ABSTRACT

This document provides the LIME Toolbox verification plan, i.e. testing of the developed software, for the LIME-2 project.

Ramiro González
Carlos Toledano
Javier Gatón
Pieter De Vis
Jacob Fahy
Stefan Adriaensen

UVa , NPL, VITO

27 January 2022

LIME TOOLBOX

VERIFICATION PLAN

LUNAR IRRADIANCE MODEL OF ESA

This document was produced as part of the ESA-funded project "Improving the Lunar Irradiance Model of ESA" under ESA contract number:

esa NPL National Physical Laboratory vito UVa

Signatures and version history

	Name	Organisation	Date
Written by	Carlos Toledano	UVa	12 January 2022
Reviewed by (consortium)		NPL, VITO	27 January 2022
Approved by (ESA)	Marc Bouvet	ESA	05 April 2022

Version history

Version	Date	Publicly available or private to consortium?
0.1	15/03/2022	Public

Contents

1. Introduction

1.1 Purpose and Scope

This document provides the LIME Toolbox (TBX) verification plan, therefore all the testing developed in parallel to the software of the toolbox itself. The scope is to ensure the correct functioning of the LIME TBX.

Testing can demonstrate that software has errors, but it cannot demonstrate that it is error free. This verification and validation plan outlines how software will be tested. Verification is aimed at making sure the system supplied meets the stated requirements, validation is about making sure that the software addresses what the customer really wants.

1.2 Applicable and reference documents

1.2.1 Applicable Documents

The following applicable documents are those specification, standards, criteria, etc. used to define the requirements of this task.

Number	Reference
[AD0]	ESA-EOPG-EOPGMQ-SOW-24. Improving the Lunar Irradiance Model of ESA.

1.3 Glossary

1.3.1 Abbreviations

Abbreviation	Stands For	Notes
ASD	Analytical Spectral Devices	Instrument manufacturer
Cimel	(Not an abbreviation)	Instrument manufacturer, also used as shorthand for instrument itself
EO	Earth Observation	
ESA	European Space Agency	Project customer
FOV	Field of View	
GIRO	GSICS Implementation of the ROLO Model	
GSICS	Global Space Based Inter-calibration System	
GUI	Graphical User Interface	
KO	Kick-off meeting	
LIME	Lunar Irradiance Model of ESA	
NPL	National Physical Laboratory	Project partner
ROLO	RObotic Lunar Observatory	
SoW	Statement of Work	
TBX	Toolbox	
TOA	Top of Atmosphere	
UVa	University of Valladolid	Project partner
VITO	Vlaamse Instelling voor Technologisch Onderzoek; Flemish Institute for Technological Research	Project partner

2. LIME Toolbox verification strategy

The LIME TBX implementation plan is described in deliverable D7. The SoW indicates the need of a SW verification plan that describes the tests needed to verify the LIME SW.

The strategy to ensure that the code is of high quality is to perform unit tests and end-to-end tests. This document describes the tests needed to verify the LIME TBX and how they will be implemented throughout the project.

The tests will be written prior to or in conjunction with the development of the code and will be included in a continuous integration pipeline using GitLab tools. The LIME TBX verification report will be produced to report the testing results of the delivered SW (deliverable D9).

The overall testing strategy is as follows:

- Most testing effort will go into the most complex parts.
- After that, the next priority area will be the most used part.
- The level of testing will factor in the level of risk.
- Users will be involved in the testing.
- Testing takes place at all stages.
- Tests will be quantitative, documented, repeatable and have clear acceptance criteria.

3. Tests

3.1 Unit tests

Every non-trivial function will have automatic unit tests that will be based on white-box testing techniques, specifically ones based on code coverage like Statement Coverage, Decision Coverage, etc., and black-box techniques like Equivalence Class Testing in order to homogenize the code.

These tests will be designed during implementation.

In order to perform these tests automatically, GitLab CI pipelines will be used, which allows the automatic execution of tests for every commit or push. These pipelines will be used not only for unit testing, but also for code-linting with “Pylint” in order to follow good practices and homogenize the code.

3.2 End-to-end tests

End-to-end tests will be performed for each use-case stated by the customer. They will be automated as much as possible. The aim is to detect defects caused by bringing the various parts of the system together.

Two kinds of end-to-end tests will be performed. Testing the software product as a whole which will be performed testing using the GUI, and testing parts of the code involving several (critical) functions, like testing exported functions from important modules without having to instantiate the GUI.

The end-to-end tests will be validated against synthetic data created (and quality-checked) manually outside the LIME-TBX. This synthetic data will be based on the existing ROLO model, the ASD measurements taken within this project, and separate tools (to do e.g. the convolution with the spectral response function).

3.3 Acceptance Tests

Acceptance testing checks that the system meets customer requirements, which for this project are described in Annex I. These tests are performed by the customer. These usually are similar to the

end-to-end tests, as they both contain all or most of the use-cases required by the customer, which are described in Annex II.

Down below, the proposed acceptance tests are described. They are based on the use cases.

3.3.1 Simulate lunar observations from Earth

The actor user will be able to simulate lunar observations for any observer position on Earth.

1. The actor user will inform the system that they wish to simulate lunar observations from Earth.
2. The system will ask the user to input the needed data for the location.
3. The actor user inputs valid data.
4. [Optional]: Use case “Simulate the lunar observations for a user-defined spectral response”.
5. The system informs the user that they can proceed with the calculation.
6. The actor user asks the system to simulate the lunar observations.
7. The system will show the user the result of the simulation.
8. The system will let the user export the output to a GLOD format file.
9. The user will choose to export the output to a GLOD format file.
10. The system will export the output to a GLOD format file.

3.3.2 Simulate lunar observation for custom selenographic latitude and longitude

The actor user will be able to simulate lunar observations for a custom observer and solar selenographic latitude and longitudes.

1. The actor user will inform the system that they wish to simulate lunar observations for custom selenographic latitude and longitude.
2. The system will ask the user to input the needed data for the simulation.
3. The actor user inputs valid data.
4. [Optional]: Use case “Simulate the lunar observations for a user-defined spectral response”.
5. The system informs the user that they can proceed with the calculation.
6. The actor user asks the system to simulate the lunar observations.
7. The system will show the user the result of the simulation.
8. The system will let the user export the output to a GLOD format file.
9. The user will choose to export the output to a GLOD format file.
10. The system will export the output to a GLOD format file.

3.3.3 Simulate lunar observations from a satellite position

The actor user will be able to simulate lunar observations from a satellite position, for at least the satellites defined in RNF104 (Annex I).

1. The actor user will inform the system that they wish to simulate lunar observations from a satellite position.
2. The system will ask the user to select the ESA satellite.
3. The user will select an ESA satellite.
4. The system will ask the user to input the needed data for the simulation.
5. The actor user inputs valid data.
6. [Optional]: Use case “Simulate the lunar observations for a user-defined spectral response”.
7. The system informs the user that they can proceed with the calculation.
8. The actor user asks the system to simulate the lunar observations.
9. The system will show the user the result of the simulation.
10. The system will let the user export the output to a GLOD format file.

11. The user will choose to export the output to a GLOD format file.
12. The system will export the output to a GLOD format file.

3.3.4 Simulate the lunar observations for a user-defined spectral response

The actor user will be able to simulate the lunar observations for a specific user defined instrument spectral response SRF.

1. The actor user will inform the system that they want to use a user defined spectral response SRF instead of a default satellite spectral band SRF.
2. The system will ask the user to choose the input file from which the spectral response SRF will be read from.
3. The actor user will select the SRF file.
4. The system will inform the user that the SRF has been loaded successfully.

3.3.5 Compare lunar observations from a remote sensing instrument

The actor user will be able to perform comparisons of lunar observations from a remote sensing instrument to the LIME output.

1. The actor user will inform the system that they wish to perform comparisons of lunar observations from a remote sensing instrument to the LIME output.
2. The system will ask the user to select the GLOD format file that stores the remote sensing instrument observations.
3. The user will select the correct file.
4. The system will show the user plots providing the data described in requirements RNF202 and RNF203.
5. [Optional]: Use case “Export plot”.

3.3.6 Export plot

The actor user will be able to export plots.

1. The actor user will inform the system that they wish to export a plot.
2. The system will ask the user to select the type of file it will be exported to (pdf or jpg).
3. The actor user will select a format.
4. The system will export the file following that format.

3.3.7 Manually check for updates

The actor user will be able to manually check for updates.

1. The actor user will inform the system that they wish to manually check for updates.
2. The system will inform the user that it’s looking for updates.
3. The system will tell the user that there are no updates, or that there are and they have been downloaded.

3.4 Documentation and reporting

All tests developed during the code implementation, will be included in the documentation and its results be used to compile the TBX verification report (D9). In this report, a summary of each test will be included, using the format given in table 1.

Unique test reference/run	A unique number to identify each test followed by "/" and a number incremented each time this test is run.	**Version of requirements**	A reference to the Functional Requirements document and its version
Requirement Reference to the requirement being tested (where applicable).			
Test purpose Specific test for this requirement (there may be more than one).			
Pre-requisites The setup required before the test can be executed.			
Test instructions A step-by-step set of instructions to follow by the tester.			
Acceptance criteria The expected result.			
Observation If the test failed then document what was seen. If it passed then this slot may be used for additional comments if required. Supplement by screen grabs etc if necessary.			
Pass/fail	Was the test passed?		
Tester	Who performed the test?		
Date	Date of test.		

Table 1. Summary file for each verification test of the TBX.

4. Annex I: Requirements elicitation

Requirements have been divided into four groups.

- **General user functionalities**: Requirements related to most main functionalities.
- **Comparisons**: Requirements related to the comparison functionality.
- **LIME Output**: Requirements related to the output and its format.
- **System requirements**: Other requirements like performing automatic updates, the programming language, etc.

4.1 General user functionalities

4.1.1 Functional Requirements

RF101: Allow users to simulate lunar observations for any observer's position around the Earth and at any time.

RF102: Allow users to simulate lunar observation for any observer/solar selenographic latitude and longitude (thus bypassing the need for their computation from the position/time of the observer).

RF103: Allow users to simulate the lunar observations for any user defined instrument spectral response SRF.

RF104: Allow users to simulate lunar observation for a single observation or for a time series of observations.

RF105: The user must be able to simulate lunar observations for an ESA satellite.

RF106: Allow users to either choose between default satellite spectral band SRF or select a user defined SRF (**RF103**).

4.1.2 Non-Functional Requirements

RNF101: The simulation input of a single lunar observation (**RF104**) must be introduced via the GUI.

RNF102: The simulation input for a time series of lunar observations (**RF104**) must be done via an input file.

RNF103: In order to simulate lunar observations from a satellite position (**RF105**) the user must provide an orbital scenario file in EOCFI compatible format.

RNF104: The ESA satellites available for selection must include ENVISAT, Proba-V, S2, S3, FLEX.

RNF105: The user defined SRF (**RF103**) must be defined via a user generated SRF file.

RNF 106: The user should be able to perform the simulation via command line using parameters or input files.

4.2 Comparisons

4.2.1 Functional Requirements

RF201: Allow performing comparisons of lunar observations from a remote sensing instrument to the LIME model output.

RF202: Allow exporting plots.

4.2.2 Non-Functional Requirements

RNF201: The remote sensing instrument observations (**RF201**) must be pre-stored in a GLOD format file.

RNF202: The comparison plots shall provide: relative differences between measured and modeled lunar irradiance/reflectance vs. time and vs. lunar phase angle.

RNF203: The comparison plots shall display statistical indicators (mean relative difference, standard deviation of the mean relative difference, temporal trend if applicable, number of comparison samples, etc.)

RNF204: The exported plots must be in .jpg or .pdf format.

RNF205: The user should be able to perform the simulations via command line using parameters or input files.

4.3 LIME output

4.3.1 Functional Requirements

RF301: The LIME TBX shall output simulated lunar disk irradiance or reflectance.

RF302: The LIME TBX shall output the simulated lunar disk degree of polarization.

RF303: The LIME TBX shall output simulated lunar irradiance or reflectance associated uncertainty.

RF304: The LIME TBX shall output the simulated lunar degree of polarization's associated uncertainty.

RF305: Allow the visualization of the user defined spectral response used for the spectral integration of the LIME output into a sensor spectral band.

4.3.2 Non-Functional Requirements

RNF301: The simulated lunar disk irradiance or reflectance (**RF301**) must be in the spectral range of 400 nm to 2500 nm.

RNF302: The simulated lunar disk degree of polarization (**RF302**) must be in the spectral range of 400 nm to 2500 nm.

RNF303: The LIME simulated output shall be available to be exported to GLOD format files.

RNF304: The LIME version number shall be visible on all outputs (plots/files) of the TBX.

4.4 System requirements

4.4.1 Functional Requirements

RF401: The LIME TBX must be able to perform automatic updates of the LIME coefficients.

RF402: The user shall be able to select past LIME coefficients with whom perform the calculations.

4.4.2 Non-Functional Requirements

RNF401: The LIME coefficients of the automatic updates (**RF401**) must be stored on a dedicated repository.

RNF402: The LIME TBX shall read the database of lunar observations formatted in GLOD format

RNF403: The LIME TBX shall use the EOCFI as orbit propagator and to derive satellite orbital positions.

RNF404: Be to the largest extent platform/operating system independent.

RNF405: Run at least under Windows / mac OS / Linux operating systems.

RNF406: Be to the largest extent a self-installing SW package.

RNF407: The LIME TBX shall be to the largest extent developed in Python.

RNF408: The LIME TBX code shall be available on a password protected web repository allowing versioning of the software (e.g., GitHub).

5. Annex II: Use cases

The use cases are:

Simulate lunar observations from Earth: The actor user will be able to simulate lunar observations for an observer position on Earth.

Simulate lunar observation for custom selenographic latitude and longitude: The actor user will be able to simulate lunar observations for a custom observer and solar selenographic latitude and longitudes.

Simulate the lunar observations for a user-defined spectral response: The actor user will be able to simulate the lunar observations for a specific user defined instrument spectral response SRF.

Simulate lunar observations from a satellite position: The actor user will be able to simulate lunar observations from a satellite position, for at least the satellites defined in RNF104.

Compare lunar observations from a remote sensing instrument: The actor user will be able to perform comparisons of lunar observations from a remote sensing instrument to the LIME output.

Export plot: The actor user will be able to export plots.

Manually check for updates: The actor user will be able to manually check for updates.