-3.1-1310
HCSS-UR-3.1-1320
HCSS-UR-3.1-1330
HCSS-UR-3.1-1350

Open issues:

- None.

Comments:

- The latest version of the building block will be used if you specify a building block in the CUS language [**AD4**: Open issue 2.12.7].
- Delete definition: Definitions cannot be deleted (as they may already be in use by observation requests). They are simply flagged as "deleted" so that they cannot be used in new observation requests [**AD4**: Open issue 2.12.7].
- Aside: Building blocks are identified within the telemetry using BBIds [**AD4**: Open issue 2.12.7].
- There is no CUS functionality required to support on-board control procedures [**AD4**: Open issue 2.12.7].
- On-board control procedures can be tried out as procedures.
- CUS language constructs: The CUS scripting language supports all the functions listed in the CUS language guide (ftp://astro.esa.int/pub/HERSCHEL/csdt/releases/doc/cus/guide/cus_language.html)
- Support for multiple users of the CUS will be present in the HCSS to support ILT.
- A single observing mode is applicable to more than one subsystem. For example, instrument and spacecraft [**AD-4**: Open issue 2.2.21].
- Building blocks containing telecommands relating to one subsystem or to more than one subsystem is purely a matter of definition/ implementation detail [**AD-4**: Open issue 2.2.21].
- Instrument identification: The user can pre-identify the instrument whose definitions are to be updated. If the instrument has not been identified then the CUS will prompt the user.
- Both the CUS definitions relates and MIB versions relate to a specific version of an instrument model [**AD-4**: Open issue 2.12.12].
- Versioning: Definitions always call the latest version of another definition. Although the system records the history, earlier versions cannot be called explicitly [**AD-4**: Open issue 2.12.12].

UCF-757 [User]: Define test procedure template

Level: **user**
Scope: **EGSE-ILT, Instrument-EGSE**
Version: **2.1**
Status: **issue**

Brief description:

This use case describes how the EGSE-ILT/ Instrument-EGSE will interact with the HCSS to enable a calibration scientist, instrument tester or instrument engineer to specify a test procedure template.

The HCSS participates in this use case as a secondary actor. Only those interactions that involve the HCSS are detailed in this use case. The detail of the interactions of the calibration scientist, instrument tester or instrument engineer with the EGSE-ILT/ Instrument-EGSE are not specified.

ILT: A test procedure (execution) is generated by supplying the required input parameters to a selected test procedure template during the run test procedure use case (UCF-711).

Phase:

Instrument level testing:	Y
Integrated system testing:	Y
Ground segment testing:	N
LEOP:	N
Commissioning:	N
Calibration/ PV:	N
Science demonstration:	N
Routine operations:	N
Post operations:	N

Note: All phases (except post operations) are shown as applicable as it will potentially be necessary to carry out test procedures on the flight spare instrument in order to help characterise instrument behaviour and diagnose problems.

Actors:

CS: Calibration scientist (primary)
IE: Instrument engineer (primary) (combined with CS)
IT: Instrument tester (primary)
HCSS: Herschel common science system (secondary)

The generic term USR (for user) is used in the “Main success scenario” and “Extensions” sections below. When reading replace USR by IT, IE or CS as appropriate.

Triggers:

ILT: Testing of an instrument is to be performed [UCF-700] [UCF-703].
A test procedure template needs to be defined.
A test procedure template needs to be modified.

Preconditions:

Observing modes have been defined [UCF-703][UCF-752].

Minimal postconditions:

The integrity of the test procedure library is maintained

Success postconditions:

The test procedure definition is generated and saved in the HCSS archive.

Stakeholders and interests:

TBW

Main success scenario:

1 USR: Request the creation of a new test procedure

USR/ EGSE-ILT interact without HCSS participation

2 USR: Request insertion of observing mode into test procedure

3 EGSE-ILT: Request list of available observing modes and associated information

4 HCSS: Supply observing mode list

5 EGSE-ILT: Display observing mode list

6 USR: Select observing mode and request insertion

7 EGSE-ILT: Insert observing mode into test procedure

USR/ EGSE-ILT interact without HCSS participation

Repeat steps 1 through 7 until satisfied

8 USR: Make the test procedure persistent

9 USR: Specify for which instrument models the test procedure is valid

10 EGSE-ILT: Request save test procedure

11 HCSS: Save test procedure

Extensions:

1a USR: Request an existing test procedure

1a1 EGSE-ILT: Request retrieval of test procedure

1a2 HCSS: Provide test procedure

Frequency of occurence:

Frequently during ILT reducing in frequency from IST through routine phase.

References:

HCSS-UR-3.1-0840 (partial)
HCSS-UR-3.1-1330 (Not compliant)
FGS-IR-4.1-05
FGS-IR-4.1-25

Open issues:

- Is it required to somehow ‘validate’ a test procedure template? For example by having a ‘dry run’, i.e., generating the command mnemonics without really executing them or making anything permanent? This seems rather difficult for scripts where the flow depends on outside data (telemetry/operator). Fixed sequences for which the validation is important should probably be written as building blocks [AD-4: Open issue 2.12.13].

Comments:

- Test control will use test procedure templates retrieved from the HCSS for subsequent execution by SCOS-2000. It will also permit the sending of individual telecommand mne-

monics to SCOS-2000. This use case relates specifically to the use of the test control interface between test control and the HCSS.

- There are two basic actions required from the HCSS: to supply the editor with the signatures of the existing observing modes and to save the final template.
- This use case requires a test procedure library in the class model to store the test procedure templates.
- Validation of a test procedure definition is EGSE-ILT responsibility. The HCSS does not validate before saving.
- The deletion of an observing mode from a test procedure definition does not require interaction with the HCSS.
- The modification of an observing mode in a test procedure definition does not require interaction with the HCSS.

4 ILT instrument tests

4.1 Overview

This section presents those use cases which detail the performance of the set of instrument tests identified for a test campaign. The preparation for the tests will have been completed (see Section 3) and the systems (EGSE and HCSS) necessary for the performance of the tests will be started and configured. For each test, the instrument and test environment will be configured. A test will be executed. After each test the test completion procedures will then be performed. This will involve returning the systems, the instrument and the test hardware to a nominal state and the backing up/ archiving of the test data/ results. Finally, when all the tests have been completed, the test campaign completion procedures will be performed.

Figure 2 shows the processes which will be involved in performing a single instrument test. It is noted that the main system that will be used when performing an instrument test will be the EGSE-ILT system with the HCSS participating as a secondary actor.

EGSE-ILT test control will first retrieve the requested test procedure template from the HCSS. The instrument tester actor will supply parameters for each of the observing modes referenced in this test procedure template.

The instrument tester actor will then use test control to initiate test procedure execution. In accordance with the logic defined in the test procedure each observing mode reference/ mnemonic and associated parameters will be passed back to the HCSS. From this information the HCSS will generate the command sequence (sequence of telecommand mnemonics and associated parameters) for the observing mode and will pass it to test control. Test control will forward the command sequence to the SCOS 2000 commanding process. SCOS 2000 will convert the command sequence to actual telecommand packets which will be forwarded, via the EGSE-ILT router, to the EGSE-ILT test environment. In response to the telecommand packets the instrument (not shown in Figure 2) and the test environment will produce telemetry.

The generated telemetry will be sent to the EGSE-ILT router which will forward it to the registered telemetry clients. In Figure 2 there are 3 EGSE-ILT router telemetry clients can be seen:

- Telemetry ingestion. The telemetry ingestion process will store the telemetry packets in the HCSS archive and will convert the science telemetry packets into data frames which it will also store.
- Telemetry gateway. This is an external process which will be responsible for forwarding the housekeeping telemetry packets to RTA for analysis.
- QLA. The QLA process will analyse, in near real time, the instrument science data. It will use the HCSS access API to connect to the router and to perform the necessary conversion of the incoming science packets to data frames. QLA products will be generated and will be stored in the HCSS archive (TBC for ILT).

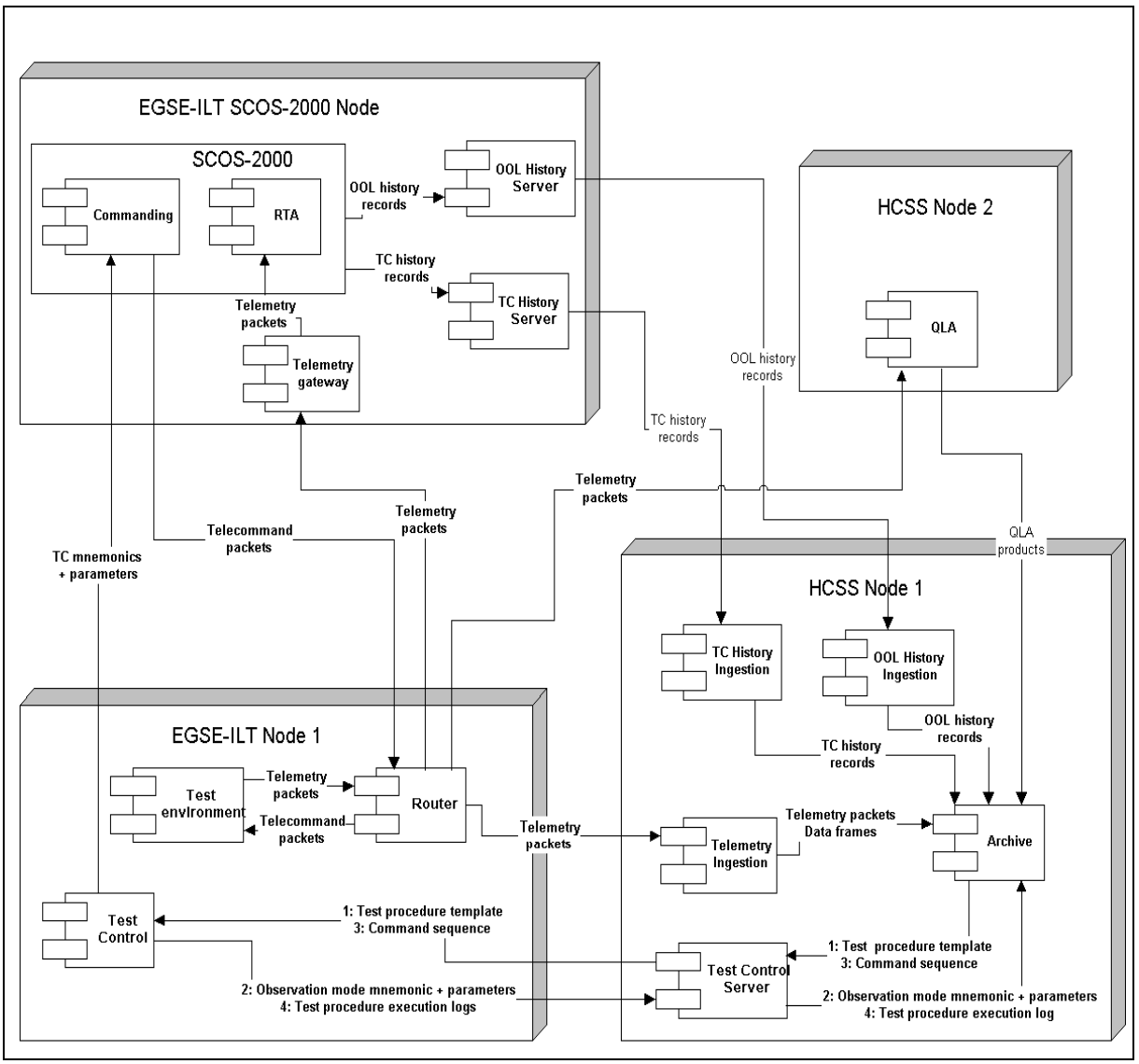
RTA will also be responsible for making available to the HCSS:

- the telecommand history (created by the commanding process)
- the RTA out of limits reports.

The HCSS telecommand history ingestion and out of limit ingestion processes will retrieve these and store them in the HCSS archive.

UCF-701 is the summary level use case showing the steps which will be involved in executing a set of instrument tests. This use case is seen to consist of the following steps:

Figure 2: Perform instrument test process diagram



- Starting the EGSE processes [UCF-708]. As shown in Figure 2 this will include test control, the EGSE router and so on.
- Starting the HCSS processes [UCF-709]. As shown in Figure 2 this will include the telemetry ingestion process, telecommand history ingestion and so on.
- For each test the instrument and test environment is configured.

- Each test is performed/ executed. The test procedure for the test is run [UCF-711], the generated telemetry is ingested into the HCSS archive [UCF-758] and the resulting telemetry is processed by RTA [UCF-601] and QLA [UCF-749].
- After the test is performed the instrument and test environment is returned to a nominal configuration.
- The archive will be mirrored to an off-line archive in order to allow off-line analysis to be performed.
- At regular intervals the telecommand history [UCF-759] and the out of limits history [UCF-763] are ingested into the HCSS.

UCF-701 [Summary]: Execute ILT tests

Level: **summary**
Scope: **EGSE-ILT and HCSS**
Version: **2.0**
Status: **issue**

Brief description:

This use case describes the steps that will be involved in performing a series of tests on an instrument. This includes the starting of the systems involved in the tests, the running of the tests, and the monitoring and analysis of the results from these tests.

Phase:

Instrument level testing:	Y
Integrated system testing:	N
Ground segment testing:	N
LEOP:	N
Commissioning:	N
Calibration/ PV:	N
Science demonstration:	N
Routine operations:	N
Post operations:	N

Actors:

IE: Instrument engineer (primary actor)
IT: Instrument tester (secondary actor)

Triggers:

ILT testing of the instrument is to be performed [UCF-700].

Preconditions:

ILT test preparation has been performed [UCF-703].

Minimal postconditions:

None.

Success postconditions:

The tests identified for the test campaign are executed and the resulting telemetry analysed.

Stakeholders and interests:

TBW.

Main success scenario:

1 IE: Start the EGSE [UCF-708]
2 IE: Start the HCSS [UCF-709]
3 IE: Setup hardware in test configuration (EGSE-ILT)
4 IT: Execute test
  4.1 IT: Run ILT test procedure (EGSE-ILT) [UCF-711]
  4.2 Router: Receive telemetry packet and forward to registered clients (EGSE-ILT)
  4.3 Telemetry ingestion: Receive telemetry and process it (HCSS) [UCF-758]

4.4 RTA: Receive telemetry and process it (EGSE-ILT) [UCF-601]
4.5 QLA: Receive telemetry and process it (HCSS) [UCF-747]
Repeat steps 4.2 to 4.5 until test completes
5 IE: Perform test completion process
5.1 IE: Set hardware to nominal mode
5.2 IE: Mirror HCSS archive to an off-line archive
Repeat steps 3 to 5 until all tests in the campaign are completed
6 HCSS: Ingest TC history into HCSS [UCF-759]
7 HCSS: Ingest out of limits history into HCSS [UCF-763]
Steps 6 and 7 are performed at regular intervals throughout the test campaign (see comments).
8 IE: Shutdown the HCSS and EGSE
9 IE: Shutdown the instrument and EGSE hardware

Extensions:
None.

Frequency of occurence:
Often during ILT

References:
HCSS-UR-3.1-0840 (partial)

Open issues:
- None.

Comments:
- Setting the system to a nominal mode will involve, for example, leaving the instrument in the cryostat but not actively being used.
- Mirroring the HCSS archive to an off-line archive ensures that off-line analysis can take place in parallel to the actual testing [see UCF-700].
- Automatic analysis of the test results (telemetry) is not required for v 0.1 of the HCSS [**AD-4**: Open issue 2.12.5].
- Running QLA/IA requires that the proper algorithms and procedure are present in the HCSS [**AD-4**: Open issue 2.12.5].
- TC history ingestion and OOL history ingestion are continuously running processes which will checks for updates to the TC history and the OOL history at regular intervals.

UCF-708 [Summary]: Start the EGSE

Level: **summary**
Scope: **EGSE**
Version: **2.0**
Status: **issue**

Brief description:

This use case details the steps that will be involved in starting the EGSE processes prior to ILT or IST tests or the playback of instrument test telemetry resulting from these tests.
In the ILT mission phase the EGSE system is referred to as the EGSE-ILT. In the IST mission phase the EGSE system is referred to as the Instrument-EGSE.

Phase:

Instrument level testing:	Y
Integrated system testing:	Y
Ground segment testing:	N
LEOP:	N
Commissioning:	N
Calibration/ PV:	N
Science demonstration:	N
Routine operations:	N
Post operations:	N

Actors:

IT: Instrument tester (primary actor) (ILT)
IE: Instrument engineer (primary actor) (IST)
HCSS: Herschel common science system (secondary actor)

The generic term USR (for user) is used in the “Main success scenario” and “Extensions” sections below. When reading replace USR by IT or IE as appropriate.

Triggers:

Testing of the instrument is to be performed during ILT [UCF-701].
An instrument specific test is to be performed during IST [UCF-713]
Playback of instrument data is to be performed [UCF-702].

Preconditions:

Test preparation has been performed for ILT tests [UCF-703].
Test preparation has been performed for IST tests [UCF-712].

Minimal postconditions:

None.

Success postconditions:

The EGSE processes are successfully started.

Stakeholders and interests:

TBW.

Main success scenario:

1 USR: Start router (EGSE)
2 USR: Start telemetry gateway and configure for test (EGSE)
3 Telemetry gateway: Register with router (EGSE)
4 USR: Start RTA [UCF-601]
5 USR: Start test control (EGSE) (optional) (see comments)

Extensions:

None.

Frequency of occurence:

Often during ILT and IST

References:

HCSS-UR-3.1-0840 (partial)

Open issues:

- None.

Comments:

- For the purposes of this use case it has been shown that the instrument tester/ instrument engineer starts the processes. However, it is possible that some of the EGSE processes will be started automatically as part of a boot process.
- If telemetry playback is to be performed [UCF-702] then it will not be necessary to start test control. The telemetry will be coming directly from the HCSS archive.

UCF-709 [Summary]: Start the ILT/ IST HCSS

Level: **summary**
Scope: **HCSS**
Version: **2.0**
Status: **issue**

Brief description:

This use case details the steps that will be involved in starting the HCSS processes prior to the testing of an instrument during ILT and IST mission phases or the playback of telemetry from the HCSS archive during these 2 mission phases.

Phase:

Instrument level testing:	Y
Integrated system testing:	Y
Ground segment testing:	N
LEOP:	N
Commissioning:	N
Calibration/ PV:	N
Science demonstration:	N
Routine operations:	N
Post operations:	N

Actors:

IT: Instrument tester (primary actor) (ILT)
IE: Instrument engineer (primary actor) (IST)
EGSE: Electrical ground segment equipment (secondary actor)

The generic term USR (for user) is used in the “Main success scenario” and “Extensions” sections below. When reading replace USR by IT or IE as appropriate.

Triggers:

Testing of the instrument is to be performed during ILT [UCF-701].
An instrument specific test is to be performed during IST [UCF-713]
Playback of instrument data is to be performed [UCF-702].

Preconditions:

Test preparation has been performed for ILT tests [UCF-703].
Test preparation has been performed for IST tests [UCF-712].
The EGSE systems have been started [UCF-708].

Minimal postconditions:

None.

Success postconditions:

The HCSS processes are successfully started.

Stakeholders and interests:

TBW.

Main success scenario:

1 USR: Start telemetry ingestion and configure for test (HCSS) [UCF-758]
2 Telemetry ingestion: Register with router (EGSE)
3 USR: Start QLA in NRT mode (HCSS) [UCF-747]
4 Router: Register QLA (EGSE)
5 USR: Start TC history ingestion process [UCF-759]
6 USR: Start OOL history ingestion process [UCF-763]

Extensions:

1-6a: USR: Start telemetry playback selector [UCF-705] (see comments)
7 USR: Start TC history server on the SCOS-2000 machine [UCF-759] (see comments)
8 USR: Start OOL history server on the SCOS-2000 machine [UCF-763] (see comments)

Frequency of occurence:

Often during ILT and IST

References:

HCSS-UR-3.1-0840 (partial)

Open issues:

- None.

Comments:

- For the purposes of this use case it has been shown that the instrument tester/ instrument engineer starts the processes. However, it is possible that some of the HCSS processes will be started automatically as part of a boot process.
- If telemetry playback is being performed [UCF-702] then it will only be necessary to start the telemetry playback selector. This will be used to send the telemetry from the HCSS archive to the EGSE router.
- Telecommand history and out of limits history servers: In the ILT phase there are servers resident on the SCOS-2000 hardware which interact with SCOS to retrieve the telecommand history and OOL history and pass it to telecommand history and out of limits history ingestion processes on the HCSS hardware. During the IST phase the telecommand history and OOL history files are delivered off-line and ingested into the HCSS.

UCF-711 [User]: Run ILT test procedure

Level: **user**
Scope: **EGSE-ILT**
Version: **2.3**
Status: **issue**

Brief description:

This use case describes how the EGSE-ILT will interact with the HCSS to enable an instrument tester to execute a test procedure.

The HCSS will participate in this use case as a secondary actor. Only those interactions that will involve the HCSS are detailed in this use case. The detail of the interactions of the instrument tester with the EGSE-ILT are not specified.

Phase:

Instrument level testing:	Y
Integrated system testing:	N
Ground segment testing:	N
LEOP:	N
Commissioning:	N
Calibration/ PV:	N
Science demonstration:	N
Routine operations:	N
Post operations:	N

Actors:

IT: Instrument tester (primary actor)
HCSS: Herschel common science system (secondary actor)

Triggers:

Testing of an instrument is to be performed [UCF-701].

Preconditions:

Test preparation has been performed [UCF-703].
EGSE systems have been started [UCF-708].
HCSS systems have been started [UCF-709].

Minimal postconditions:

None.

Success postconditions:

The test procedure is successfully executed.

Stakeholders and interests:

TBW.

Main success scenario:

1 IT: Select instrument model
2 EGSE: Store instrument model

3 IT: Select test procedure from those available
- 3.1 IT: Request list of test procedures
- 3.2 EGSE: Request list of test procedures for this instrument model
- 3.3 HCSS: Supply list of test procedures
- 3.4 IT: Select test procedure
- 3.5 EGSE: Request test procedure
- 3.6 HCSS: Supply test procedure

4 IT: Supply test procedure parameters
5 EGSE: Start test procedure execution
6 HCSS: Store test procedure execution time, test procedure parameters, instrument model
7 EGSE: Start logging (see comments)

IT/ EGSE interaction without HCSS participation to execute the test procedure

8 EGSE: Request observing mode execution to be instantiated and executed (see comments)
- 8.1 EGSE: Supply observing mode mnemonic and associated parameters
- 8.2 HCSS: Instantiate observing mode into observation execution
- 8.3 HCSS: Generate command sequence with absolute timing
- 8.4 EGSE: Fix absolute timing according to current time
- 8.5 EGSE: Report back corrected absolute time for command sequence
- 8.6 HCSS: Fix absolute timing of building block tree for this observation
- 8.7 EGSE: Execute command sequence

Repeat step 8 for all observing modes encountered during test procedure execution

9 EGSE: Detects end of test procedure execution
- 9.1 EGSE: Pass untransmitted log entries to HCSS
- 9.2 HCSS: Accept log records and store them
- 9.3 EGSE: Add log entry with detailed information about end of test procedure
- 9.4 HCSS: Accept log record and store it
- 9.5 EGSE: Report reason for end of test procedure
- 9.6 HCSS: Flag testProcedureExecution with reason for end of test procedure

Extensions:

3a IT: Request test procedure by name
- 3a.1 IT: Specify test procedure by name
- 3a.2 EGSE: Request test procedure
- 3a.3 HCSS: Supply test procedure

3b IT: Request locally saved test procedure
- 3b.1 IT: Specify test procedure name
- 3b.2 EGSE: Load test procedure

6-9a HCSS: Link to EGSE removed and no reconnection within time-out limit
- 6-9a.1 HCSS: Add log record
- 6-9a.1 HCSS: Save log
- 6-9a.1 HCSS: Flag test Procedure Execution

7a EGSE: Abort test
- 7a.1 EGSE: Add log record
- 7a.2 EGSE: Detects end of test procedure execution
  - 7a.2.1 EGSE: Pass untransmitted log entries to HCSS
  - 7a.2.2 HCSS: Accept log records and store them
  - 7a.2.3 EGSE: Add log entry with detailed information about end of test procedure
  - 7a.2.4 HCSS: Accept log record and store it
  - 7a.2.5 EGSE: Report reason for end of test procedure
  - 7a.2.6 HCSS: Flag testProcedureExecution with reason for end of test procedure

8a EGSE/ IT aborts test/ observation
If there are no commands resulting from an observation request pending
it is safe to continue with step 9.
otherwise
Command instrument to ‘reset’ the OBSID and BBID in the TM packets to a default value (see open issues)
flag observation Execution
Continue with step 9
8.2a Generation of observation execution fails
EGSE: Report error back to IT
EGSE: Add log record
IT may choose to continue test procedure or step to 8a
8.6a Fixing times of building block tree fails
EGSE: Report error back to IT
EGSE: Add log record
IT may choose to continue test procedure or step to 8a
Building block data inside the databases of the HCSS has to be fixed off-line

Frequency of occurence:
Often during ILT and IST

References:
HCSS-UR-3.1-0840 (partial)
HCSS-UR-3.1-1290
HCSS-UR-3.1-1340
FGS-IR-4.1-10
FGS-IR-4.1-20
FGS-IR-4.1-30

Open issues:
- An extension use case is needed to describe the case where the EGSE-ILT does not follow the usual sequence of events; for example, EGSE-ILT dies halfway through a test procedure and does not generate a log [**AD-4**: Open issue 2.12.2].
- The current scenario does not allow nested procedure templates. Test procedure templates may use nested constructs as the scripting language allows, but the current model requires each procedure to be a self-contained script [**AD-4**: Open issue 2.12.2].
- There must be a way to terminate an ongoing observation in a clean way without starting a new one. This is needed for ILT and IST but also for operations, e.g. before a time window starts which is reserved for spacecraft operations. One approach can be a defined ‘idle’ observation ID, which is set whenever the instrument is not executing an observation.
- HCSS must store which test procedure and which version of it was used.

Comments:
- Note: Test control can use test procedure templates retrieved from the HCSS for subsequent execution by SCOS 2000. It also permits the sending of individual telecommand mnemonics to SCOS 2000. This use case relates specifically to the use of the test control interface between test control and the HCSS.
- The interfacing between the EGSE and the HCSS described in this use case is formally defined in the “HCSS - Test Control ICD.

- The observing mode to command sequence loop should take no longer than n seconds (Albrecht - 5 seconds). Add to IRD. The reaction to events during test procedure execution will not involve the HCSS.
- The log is made persistent as a BLOB - it is not ingested/ processed.
- Instantiate observing mode into observation execution: This will include building blocks.
- Selecting the instrument model shall not be repeated for every ILT test procedure execution. The selection shall be saved, for example in the user environment.
- The HCSS will not read the test procedure script but it will store certain attributes of test procedure like for which instrument models the test procedure is valid.
- A test procedure (execution) is generated by supplying the required input parameters to a selected test procedure template during the run test procedure use case.
- An observation request is an observing mode with parameter values specified.
- Log records are passed immediately to the HCSS for storage.

UCF-758 [User]: Ingest near real-time telemetry during ILT/ IST

Level: **user**
Scope: **HCSS**
Version: **3.6**
Status: **issue**

Brief description:

This use case describes the steps that will be performed during the ingestion of unconsolidated near real-time telemetry into the HCSS during the ILT and IST mission phases.

Phase:

Instrument level testing:	Y
Integrated system testing:	Y
Ground segment testing:	N
LEOP:	N
Commissioning:	N
Calibration/ PV:	N
Science demonstration:	N
Routine operations:	N
Post operations:	N

Actors:

IT: Instrument tester (primary actor) [ILT]
IE: Instrument engineer (primary actor) [IST]

The generic term USR (for user) is used in the “Main success scenario” and “Extensions” sections below. When reading replace USR by IT or IE as appropriate.

Triggers:

Testing of an instrument is to be performed [UCF-701].
Integrated system testing is to be performed [UCF-713]

Preconditions:

The EGSE router is running [UCF-708].

Minimal postconditions:

No telemetry should be unintentionally lost.

Success postconditions:

The telemetry packets and the generated data frames are stored within the HCSS.
The data frames, where applicable, are correctly linked to the observations and block execution with which they are associated.

Stakeholders and interests:

TBW

Main success scenario:

1 USR: Start telemetry ingestion (see comment)

1.1 USR: Configure telemetry ingestion
1.1.1 USR: Identify mission configuration associated with the telemetry to be processed
1.1.2 USR: Identify data frame generators to be used (see comments)
1.1.3 USR: Identify APIDS of the telemetry packets to be stored
1.1.4 USR: Identify machine and port of the EGSE router
1.1.5 USR: Identify database in which to store telemetry packets and data frames
1.1.6 USR: Identify no of objects before database commit
1.1.7 USR: Identify duration before timer commit
1.2 USR: Invoke telemetry ingestion process
2 HCSS: Start telemetry ingestion process
2.1 HCSS: Open new telemetry ingestion log
2.2 HCSS: Establish connection with identified database
2.3 HCSS: Verify inst. model exists in database for requested data frame generators
2.4 HCSS: Invoke requested data frame generator
2.5 HCSS: Establish connection with EGSE router
2.6 HCSS: Inform EGSE router of required telemetry packets (by APID)
2.7 HCSS: Start commit timer running
3 HCSS: Receive telemetry packet from EGSE router
3.1 HCSS: Receive telemetry packet as a byte stream from EGSE router
3.2 HCSS: Determine if telemetry packet is from instrument, spacecraft or EGSE
3.3 HCSS: Create tm packet from byte stream and associate with inst, s/c or EGSE
4 HCSS: Process received telemetry packet
4.1 HCSS: Validate telemetry packet
4.1.1 HCSS: Check that telemetry packet time is synchronized (see comments)
4.1.2 HCSS: Check that telemetry packet is next in sequence
4.2 HCSS: Generate data frame from science telemetry packets
4.3 HCSS: Associate data frame with building block
4.4 HCSS: Associate telemetry packet with building block
4.5 HCSS: Commit tm/df to database when identified number of objects processed
4.6 HCSS: Commit telemetry packet / data frame to database when triggered by timer
Repeat steps 3 and 4 for all telemetry packets

Extensions:

1-4a USR: Request to stop telemetry ingestion
1-4b HCSS: Commit objects to the database
1-4c HCSS: Stop telemetry ingestion
2.2a HCSS: Unable to establish connection with database
2.2a1 HCSS: Log error message and terminate
2.3a HCSS: Instrument model does not exist in the database
2.3a1 HCSS: Log error message and terminate
2.4a HCSS: Unable to invoke data frame generator
2.4a1 HCSS: Log error message and terminate
2.5a HCSS: Unable to establish connection with EGSE router
2.5a1 HCSS: Log error message
2.5a2 HCSS: Wait pre-configure amount of time, re-establish contact
Repeat step 2.4a until connection is re-established
3.3a HCSS: Error detected when creating telemetry packet
3.3a1 HCSS: Log error message and continue
4a HCSS: Connection to router fails
4a1 HCSS: Log error message

4a2 HCSS: Wait pre-configure amount of time, re-establish contact
Repeat step 4a until connection is re-established
4.1.1a HCSS: Telemetry packet time is not synchronized (see comments)
HCSS: Log the identification of unsynchronized telemetry packet and continue
4.1.2a HCSS: Telemetry packet is not next in sequence
HCSS: Log the identification of missing telemetry packets and continue
4.2a HCSS: Failure in data frame generation process
HCSS: Log failure and continue
4.3a HCSS: No building block associated with building block id of data frame
HCSS: Log message and continue
4.4a HCSS: Error committing objects to the database (see comments)
HCSS: Log error message and terminate
4.5a HCSS: Error committing objects to the database (see comments)
HCSS: Log error message and terminate
4.6a HCSS: Unrecognized telemetry packet (see comments)
HCSS: Log error message

Frequency of occurence:

ILT: Repeated use, several times per day.
IST: Whenever an instrument specific IST test is being performed.

References:

- HCSS Telemetry Ingestor Technical Note, FSCDT/TN-015
- Herschel science ground segment to instruments ICD, FIRST-FSC-DOC-0200, Issue 1.0, 13 July 2001
  - Section 2.1.1 OBSID and BBID FIELDS DEFINITION
- FIRST ground segment interface requirements document, FIRST/FSC/DOC/0117, Issue 1.3, 03 November 2000:
  - Section 3.1.1 CONSOLIDATED TM
  - Section 4.2.1 TELEMETRY
- FIRST ground segment design document, FIRST/FSC/DOC/0146, Issue 1.3, 03 November 2000:
  - Section 3.1.8.2 The TM I/F towards FINDAS
  - Section 3.2.7.3 TM I/F
  - Section 3.3.7.3 TM I/F
- ILT working group technical note “Analysis of ILT use-cases”, version 0.5,
  - Section 6.1.1 Comments
- Packet structure ICD, SCI-PT-ICD-07527, issue 5.0

HCSS-UR-3.1-0840 (partial)
FGS-IR-4.2-10
FGS-IR-4.2-20
FGS-IR-4.2-21
FGS-IR-4.2-25
FGS-IR-4.2-30
FGS-IR-4.2-40
FGS-IR-4.2-50

Open issues:

- Which instruments generate telemetry before time synchronization has been performed [**AD-4:** Open issue 2.9.6].

Comments:

- The instrument science telemetry contains observation and building block identifiers [**AD-4:** Open issue 2.9.2].
- For PACS, due to the compression of the science data, it is not possible to associate all science telemetry packets to a single building block. These packets will be associated with the observation [**AD-4:** Open issue 2.12.9].
- A log will be generated of all telemetry ingestion activities and detected anomalies [**AD-4:** Open issue 2.12.9].
- Telemetry occurring outside of an observation context will be assigned default observation and building block identifiers [**AD-4:** Open issue 2.9.3 and 2.9.4].
- The source sequence count of every telemetry packet will be checked to determine if there are missing telemetry packets [**AD-4:** Open issue 2.12.9].
- A telemetry packet is unsynchronized if the MSB of the time field is 1 [**AD-4:** Open issue 2.9.5].
- ILT: TM ingestion supports the scenario where tests on different models run in parallel. The instrument whose telemetry is to be processed is identified to the telemetry ingestion process at start-up. The telemetry ingestion process/ HCSS must be able to distinguish between different instrument models. For example, one test could be performed using the instrument HIFI-1. Another test could be performed using a different instrument, HIFI-2 say. This is **NOT** to be confused with parallel operation. The telemetry ingestion process will not have to handle two instruments operating in parallel [**AD-4:** Open issue 2.12.9].
- ILT: EGSE-APIDs may not be unique. A number of APIDs is reserved for EGSE purposes. This number is so small that the different instruments will probably use the same APID for different purposes. At the time of ingestion the telemetry ingestion process knows the instrument by which the telemetry was generated. This causes no problems [**AD-4:** Open issue 2.12.9].
- ILT: The HIFI and SPIRE test equipment interface (TEI) generated telemetry will be labelled with observation and building block identifiers. PACS is TBC.
- There are the following types of telemetry packet which will not be associated with observation and/ or building block: Housekeeping telemetry which is generated continuously while the instrument is active and telemetry which is generated as a result of manual commanding.
- For the purposes of this use case it has been shown that the instrument tester/ instrument engineer starts the telemetry ingestion process. However, it is possible that the process will be configured once and then subsequently started automatically as part of a boot process.
- Non-synchronized telemetry is not produced during ILT phase. TBD for IST phase.
- The telemetry ingestion process must try to continue under all circumstances. For example: (1) If the connection to the router is lost it must try to re-establish the connection. (2) If an error is detected when creating a telemetry packet the error is logged and the telemetry ingestion process moves on to the creation of the next telemetry packet.
- Identify data frame generators: For each instrument the USR actor will indicate whether data frames are to be generated and if so, which data frame generator to use.
- Error committing objects to the database: If a database error is detected it will not be possible to recover automatically. The telemetry ingestion process terminates.
- Unrecognized telemetry packet: The APIDs to be processured by the telemetry ingestion process are identified in the packet structure ICD (see references section above). If an

invalid APID is detected an error message is logged. Note it is note possible to create the telemetry packet as the telemetry ingestion process has no way of knowing to which group the packet belongs (instrument, spacecraft, egse equipment etc.)

UCF-601 [User]: Perform RTA

Level: **user**
Scope: **HCSS and RTA**
Version: **2.1**
Status: **issue**

Brief description:

This use case describes the steps that will be involved in the running of RTA and its interactions with the HCSS.

Phase:

Instrument level testing:	Y
Integrated system testing:	Y
Ground segment testing:	Y
LEOP:	N
Commissioning:	Y
Calibration/ PV:	Y
Science demonstration:	Y
Routine operations:	Y
Post operations:	N

Actors:

IT: Instrument tester (primary actor)
IE: Instrument engineer (primary actor)
CS: Calibration scientist (primary actor)
HCSS: Herschel common science system (secondary actor)

The generic term USR (for user) is used in the "Main success scenario" and "Extensions" sections below. When reading replace USR by IT, IE or CS as appropriate.

Triggers:

- An instrument test is to be performed during ILT mission phase [UCF-701].
- An instrument test is to be performed during IST mission phase [UCF-713].
- Instrument housekeeping data needs to be re-analysed to investigate (anomalous) instrument behaviour [UCF-702].

Preconditions:

- ILT: The relevant MIB has been made available to RTA [UCF-703] [UCF-762]
- IST: The relevant spacecraft database has been made available to RTA [UCF-712][UCF-763]
- The telemetry gateway has been started and configured [UCF-708]

Minimal postconditions:

None.

Success postconditions:

RTA processes the telemetry packet data.
The RTA process products are made available to the HCSS.

Stakeholders and interests:

ICC: Wants to use RTA as one of many means to analyse instrument behaviour

Main success scenario:

1 USR: Start RTA client
2 USR: Configure RTA client
3 RTA: Establish link with telemetry gateway
Telemetry packet data stream arrives via telemetry gateway
4 RTA: Receive and analyse telemetry packet data stream, generate products
5 RTA: Make logs available to HCSS ingestion processes

Extensions:

3a RTA: Unable to establish link with telemetry gateway

Frequency of occurence:

Several (many?) times per day during ILT and IST
A few times per day during early operations
A few times per week during later operations

References:

HCSS-UR-3.1-0840 (partial)
HCSS-UR-3.1-0990
HCSS-UR-3.1-0991
HCSS-UR-3.1-0992
HCSS-UR-3.1-0993
HCSS-UR-3.1-0994
FGS-IR-3.8-10
FGS-IR-3.8-20
FGS-IR-3.8-30
FGS-IR-4.3-10
FGS-IR-4.3-20
FGS-IR-4.3-25
FGS-IR-4.3-26
FGS-IR-4.3-30
FGS-IR-4.3-35
FGS-IR-4.3-40 (Not compliant)
FGS-IR-4.3-50 (Not compliant)
FGS-IR-4.3-60 (Not compliant)
FGS-IR-4.4-10 (Not compliant)
FGS-IR-4.4-20 (Not compliant)
FGS-IR-4.4-30 (Not compliant)
FGS-IR-4.4-40
FGS-IR-4.4-50

Open issues:

- None.

Comments:

- The telemetry gateway specifies which telemetry packets are wanted (from the telemetry router). It forwards these packets to the RTA system (see UCF-701, UCF-702, UCF-713).

- If RTA is to be used for analysis of near real-time (unconsolidated) telemetry outside of the context of MOC, the link between MOC and ICC is not provided by HCSS.
- If RTA is running at MOC and it is required to ingest RTA results (logs, events, etc.) into HCSS, a link has to be provided.
- Is there a need for an 'automatic' running of RTA (e.g. in a quality control process at the ICCs)? Running RTA automatically can be covered by proper procedures surrounding the current use case. Nothing in the use case makes it special.
- The interface between the HCSS and RTA is described in a set of ICDs (ICD-7, ICD-8 and ICD-12) [**AD-4**: Open issue 2.7.3].
- The products currently identified for ingestion into the HCSS [UCF-701] are the telecommand history and the out of limits history.
- RTA logs are OOL data and they will be associated with observation objects [**AD-4**: Open issue 2.10.9].
- If RTA logs for a particular observation already exist the user gets the choice to insert/overwrite this if a new logs is generated (e.g. from a new MIB) [**AD-4**: Open issue 2.10.9].
- When inserting the OOL data into the HCSS the version of the MIB used to generate this data must be identified [**AD-4**: Open issue 2.10.9].

UCF-747 [User]: Perform QLA

Level: **user**
Scope: **HCSS**
Version: **2.2**
Status: **issue**

Brief description:

This use case describes the running of QLA in either playback mode or NRT mode and its interactions with the HCSS archive. It does not identify the specific details of QLA processing.

A distinction is made below between HCSS QLA and HCSS access. HCSS access is an API used by QLA in order to access the data to be analysed.

Phase:

Instrument level testing:	Y
Integrated system testing:	Y
Ground segment testing:	Y
LEOP:	N
Commissioning:	Y
Calibration/ PV:	Y
Science demonstration:	Y
Routine operations:	Y
Post operations:	N

Actors:

IT: Instrument Tester (primary actor)
IE: Instrument Engineer (primary actor)
CS: Calibration Scientist (primary actor)

The generic term USR (for user) is used in the “Main success scenario” and “Extensions” sections below. When reading replace USR by IT, IE or CS as appropriate.

Triggers:

ILT: Instrument testing is being performed [UCF-701].
IST: ITS test is being performed [UCF-713]
ILT/IST: Off-line analysis is to be performed [UCF-706].

IST to routine operations: Instrument science data needs to be re-analysed to investigate (anomalous) instrument behaviour.

Preconditions:

ILT: Test preparation has been performed for ILT testing [UCF-703].
IST: Test preparation has been performed for IST testing [UCF-712].
ILT: The required playback test data is in the archive [UCF-701].
IST: The required playback test data is in the archive [UCF-710].

Minimal postconditions:

None.

Success postconditions:

QLA processes the data.
QLA process products, if any, are made persistent in HCSS (TBC).

Stakeholders and interests:

ICC: Want to use QLA as one of many means to analyse instrument behaviour

Main success scenario:

1 USR: Start QLA
2 USR: Configure QLA for playback
  2.1 USR: Select playback mode
  2.2 USR: Select database from which data is to be retrieved
  2.3 USR: Select telemetry packets or data frame data
  2.4 USR: Specify data selection criteria (see comments)
  2.5 USR: Select data playback rate
3 USR: Request connection
4 HCSS access: Establish connection with identified data source (see comments)
5 HCSS access: Get data from database and pass to QLA
6 USR: Interact with QLA to analyse data
7 HCSS QLA: Perform requested analysis steps
Steps 5 to 7 can be performed in parallel
Repeat steps 2 to 7 until analysis complete

Extensions:

2a: USR: Configure QLA for NRT connection
  2a1: USR: Select NRT mode
  2a2: USR: Select telemetry packets or data frames
6a: USR: Request automatic analysis
7a: HCSS QLA: Perform automatic analysis on the data
8: HCSS QLA: Automatically generate QLA report and store in archive (TBC)
8a: USR: Create QLA report and request storage in the archive (TBC).

Frequency of occurence:

For ILT the processing is performed at least once for every observation execution
For IST the processing is performed at least once for every observation execution
A few times per day during early operations
A few times per week during later operations

References:

HCSS-UR-3.1-0840 (partial)
HCSS-UR-3.1-0990
HCSS-UR-3.1-0991
HCSS-UR-3.1-0992
HCSS-UR-3.1-0993
HCSS-UR-3.1-0994

Open issues:

- None.

Comments:

- In playback mode it will be possible to request the retrieval of data based on:

    - Observation identifier
    - Time period
- NRT connection to data source:
    - During the ILT mission phase this will be the EGSE-ILT router.
    - During the IST mission phase this will be the I-EGSE router.
    - TBC for the subsequent mission phases.
- Playback connection to data source:
    - For all mission phases this will be the identified database.
- Details of any QLA reporting and its subsequent storage in the archive are still TBC. PACS currently state that QLA will produce a printed summary report.

UCF-759 [User]: Ingest telecommand history during ILT phase

Level: **user**
Scope: **HCSS**
Version: **2.5**
Status: **issue**

Brief description:

This use case describes the steps that will be performed to ingest the telecommand history into the HCSS during the ILT phase.

Phase:

Instrument level testing:	Y
Integrated system testing:	N
Ground segment testing:	N
LEOP:	N
Commissioning:	N
Calibration/ PV:	N
Science demonstration:	N
Routine operations:	N
Post operations:	N

Actors:

IT: Instrument tester (primary)

Triggers:

ILT: Instrument tests are being performed [UCF-701].

Preconditions:

None.

Minimal postconditions:

The previously ingested TC history is unaffected.

Success postconditions:

The TC history is ingested

Stakeholders and interests:

TBW

Main success scenario:

1 IT: Start telecommand history file server on SCOS-2000 machine
  1.1 IT: Configure telecommand history file server
  1.1.1 IT: Identify server and port of the machine on which SCOS-2000 is running
  1.1.2 IT: Identify directory to which the history file should be written
  1.1.3 IT: Identify SCOS-2000 time zone
  1.2 IT: Invoke telecommand history file server
2 IT: Start telecommand history ingestion process on HCSS machine
  2.1 IT: Configure telecommand history ingestion process
  2.1.1 IT: Identify database in which to store telecommand history file records

2.1.2 IT: Identify server and port of the machine on which SCOS-2000 is running
2.1.3 IT: Identify wait time before attempting to re-establish lost contact with THF server
2.1.4 IT: Identify maximum number of attempts to re-establish contact with THF server
2.1.5 IT: Identify start time from which retrieval will start (see comments)
2.1.6 IT: Identify wait time period before retrieval (see comments)
2.1.7 IT: Identify time tolerance (see comments)
2.2 IT: Invoke telecommand history ingestion process
3 HCSS: Establish connection with identified database
4 HCSS: Establish connection with THF server
5 THFServer: Establish connection with HCSS
6 HCSS: Request telecommand history for current time period (see comments)
7 THFServer: Receive request from HCSS for specified time period
8 THFServer: Establish connection with SCOS-2000
9 THFServer: Request telecommand history data for the specified time period
10 THFServer: Receive name of the created telecommand history data file from SCOS-2000
11 THFServer: Retrieve telecommand history data file
12 THFServer: Send telecommand history data file to HCSS
13 HCSS: Receive telecommand history data file from THFServer
14 HCSS: Validate telecommand history data file
14.1 HCSS: Check first line of file (see comments)
14.2 HCSS: Check that time window covered by the file is as requested
15 HCSS: Using telecommand id associate TCH record with corresponding TC in database
16 HCSS: Update success flags of observations and block executions covered by time period
Repeat steps 15 and 16 for each telecommand history record in the file.
17 HCSS: Validate that all telecommands for the specified time period have a TCH record
18 HCSS: Commit the changes to the database
Repeat steps 6-17 after the specified wait time period

Extensions:

2.2a HCSS: Invalid format for retrieval start time
2.2a1 HCSS: Log error message and terminate
3a HCSS: Connection to database fails
3a1 HCSS: Log error message and terminate
4-16a HCSS: Established connection to THFServer drops
4-16a1 HCSS: Log error message and attempt reconnect
4-16b HCSS: Connection to THFServer fails after specified number of reconnection attempts
4-16b1 HCSS: Log error message and terminate
4a HCSS: Requested server/ port for THFServer not recognized
4a.1 HCSS: Log error message and attempt reconnect
4a.1a HCSS: Connection to THFServer fails after specified number of reconnection attempts
4a.1a.1 HCSS: Log error message and terminate
8a THFServer: Requested server/ port for SCOS-2000 not recognized
8a.1 HCSS: Log error message and terminate
14.1a HCSS: Invalid first line of file detected
14.1a1 HCSS: Log error message and terminate
14.2a HCSS: Invalid time window detected
14.2a1 HCSS: Log error message and wait for next retrieval request to be triggered
15a HCSS: Telecommand history record is from a manual command
15a1 HCSS: Store telecommand history record in database without association (TBC)
15b HCSS: Unable to locate telecommand corresponding to telecommand identifier

15b1 HCSS: Log error message
15c HCSS: Telecommand history record already exists in database (see comments)
15c1 HCSS: Log error message
17a HCSS: Telecommand detected without corresponding TCH record
17a1 HCSS: Log error message

Frequency of occurence:
TBW

References:
TC History ICD, M.J.Graham, SPIRE-ICS-DOC-000900, 1.0, 30 October 2002

HCSS-UR-3.1-0840 (partial)

Open issues:
- None.

Comments:
- TC history ingestion is a continuously running processes which will checks for updates to the TC history at regular intervals.
- Requesting the telecommand history for current time period: The first request for telecommand history records is from the “start time” to the current time. Subsequent requests are made after the “wait time period”.
- The telecommands will be associated with the relevant observation execution (assuming all ICCs use the SCOS v2.2e evolution line [**AD4**: Open issue 2.12.10]. Note: v2.2e was cancelled and v2.3 is now applicable.
- Validation: The first record of the retrieved file must conform to the specification in TC History ICD (see references).
- Record already associated with telecommand in database: The new record is **not** associated with the telecommand

UCF-763 [User]: Ingest OOL history during ILT phase

Level: **user**
Scope: **HCSS**
Version: **2.4**
Status: **issue**

Brief description:

This use case describes the steps that will be performed to ingest the out of limits (OOL) history into the HCSS during the ILT mission phase.

Phase:

Instrument level testing:	Y
Integrated system testing:	N
Ground segment testing:	N
LEOP:	N
Commissioning:	N
Calibration/ PV:	N
Science demonstration:	N
Routine operations:	N
Post operations:	N

Actors:

IT: Instrument tester (primary)

Triggers:

ILT: Instrument tests are being performed [UCF-701].

Preconditions:

None.

Minimal postconditions:

The previously ingested OOL history is unaffected.

Success postconditions:

The OOL history is ingested

Stakeholders and interests:

TBW

Main success scenario:

1 IT: Start OOL file server on SCOS-2000 machine
  1.1 IT: Configure OOL file server
  1.1.1 IT: Identify server and port of the machine on which SCOS-2000 is running
  1.1.2 IT: Identify directory to which the history file should be written
  1.1.3 IT: Identify SCOS-2000 time zone
  1.2 IT: Invoke OOL file server
2 IT: Start OOL history ingestion process on HCSS machine
  2.1 IT: Configure OOL history ingestion process
  2.1.1 IT: Identify database in which to store OOL history file records

2.1.2 IT: Identify server and port of the machine on which SCOS-2000 is running
2.1.3 IT: Identify wait time before attempting to re-establish lost contact with OOL server
2.1.4 IT: Identify maximum number of attempts to re-establish contact with OOL server
2.1.5 IT: Identify start time from which retrieval will start (see comments)
2.1.6 IT: Identify wait time period before retrieval (see comments)
2.1.7 IT: Identify time tolerance (see comments)
2.2 IT: Invoke OOL history ingestion process
3 HCSS: Establish connection with identified database
4 HCSS: Establish connection with OOL server
5 OOLServer: Establish connection with HCSS
6 HCSS: Request OOL history for current time period (see comments)
7 OOLServer: Receive request from HCSS for specified time period
8 OOLServer: Establish connection with SCOS-2000
9 OOLServer: Request OOL history data for the specified time period from SCOS-2000
10 OOLServer: Receive name of the created OOL history data file from SCOS-2000
11 OOLServer: Retrieve OOL history data file
12 OOLServer: Send OOL history data file to HCSS
13 HCSS: Receive OOL history data file from THFServer
14 HCSS: Validate OOL history data file
14.1 HCSS: Check that time window covered by the file is as requested
15 HCSS: Associate OOL history record with associated building execution (see comments)
Repeat steps 15 for each OOL history record in the file.
16 HCSS: Commit the changes to the database
Repeat steps 6-16 after the specified wait time period

Extensions:

2.2a HCSS: Invalid format for retrieval start time
2.2a1 HCSS: Log error message and terminate
3a HCSS: Connection to database fails
3a1 HCSS: Log error message and terminate
4-16a HCSS: Connection to OOLServer fails
4-16a1 HCSS: Log error message and attempt to reconnect
4-16b HCSS: Connection to OOLServer fails after specified number of reconnection attempts
4-16b1 HCSS: Log error message and terminate
4a HCSS: Requested server/ port for OOLServer not recognized
4a.1 HCSS: Log error message and attempt to reconnect
4a.1a HCSS: Connection to OOLServer fails after specified number of reconnection attempts
4a.1a.1 HCSS: Log error message and terminate
8a OOLServer: Requested server/ port for SCOS-2000 not recognized
8a.1 HCSS: Log error message and terminate
14.1a HCSS: Invalid time window detected
14.1a1 HCSS: Log error message and wait for next retrieval request to be triggered
15a HCSS: Unable to locate building block with which to associate OOL history record
15a1 HCSS: Log error message
15b HCSS: OOL history record already exists in database (see comments)
15b1 HCSS: Log error message

Frequency of occurence:

TBW

References:

HCSS-UR-3.1-0840 (partial)

Open issues:

- None.

Comments:

- Out of limits history ingestion is a continuously running processes which will checks for updates to the TC history at regular intervals.
- The OOL history records are associated with block execution records based on time. The block execution most closely matching the OOL record in time is the one with which the OOL record will be associated.
- The check should detect and flag any inconsistencies (for example, not possible to determine observation execution associated with OOLs).
- Requesting the OOL history for current time period: The first request for OOL history records is from the “start time” to the current time. Subsequent requests are made after the “wait time period”.
- Record already associated with building block in database: The new record is **not** associated with the building block.

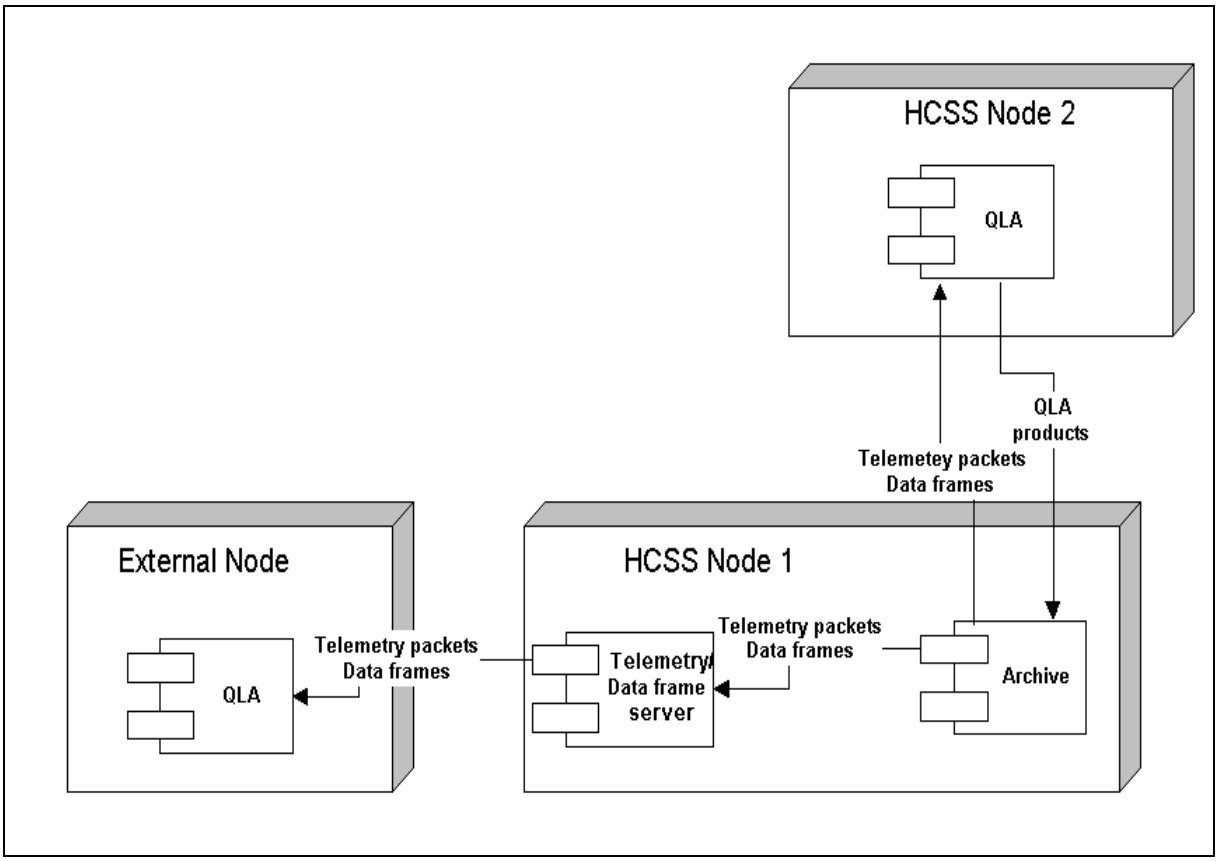
5 ILT off-line analysis

5.1 Overview

This section presents those use cases which detail the steps that will be involved in performing off-line analysis.

Two types of QLA are shown in Figure 3 and described in UCF-706. HCSS internal QLA and external QLA. A server within the HCSS is required to retrieve the required telemetry/ data frames from the HCSS archive and forward them to the external QLA. Internal QLA will be able to access the HCSS archive directly.

Figure 3: Off-line analysis process diagram



QLA is performed in both NRT analysis (Section 4) and off-line analysis. The relevant use case (UCF-747) is contained in Section 4.

UCF-706 [Summary]: ILT/ IST off-line analysis

Level: **summary**
Scope: **EGSE-ILT and HCSS**
Version: **2.0**
Status: **issue**

Brief description:

This use case describes the steps that will be performed during off-line analysis.

Phase:

Instrument level testing:	Y
Integrated system testing:	Y
Ground segment testing:	N
LEOP:	N
Commissioning:	N
Calibration/ PV:	N
Science demonstration:	N
Routine operations:	N
Post operations:	N

Actors:

IT: Instrument tester (primary actor) [ILT]
IE: Instrument engineer (primary actor) [IST]

The generic term USR (for user) is used in the “Main success scenario” and “Extensions” sections below. When reading replace USR by IT or IE as appropriate.

Triggers:

ILT: It is required to perform off-line analysis of instrument test data [UCF-700].
IST: It is required to perform off-line analysis of instrument specific IST test data [UCF-710].

Preconditions:

The required test data is in the archive [UCF-701] [UCF-713].

Minimal postconditions:

None.

Success postconditions:

The instrument test data is successfully analysed.

Stakeholders and interests:

TBW.

Main success scenario:

1 USR: Start telemetry and data frame server (optional)
2 USR: Run QLA in playback mode [UCF-747]
3 USR: Run interactive analysis

Extensions:

None.

Frequency of occurence:

Often during ILT and IST

References:

TBD

Open issues:

- None.

Comments:

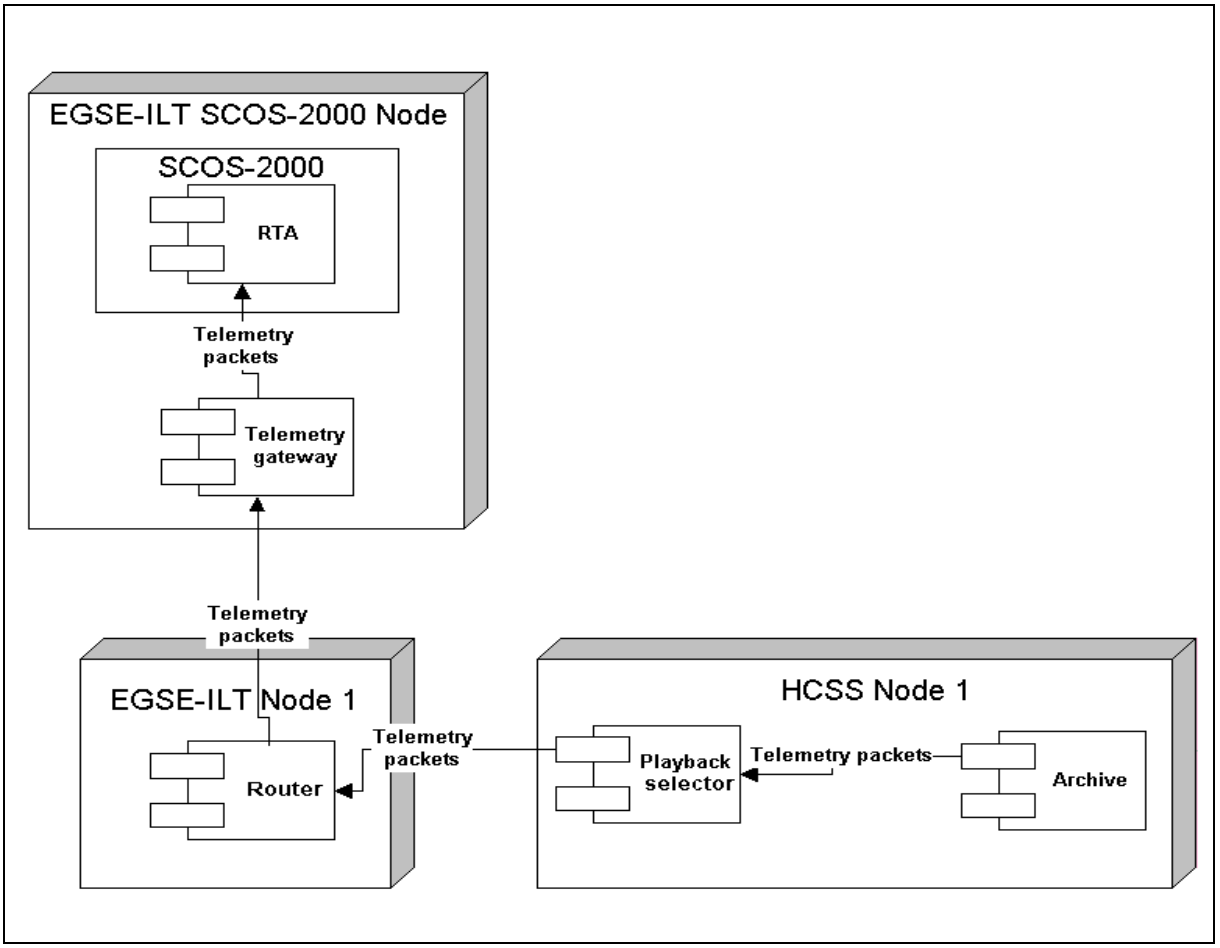
- None.

6 ILT test telemetry replay

6.1 Overview

It will be necessary to playback the instrument data in order to further analyse the performance of the instrument.This section presents those use cases which detail the steps that will be involved in performing test telemetry replay. Figure 4 shows the processes involved in performing this instrument data playback.

Figure 4: Playback instrument data process diagram



RTA is used to analyse the housekeeping telemetry and is an external system to the HCSS. The summary level use case UCF-702 describes the high level steps involved in playing back telemetry data to RTA.

As in the near real time RTA operations during the performance of an instrument test (see Section 4) RTA will receive its telemetry via the telemetry gateway and therefore the sequence of events will be much the same in the 2 scenarios. The only major difference will be that the telemetry packets will no longer be generated by the instrument and forwarded to the EGSE-ILT router but will be retrieved from the HCSS archive by the telemetry playback selector and forwarded to the EGSE-ILT router.

Due to the above mentioned commonality several of the use cases identified in UCF-702 are contained in Section 4. They are:

- UCF-708: Start the EGSE
- UCF-709: Start the HCSS
- UCF-601: Perform RTA

UCF-702 [Summary]: Replay ILT/ IST test telemetry

Level: **summary**
Scope: **EGSE-ILT and HCSS**
Version: **3.0**
Status: **issue**

Brief description:

This summary level use case describes the steps that will be performed for instrument telemetry playback (replay) during the ILT and IST mission phases.

Phase:

Instrument level testing:	Y
Integrated system testing:	Y
Ground segment testing:	N
LEOP:	N
Commissioning:	N
Calibration/ PV:	N
Science demonstration:	N
Routine operations:	N
Post operations:	N

Actors:

IT: Instrument tester (primary actor) [ILT]
IE: Instrument engineer (primary actor) [IST]
EGSE-ILT: The electrical ground segment equipment system for ILT.
Instrument-EGSE: The electrical ground segment equipment system used during IST,

The generic term USR (for user) is used in the “Main success scenario” and “Extensions” sections below. When reading replace USR by IT or IE as appropriate.

The generic term EGSE is used in the “Main success scenario” and “Extensions” sections below. When reading replace EGSE by EGSE-ILT or Instrument-EGSE as appropriate.

Triggers:

It is necessary to replay ILT telemetry [UCF-700].
It is necessary to replay IST telemetry [UCF-710].

Preconditions:

Test preparation has been performed [UCF-703] [UCF-712].

Minimal postconditions:

None.

Success postconditions:

The telemetry playback is successful.

Stakeholders and interests:

TBW.

Main success scenario:

1 USR: Start EGSE [UCF-708]
2 USR: Start HCSS [UCF-709]
3 USR: Start playback selector and identify playback telemetry (HCSS) [UCF-705]
4 Router: Register playback selector (EGSE)
5 Playback selector: Retrieve TM from database and forward to router (HCSS) [UCF-704]
6 Router: Receive telemetry packet and forward to registered clients (EGSE)
7 RTA: Receive telemetry and process it (EGSE) [UCF-601]
Repeat steps 5 to 7 until telemetry playback is complete

Extensions:

None.

Frequency of occurence:

Often during ILT

References:

TBD

Open issues:

- None.

Comments:

- None.

UCF-705 [User]: Run telemetry playback selector

Level: **user**
Scope: **HCSS**
Version: **2.0**
Status: **issue**

Brief description:

This use case describes the steps that will be performed when running the telemetry playback selector.

Phase:

Instrument level testing:	Y
Integrated system testing:	Y
Ground segment testing:	Y
LEOP:	N
Commissioning:	Y
Calibration/ PV:	Y
Science demonstration:	Y
Routine operations:	Y
Post operations:	N

Actors:

IT: Instrument tester (primary actor)
IE: Instrument engineer (primary actor)
CS: Calibration scientist (primary actor)
EGSE-ILT: Electrical ground segment equipment system for ILT (secondary actor)

The generic term USR (for user) is used in the “Main success scenario” and “Extensions” sections below. When reading replace USR by IT, IE or CS as appropriate.

Triggers:

- Telemetry playback is to be performed [UCF-702].

Preconditions:

- For ILT: The EGSE-ILT router has been started [UCF-702]
- TBD for IST onwards.

Minimal postconditions:

None.

Success postconditions:

- ILT: The required telemetry packets are retrieved and forwarded to the EGSE-ILT router.
- IST onwards: The required telemetry packets are retrieved and forwarded to the TBD.

Stakeholders and interests:

TBW

Main success scenario:

1 USR: Start the telemetry playback selector

2 USR: Configure telemetry playback selector
  2.1 USR: Identify database from which telemetry is to be retrieved
  2.2 USR: Identify location of EGSE-ILT router
  2.3 USR: Identify telemetry packets to be retrieved
3 HCSS: Establish connection with identified database
4 HCSS: Establish connection with EGSE-ILT router
5 HCSS: Retrieve telemetry packet from database and forward to EGSE-ILT router
Repeat step 5 until all telemetry packets have been retrieved.

Extensions:
TBW

Frequency of occurence:
Several (many?) times per day during ILT and check-out
A few times per day during early operations
A few times per week during later operations

References:
TBW

Open issues:
- None.

Comments:
- The HCSS will support the retrieval of all types of telemetry packets.
- The telemetry playback selector user will be able to select the types of telemetry packet to be retrieved.
- The telemetry playback selector user will be able to request telemetry packets over a given absolute time or a given time range relative to the current time.
- The telemetry playback selector user will be able to request the most recent telemetry packets.
- For the purposes of this use case it has been shown that the actor starts the processes. However, it is possible that the process will be started automatically as part of a boot process. The actor can then configure and reconfigure the process as necessary