SPIS-GEO

Simplified MEO/GEO tools for spacecraft charging

Final presentation

B. Thiebault[1], J. Forest[1], Benjamin Jeanty-Ruard[1],
J.-C. Mateo Velez[2], Pierre Sarrailh[2], Alain Demairé[3], Bjarne Andersson[3], Patrice Pélissou[4], Marc Sevoz[4]

Final presentation days - March 20th 2013

Artenum (1), in partnership with ONERA (2),
OHB-Sweden (3) and EADS Astrium (4)

Outline

Introduction
- SPINE community
- SPIS-GEO/MEO project
  - Objectives
  - Team and work breakdown structure
  - Deliverables

User requirements
- Methodology
- Requirements for the numerical kernel
- Requirements for the user interface
- Validation requirements

A major evolution
- Software lifecycle management and quality
- Technologies

SPIS-UI
- User interface overview
- Improvements summary

SPIS-NUM

Validation campaign

Conclusion and perspectives

Introduction (1/4)

SPIS

- 9 major releases since 2003
- More than 6 500 downloads (all versions/branches)
- More than 1 200 downloads for SPIS 4.3
- Open to news fields: ESD, instrument calibration, propulsion, dusty plasmas
- Components mutualised with other communities
- New SPIS generation coming
  - SPIS-GEO
  - SPIS-Science
  - AISEPS
  - SPIS Dust

SPINE, an active community

- http://dev.spis.org
- More than 600 registered members (and around 2 new registrations a week)
- About 20 active contributors (including SMEs, major industrial actors and academics)
- Regular SPINE meetings
- Training courses
- Numerous publications
  - Last publications at the 12th SCTC
  - Future publications in preparation (e.g. Web3D 2013 conference)

Introduction (2/4)

SPIS-GEO/MEO project

- ESTEC/ESA contract
  - Technical officer: David Rodgers
  - Consortium: Artenum, ONERA, EADS-Astrium, OHB-Sweden
- Developments based on the existing SPIS (same concepts, same numerical kernel with new models, new user interface)
- Adaptation to industrial needs and in particular to MEO/GEO orbit constraints

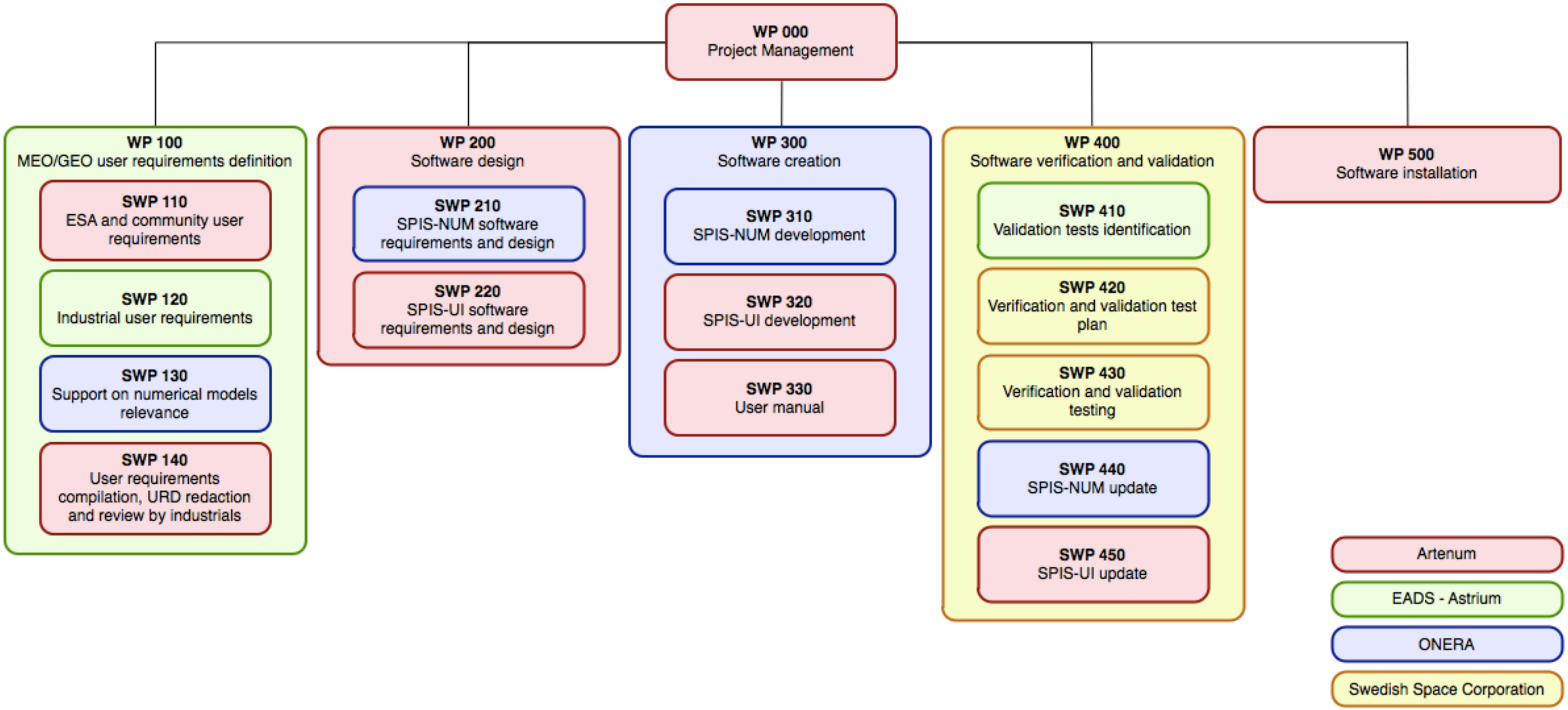
Objectives of the project

- Identification of industrial users needs
- Simplified user interface adapted to engineering applications
- Physical models adapted to MEO/GEO orbits and commercial space platforms
- Tested software against in-flight observations and existing codes

Introduction (3/4)

Team and work breakdown structure

- Consortium including industrial end users
- Validation phase performed by non developers, on real-life test cases



Introduction (4/4)

Deliverables

Ref	Description
D1-URD	User Requirements Document
D2-SQAP	Specific Quality Assurance Plan
D3-SRD	Software requirements document
D4-ADD	Architecture Design Document
D5-DVVTP	Draft Validation and Verification Test Plan
D6-SDD	Software Design Document
D7-SUM	Software User Manual
D8-VVTP	Validation and Verification Test Plan
D9-VVTR	Validation and Verification Test Report
D10-SATPR	Software Acceptance Test Plan and Report
D11-FR	Final Report

User requirements (1/2)

Methodology

- User requirements were gathered from various sources:
  - Industrial end users provided their requirements directly (WP leader)
  - A 2008 CNES study gathered community user requirements
  - ESA provided their own requirements
- An exhaustive list of 59 requirements has been identified
- Requirements were sorted by priority and a trade-off was made with respect to technical constraints and feasibility

Main requirements for the numerical kernel

- Eclipse exit with material conductivity evolution
- Pre-defined worst-case and typical environments
- Spacecraft self-shadowing
- Performance improvement

User requirements (2/2)

Main requirements for the user interface

- Overall simplification of the user interface
- Cross platform compatibility (Linux 64bits, MacOSX, Windows XP 32bits)
- Improved project persistency (quicker project loading/saving, standard file formats)
- Thin wires and plates
- Real-time monitoring
- Improved robustness and performances

Validation requirements

- Code should be validated against existing software
- Code should be validated against in-flight measurements

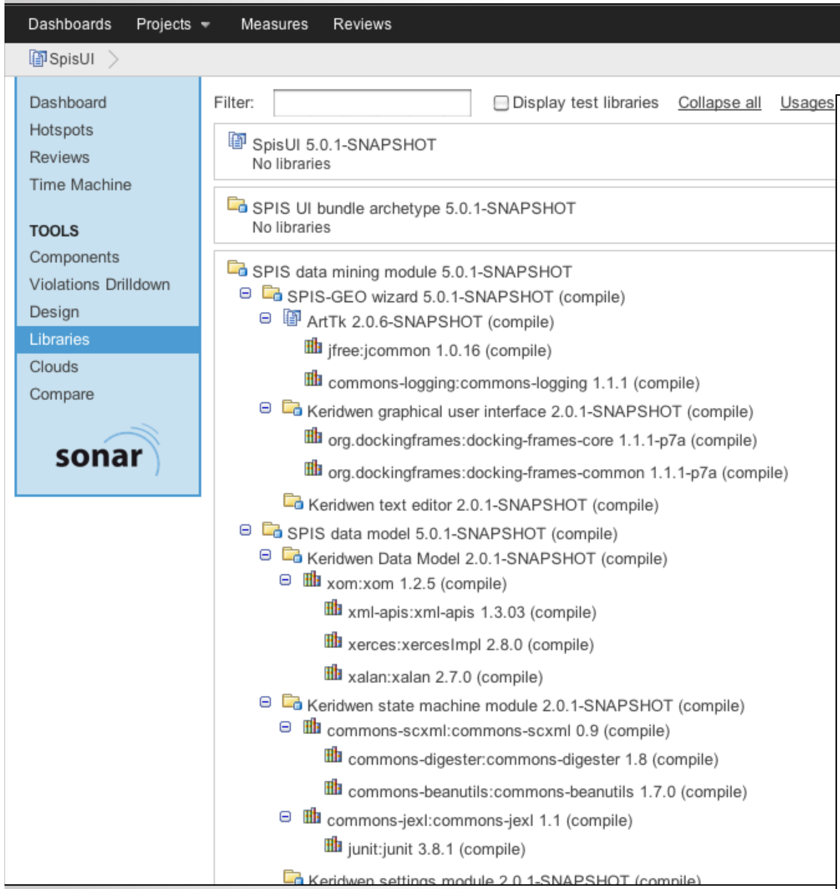
A major evolution (1/7)

Software lifecycle management and quality

- SPIS was first developed in 2002
- Industrial development tools and development standards were just emerging
- Standard tools and techniques are now widely used by the software industry
- Full redesign of the framework based on modern architecture and design patterns
- Based on industry-standard development tools and techniques:
  - Compilation, deployment and dependency management: Apache Maven
  - Source code versioning: Subversion
  - Automatic quality controls: Sonar
  - Continuous integration: Hudson

A major evolution (2/7)

Dependency management



Licenses

BSD style: XStream Core

Public Domain: MXP1: Xml Pull Parser 3rd Edition (XPP3), XML Pull Parsing API

Unknown: "Java Concurrency in Practice" book annotations, Codec, commons-beanutils, concjunit, docking-frames-common, docking-frames-core, vtk, xalan

Apache License, Version 2.0: osgi.core

The Apache Software License, Version 2.0: Commons IO, Commons JEXL, Commons Logging, Commons SCXML, Digester, Log4j, Xerces2 Java Parser, xml-apis

GNU Lesser General Public License: jxl

Indiana University Extreme! Lab Software License, vesion 1.1.1: MXP1: Xml Pull Parser 3rd Edition (XPP3)

The BSD License: Fortran to Java ARPACK

Lesser General Public License (LGPL): ArtTk, Keridwen Data Model, SwingX Core

General Public License (GPL): Frida, General utiliy classes for the IME, Global parameters module., Gmsh plugin for Penelope, Keridwen Messaging System, Keridwen graphical user interface, Keridwen internationalization module, Keridwen settings module, Keridwen state machine module, Keridwen text editor, Nisaba, Penelope NetCDF plugin, Penelope Vtk Plugin, SPIS Instruments, SPIS data model, SPIS-GEO wizard, SPIS-NUM, Xml plugin in penelope

(QPL): Penelope Core module

GNU Lesser General Public Licence: jcommon, jfreechart

Apache License: HttpClient

The GNU Lesser General Public License, Version 2.1: XOM

Common Public License Version 1.0: JUnit

MIT License: SLF4J API Module, SLF4J Simple Binding

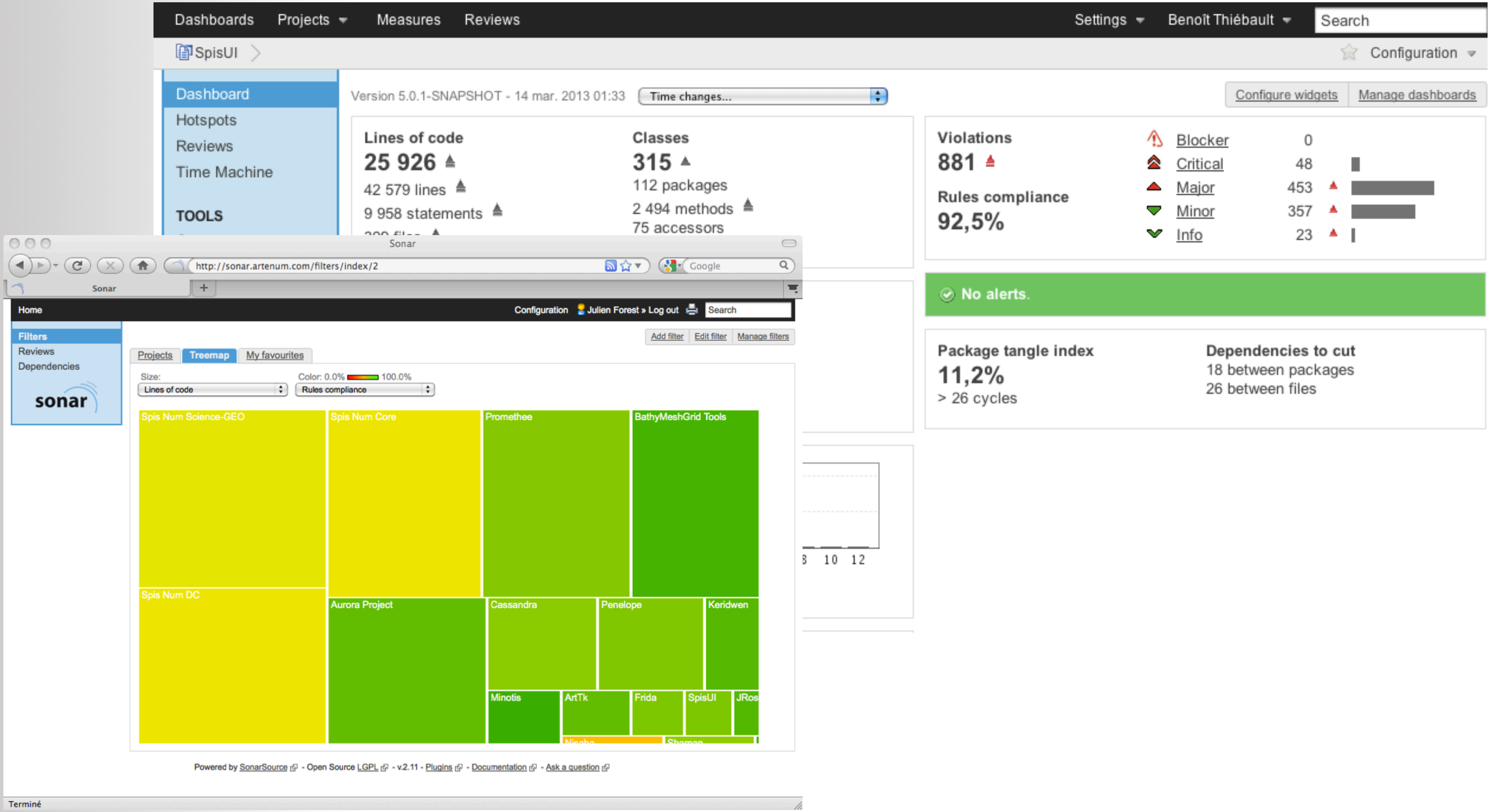
GNU Lesser General Public License 2.1: GNU Trove

(MIT-style) netCDF C library license: NetCDF-Java Library, Unidata Common

ARTENUM, PARIS
Science & Groupware

A major evolution (3/7)

Quality controls



ARTENUM, PARIS
Science & Groupware

A major evolution (4/7)

Technologies

KERIDWEN
INTEGRATED MODELLING ENVIRONMENT

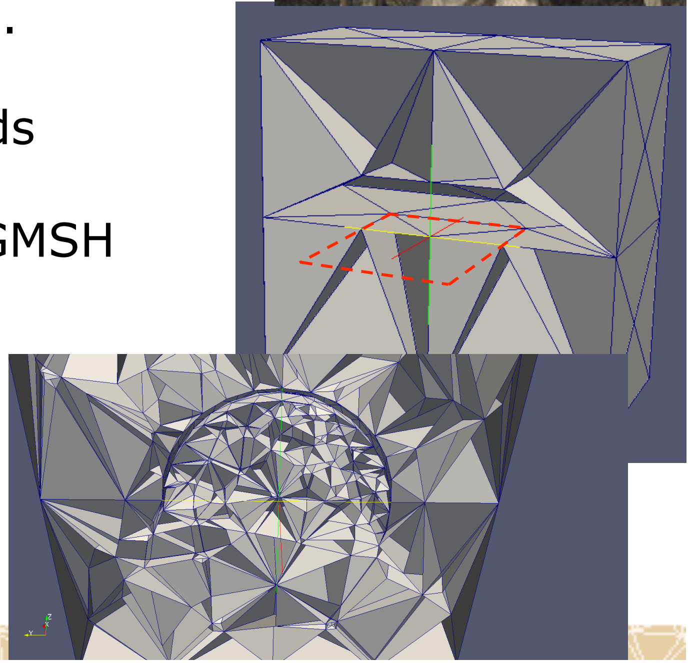
- SPIS is based on the **Keridwen 2.0**, a generic Integrated Modelling Environment (IME) and benefits from its capabilities:
  - Fully written in Java and multi-threaded
  - Based on industry standard OSGi modules system
  - Robustness, performances and reliability
  - High interoperability with other modeling tools (e.g G-Eclipse, ESABASE-2...)
  - Ready for future distributed architecture
  - Loosely coupled messaging system
  - Generic data persistency and I/O capabilities, based on standard formats (NetCDF, XML, etc.)
  - Open-source
  - Benefits from a trans-communities dynamics by sharing the maintenance effort over several communities

A major evolution (5/7)

Technologies

- **Penelope mesh library:**
  - Handle unstructured 3D meshing (tetrahedra)
  - Support rich data fields (scalars, vectors, objects)
  - Mesh and data fields operators (splitting, mapping...)
  - Improved performances compared to JFreeMesh
  - Dynamical management of mesh elements (necessary for mesh splitting and mesh edition)
  - Rich I/O capabilities: GMSH, XML, NetCDF, VTK...
  - Full support of latest GMSH file formats:
    - Support of format 2.2, including deployed fields
  - Includes the Java wrapping of GMSH
    - Better integration of GMSH (direct control of GMSH objects in code)
    - Improved performances (direct memory exchange)
    - Contribution to the GMSH community

Penelope's Mesh Library



A major evolution (6/7)

Technologies

- **Frida, the generic properties handling library:**
  - Generic handling of "Properties" (in the largest sense of the term) being applied on CAD models, meshes or logical systems:
    - Used to define, edit and allocate all required properties (materials, BC and IC) to Groups in SPIS
  - Provide rich properties editions and mapping tools
  - Java-based and open-source
  - Handle a large set of data types (scalars, vectors, tables, strings, objects...)
  - Hierarchized and module data structure
  - Expandability to support new types of materials and properties
  - Rich I/O capabilities:
    - XML base format
    - Import of NASCAP-2K materials
    - Import of SPIS legacy (4.3 and older) materials

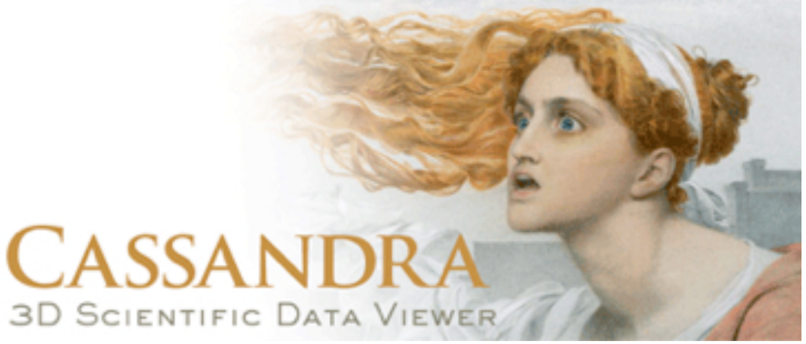
Frida, the painter

Frida is a generic properties edition and storage tool already used in several other contexts than SPIS.

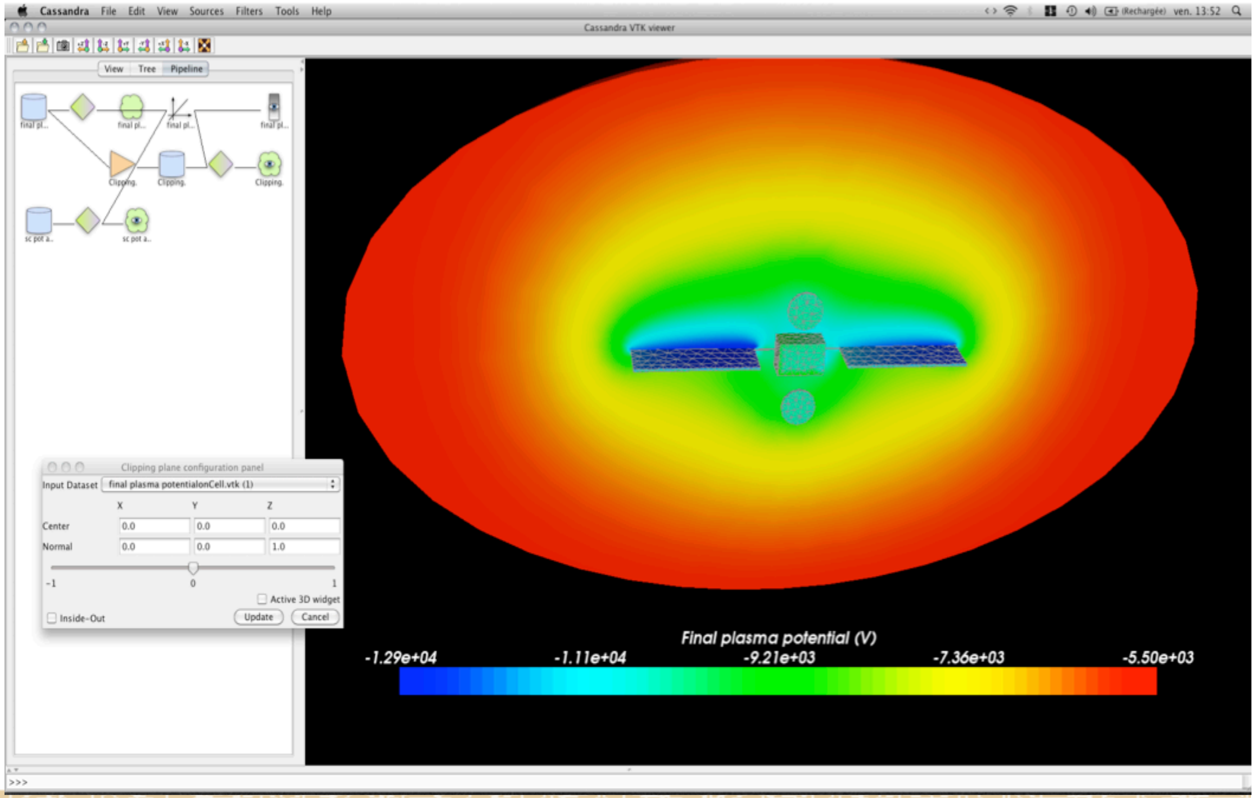
Artenum, Paris
Science & Groupware

A major evolution (7/7)

Technologies



- Cassandra 2.5, 3D Scientific Data Viewer
  - Graphic visualisation pipeline editor for complex and tailored post-processings
  - Simpler to use (improved LAF, 3D widgets)
  - Based on Java and VTK
  - Open source
  - New filters
  - New data conversion capabilities
  - Collaborative capabilities and 3D view sharing over http with the *CassandraCloud* service.



SPIS-UI (1/24)

User interface overview

- SPIS 5 uses a wizard-based approach to guide the user through the modelling chain while guaranteeing data consistency

SPIS-UI (2/24)

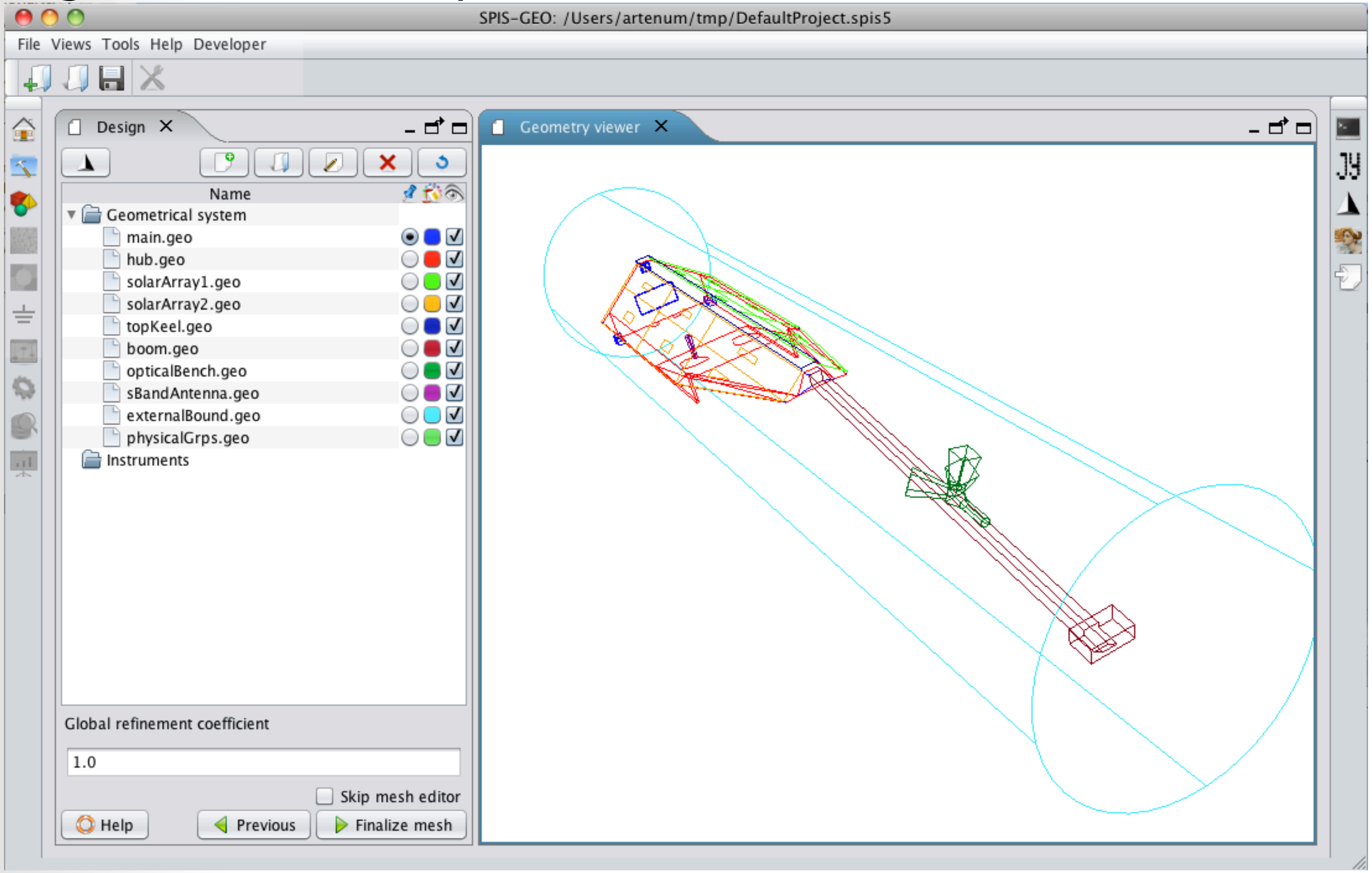
User interface overview

- This deep refactoring should:
  - Help the user, including non-expert, to perform properly the whole modelling process
  - Provide a more homogeneous GUI, easier to understand
  - Provide a step-by-step validation of the modelled system
  - Provide a better awareness of the simulation and generated data with reinforced monitoring capabilities
  - Reinforce the robustness of the whole application and stored data
  - Reduce the learning curve
  - Reduce the “incompressible minimal cost” for a mission study
  - Reinforce the global confidence in the produced results
  - Keep the advanced capabilities of the SPIS Legacy GUI

SPIS-UI (3/24)

User interface overview – Geometry editor

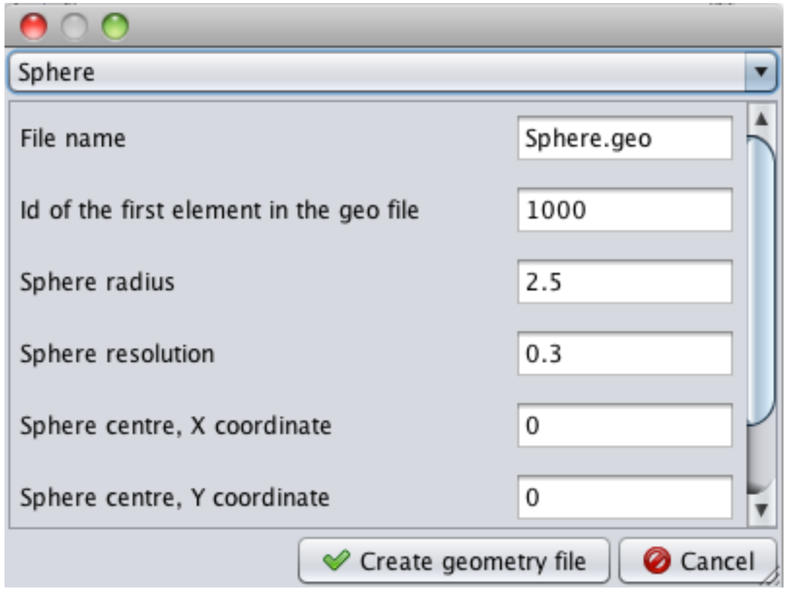
- Better integration of components resulting in more homogeneous user experience



SPIS-UI (4/24)

User interface overview – Geometry editor

- WYSIWYG editor: 3D view of the edited geometry
- Geometry editor allows to create new geometry files from templates:
  - 2D Thin plate
  - Cubic box
  - Sphere
- Open existing CAD files in various formats:
  - Gmsh .geo files
  - STEP files
  - STL tesselated files (exported from Geant4 for instance)
- Support of Gmsh composed .geo files
- Integrated text editor for CAD files modifications
- Possibility to open the selected file in Gmsh
- Possibility to apply a global mesh refinement coefficient

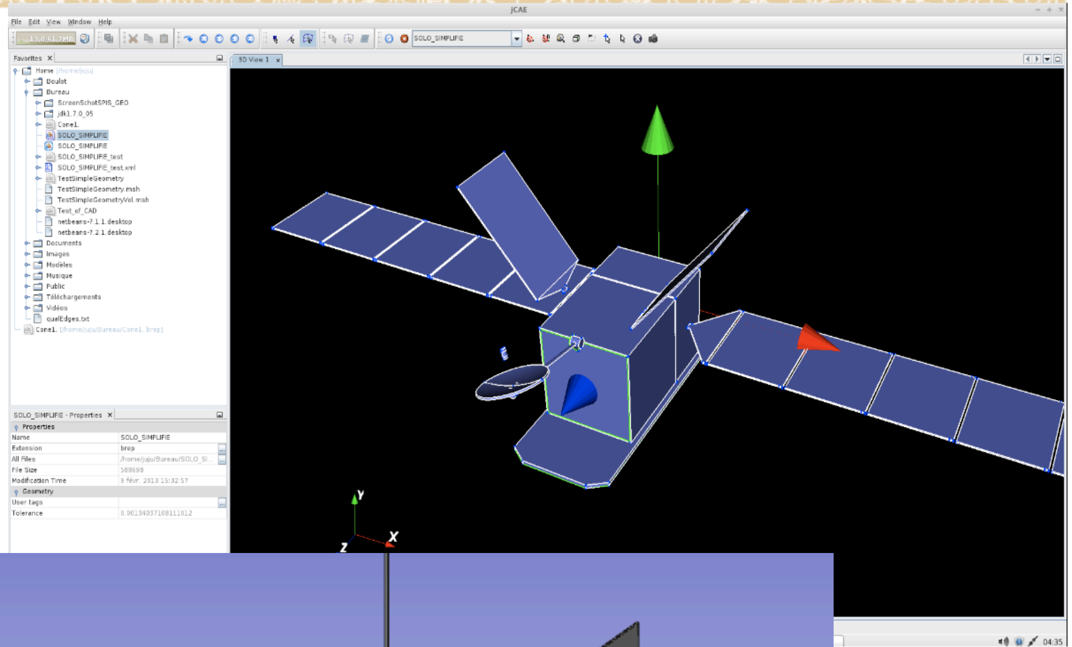


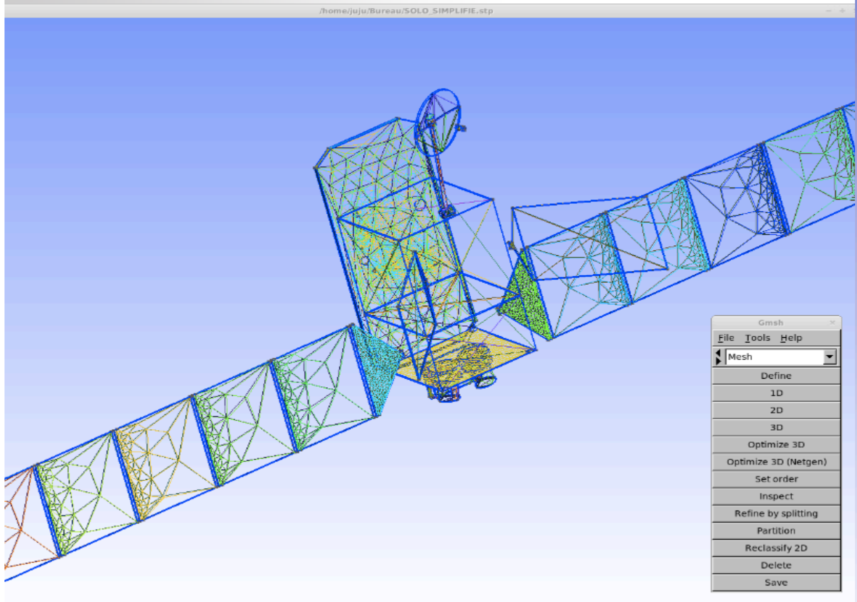
ARTENUM, PARIS
Science & Groupware

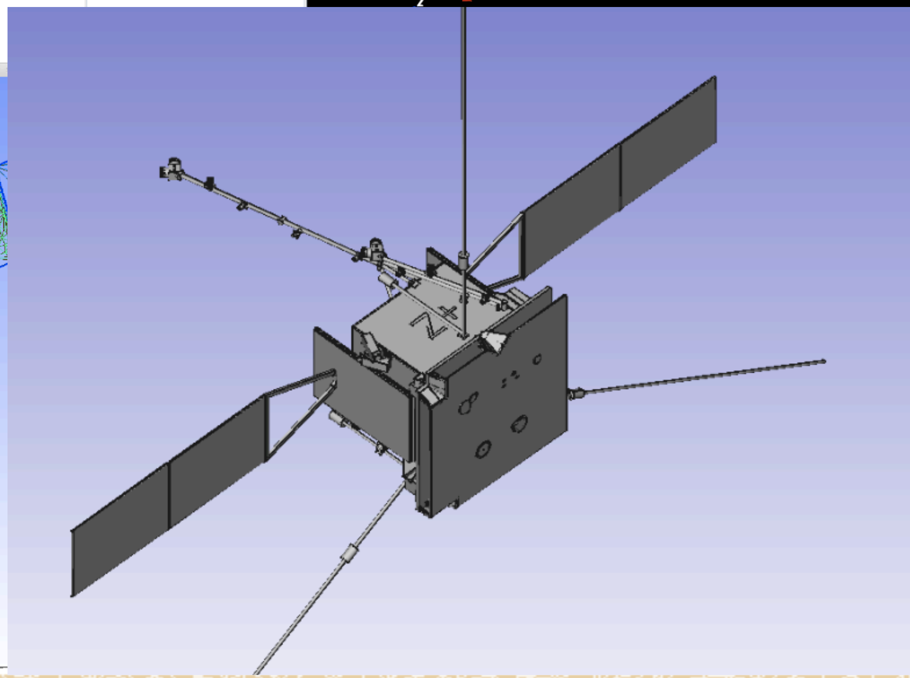
SPIS-UI (5/24)

CAD Interoperability

- STEP importer through Gmsh (OpenCascade based)
- Tessellated geometries importer (Penelope)
- Interfacing with external CAD tools (e.g. JCAE, FreeCAD, CATIA...)



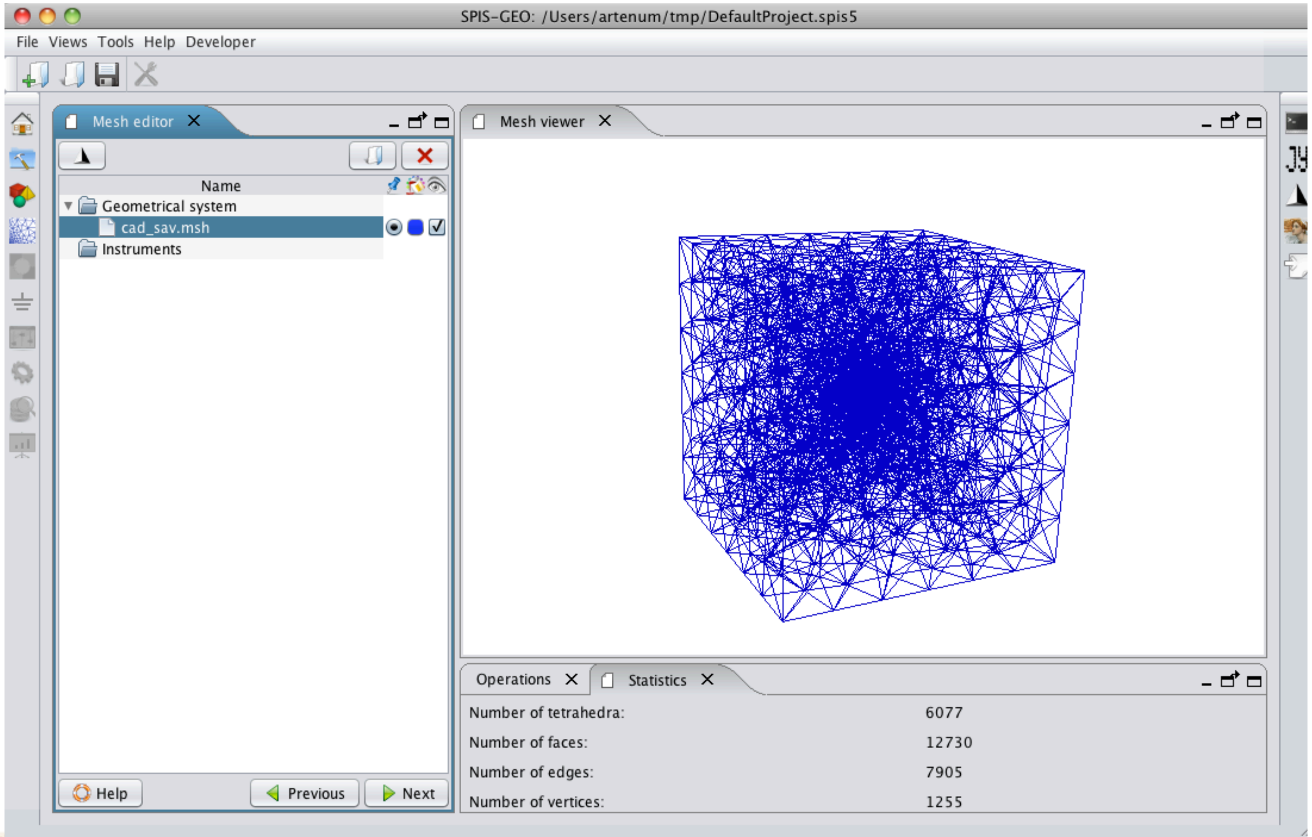




SPIS-UI (6/24)

User interface overview – Mesh editor

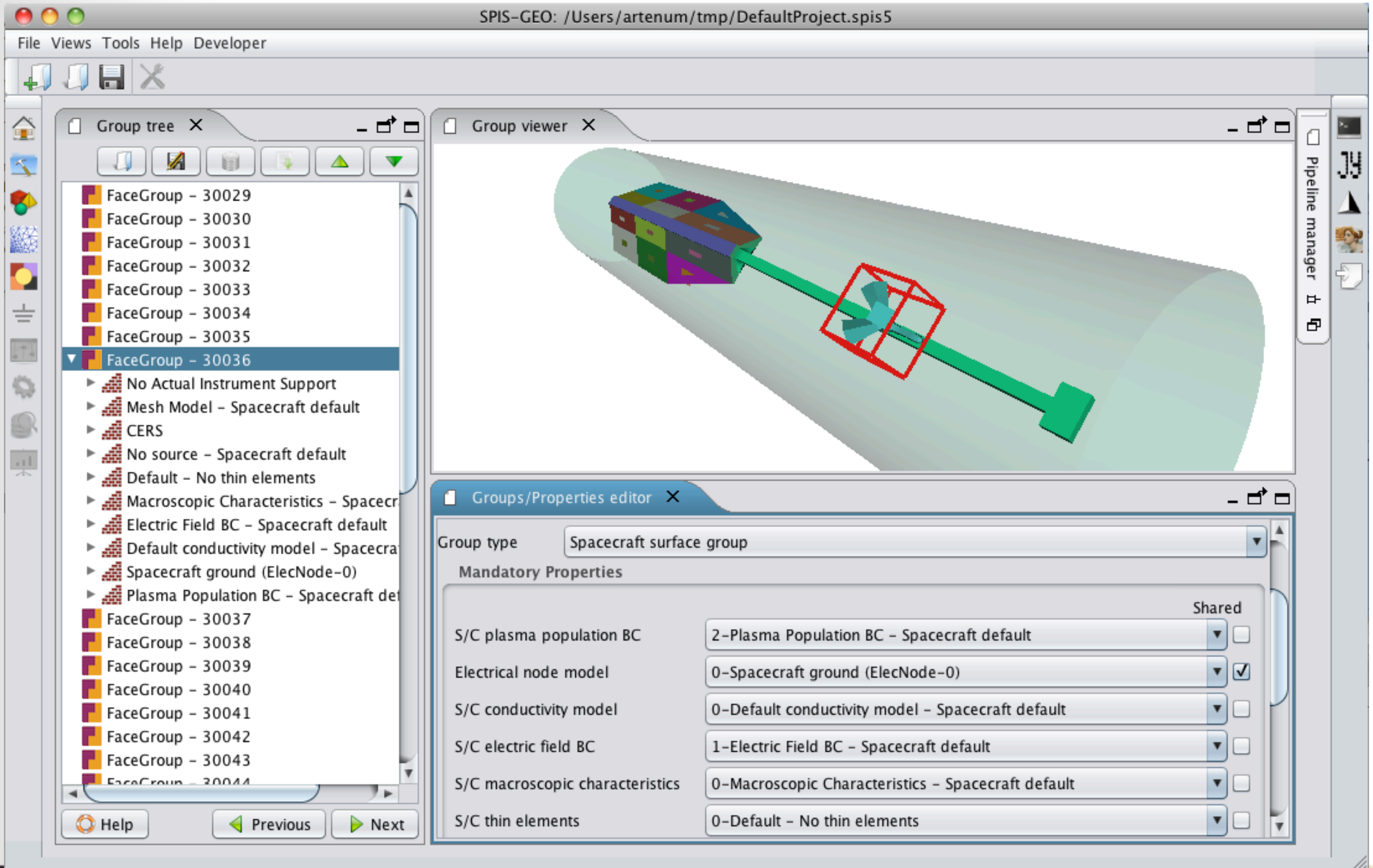
- Mesh statistics
- Possibility to directly import existing mesh files in various formats (Gmsh, STL, UNV...)
- Mesh operations mechanisms: currently, one operator provided to change the orientation of mesh faces (used for 2D thin elements)
- Mesh statistics information



SPIS-UI (7/24)

User interface overview – Group editor

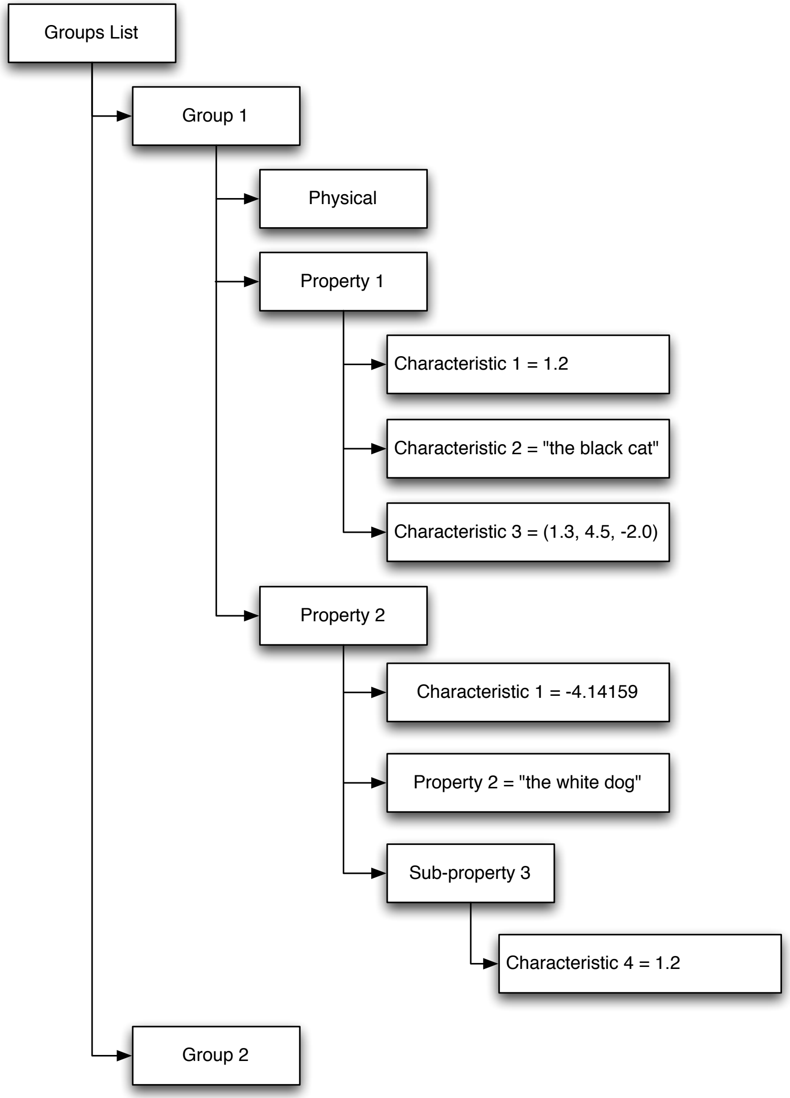
- Simplified edition of the group properties allocation and edition
- Based on Frida library



SPIS-UI (8/24)

User interface overview – Group editor

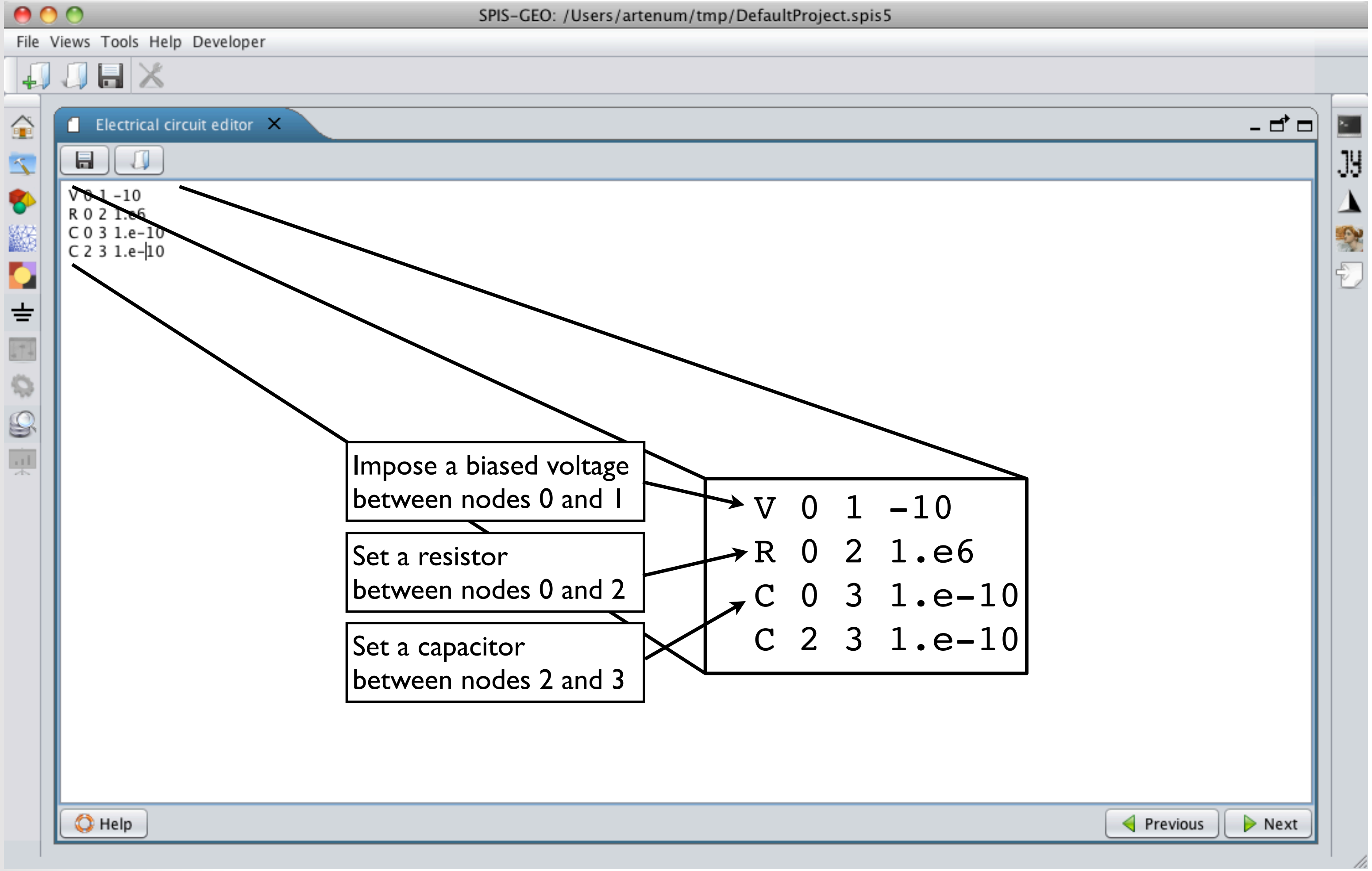
- Major simplification for the user
- WYSIWYG editor: highlighting of the selected group in the 3D view
- Pre-selection of the correct properties for a given group
- Support of thin elements (2D plates and thin wires)
- Possibility to import/export group settings in XML files
- Advanced properties editors
- Frida is very flexible and allows the creation of new properties by extension of existing ones or composition



SPIS-UI (9/24)

User interface overview – Electrical circuit editor

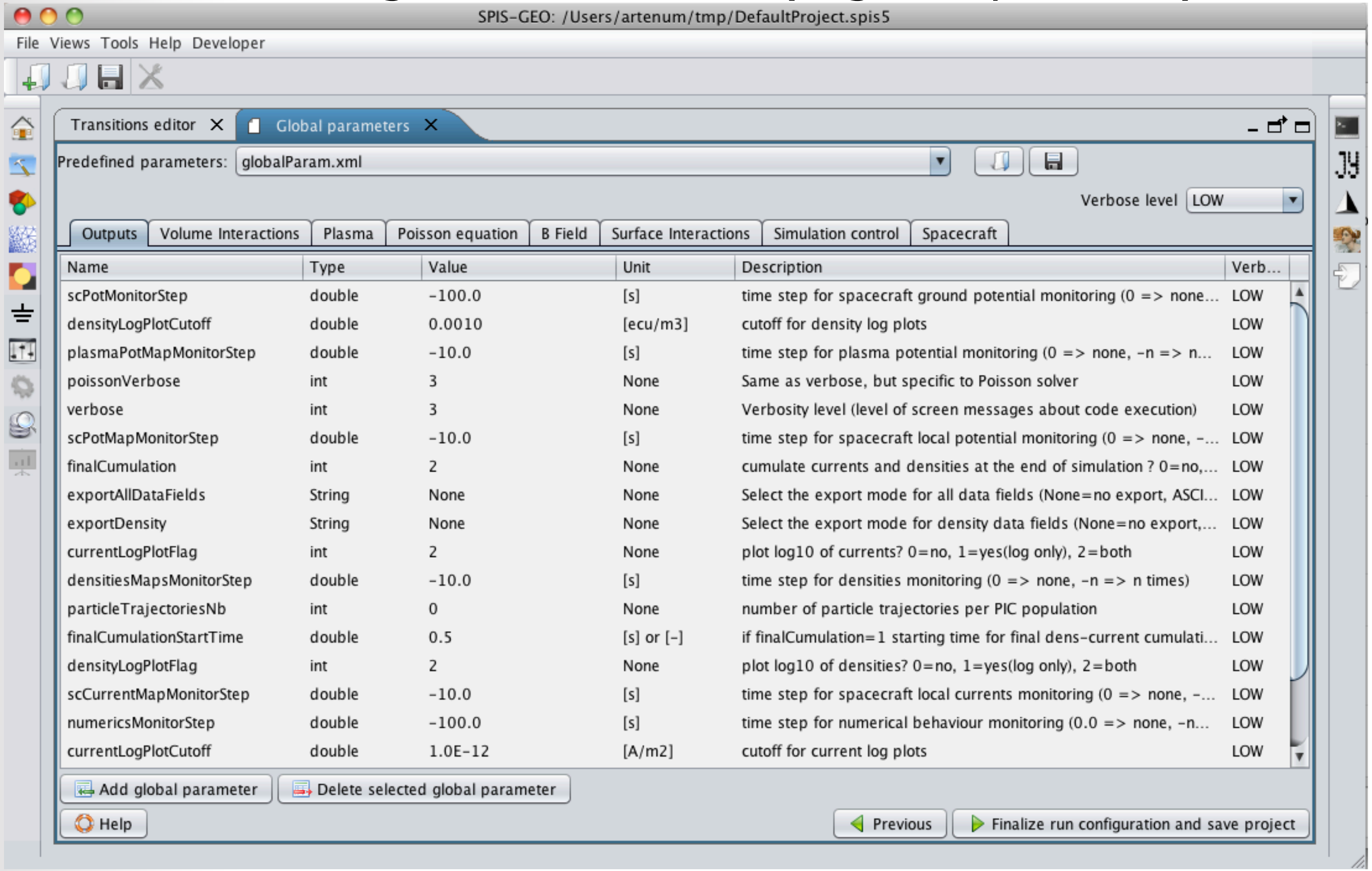
- Now an explicit step of the modelling chain
- Integrated text editor to modify and import circuit files



SPIS-UI (10/24)

User interface overview – Global parameters editor

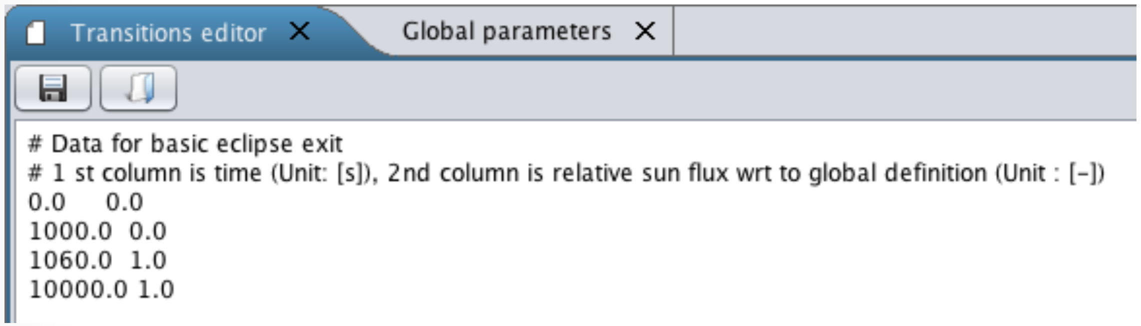
- Improved global parameters editor
- New editor to configure transitions (e.g. eclipse exit)



SPIS-UI (11/24)

User interface overview – Global parameters editor

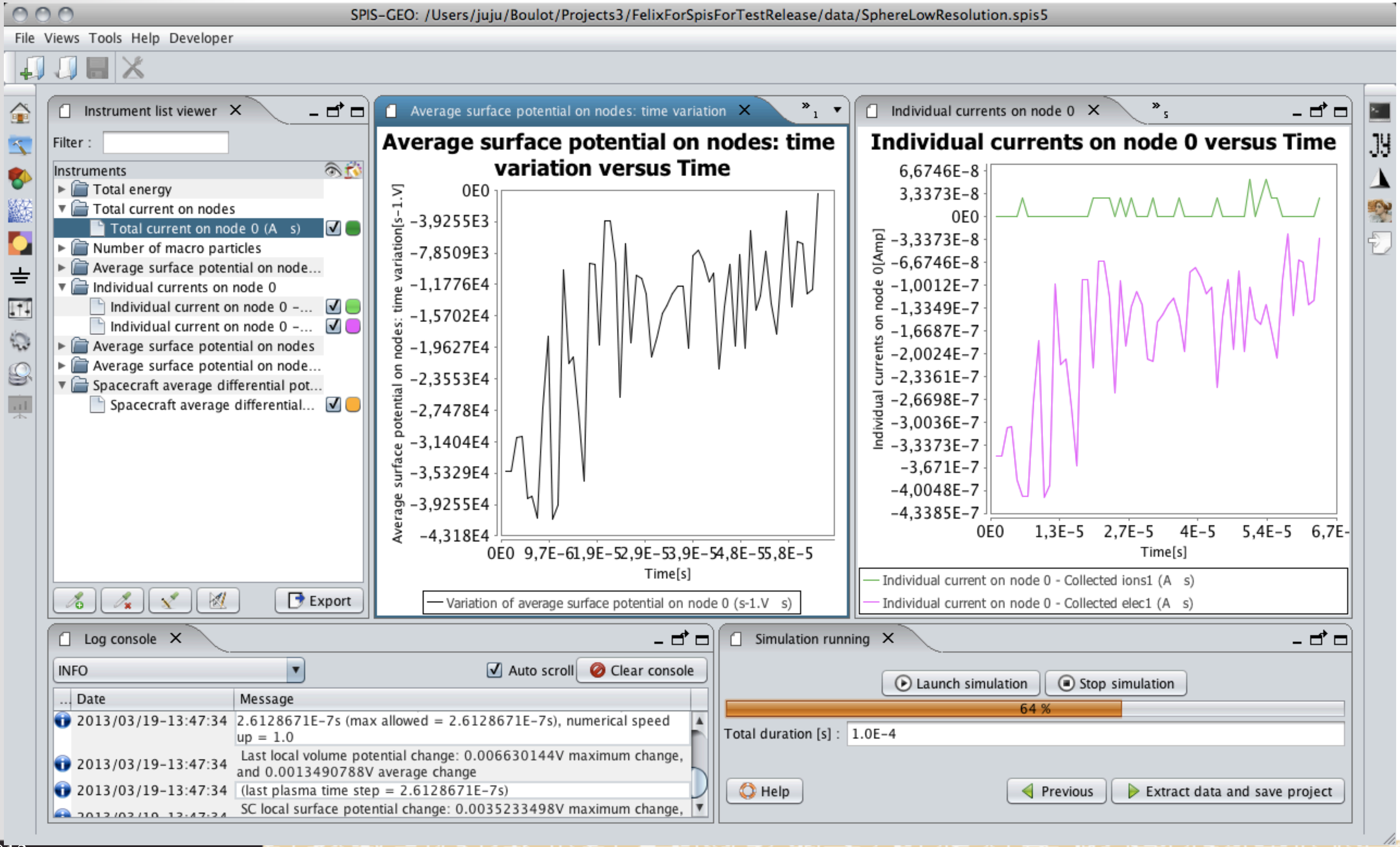
- Simplified edition of the global parameters with the help of a new filter that hides expert settings from the view by default
- Typical environment settings provided by default
- Preset worst case environment settings (ECSS and NASA worst cases)
- Possibility to sort table columns by ascending or descending order
- Data types checking
- Import/export of global parameters either in XML or Excel format
- New transitions editor to configure eclipse exit modelling



SPIS-UI (12/24)

User interface overview – Simulation control & monitoring

- Simulation control: start / pause / stop
- Real-time monitoring: displays key parameters of the simulation



SPIS-UI (13/24)

User interface overview – Simulation control & monitoring

- Reinforced awareness of the simulation evolutions through dynamical monitors
  - Real time monitoring
  - Converging factors
- Progressive generation of simulation outputs
- Simplified access to generated data

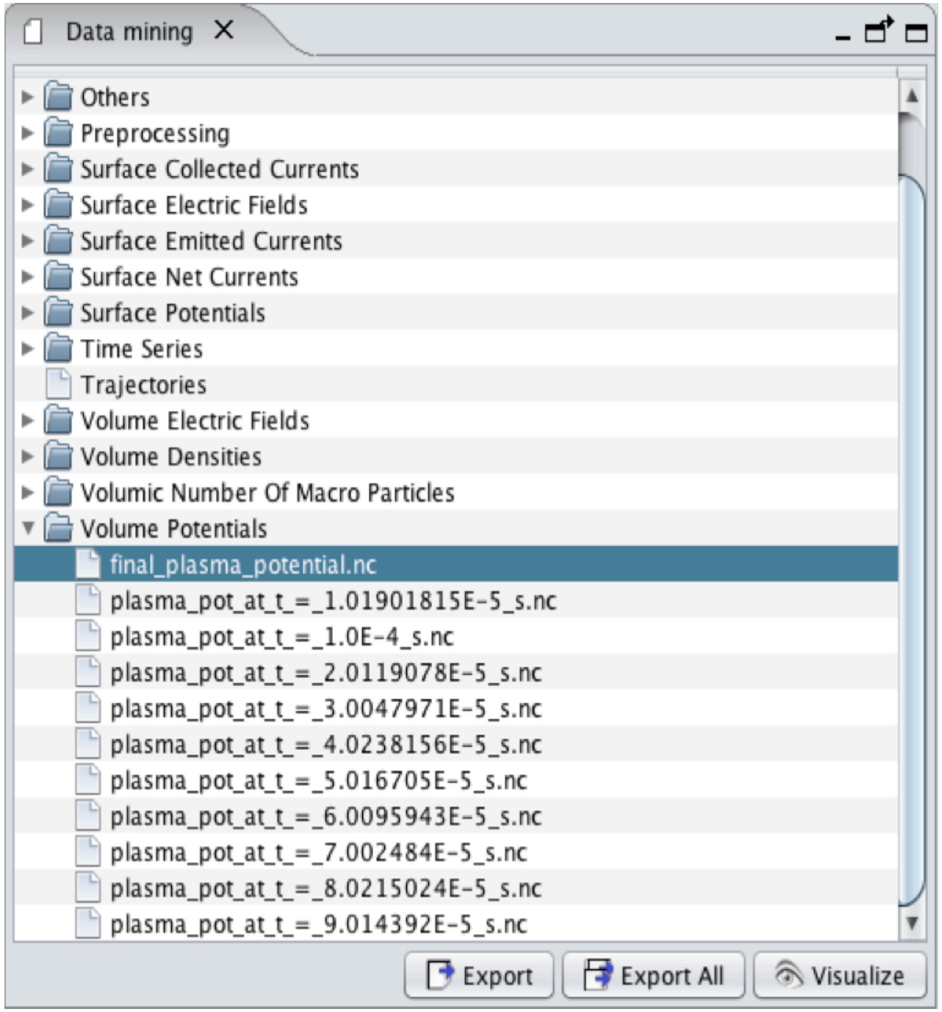
Better and Simpler Control
of the Simulation by the User

Faster and easier access to results in
an operational context

SPIS-UI (14/24)

User interface overview – Data mining

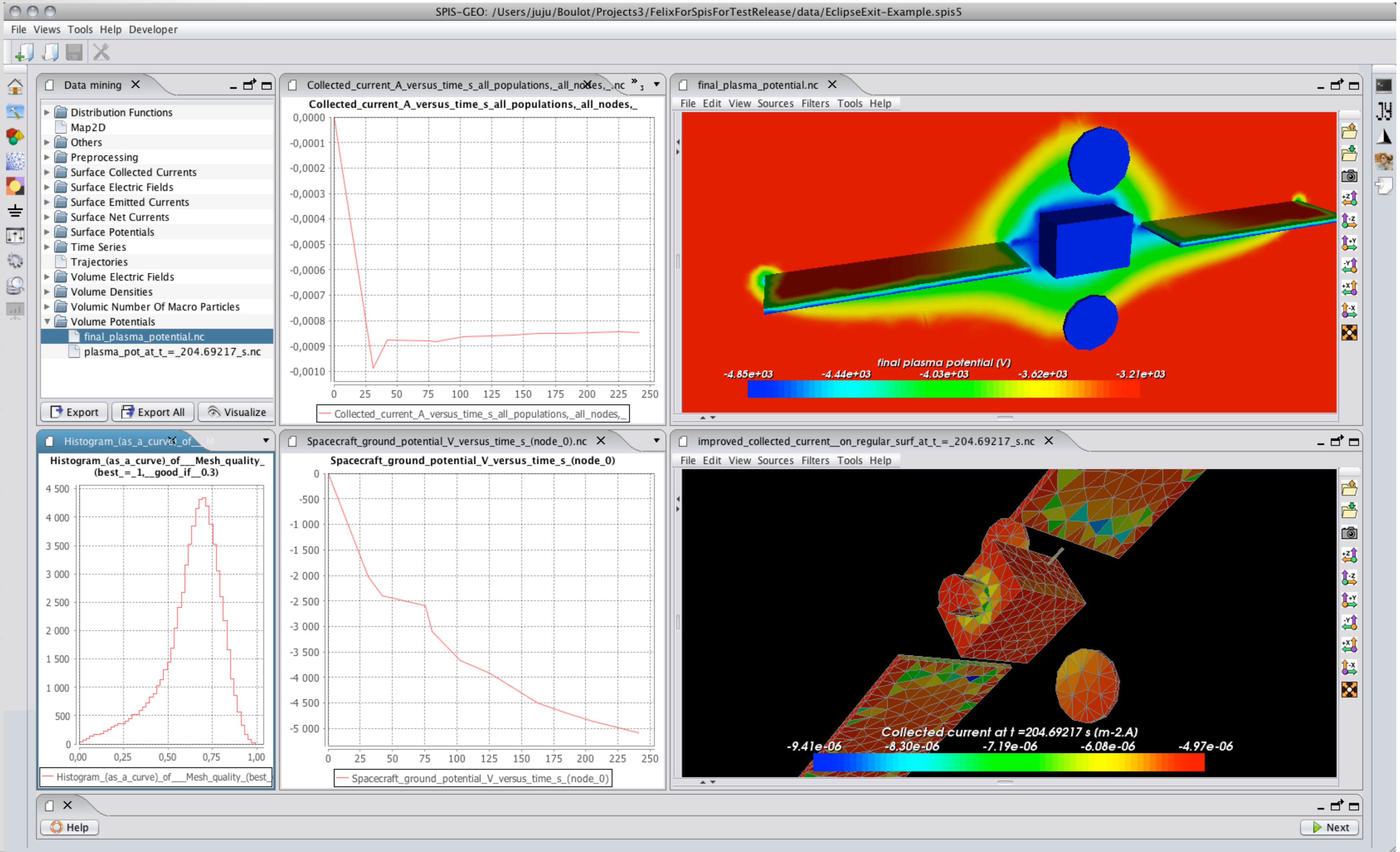
- 2D/3D post-processing tools
- Simplified data extraction and mining
- Rich multiple export capabilities
- Improved performances and memory cost
- Faster data saving and reloading (about 10 times w.r.p SPIS Legacy)



SPIS-UI (15/24)

User interface overview – Post-processing

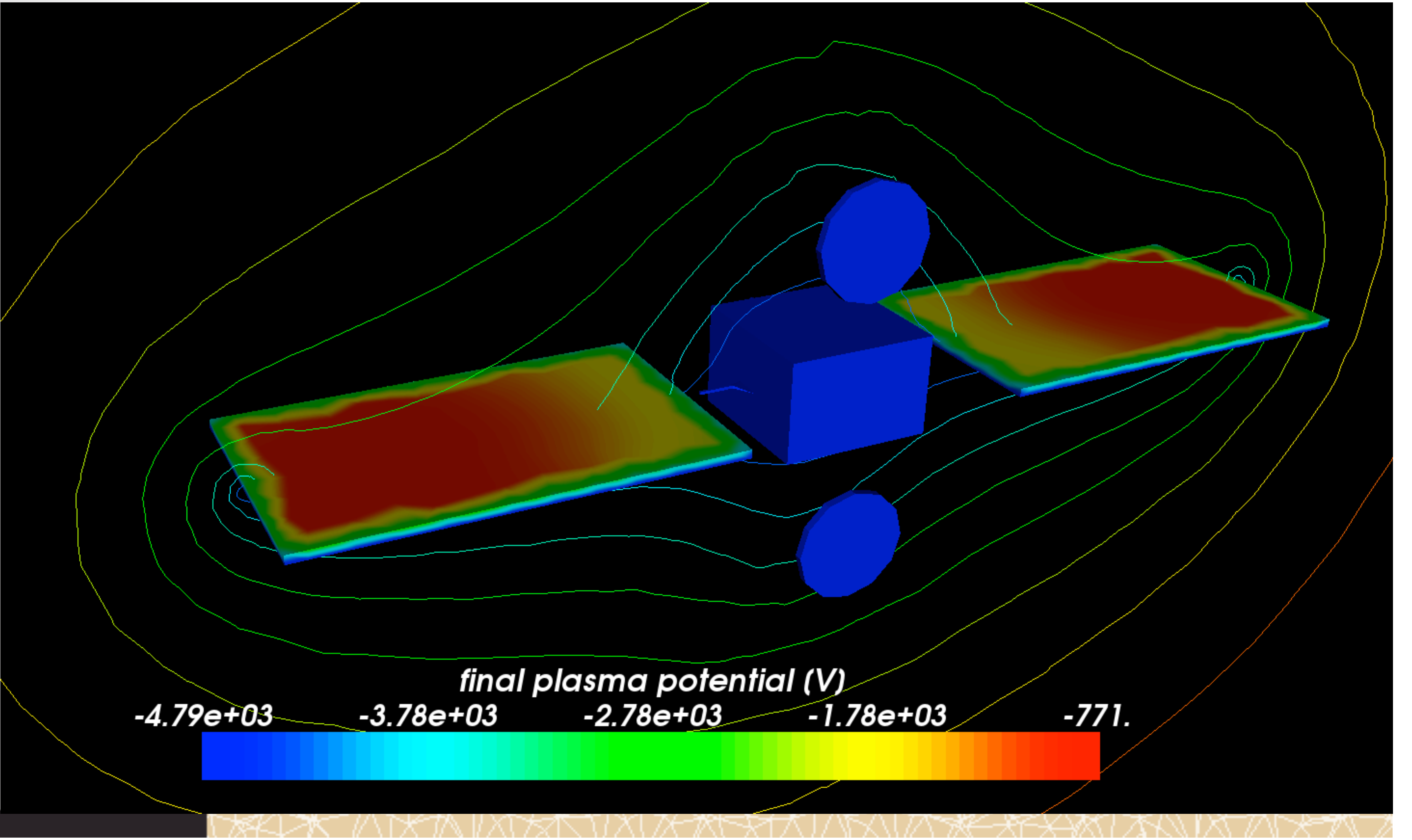
- Rich 2D/3D post-processing tools



SPIS-UI (16/24)

User interface overview – Post-processing

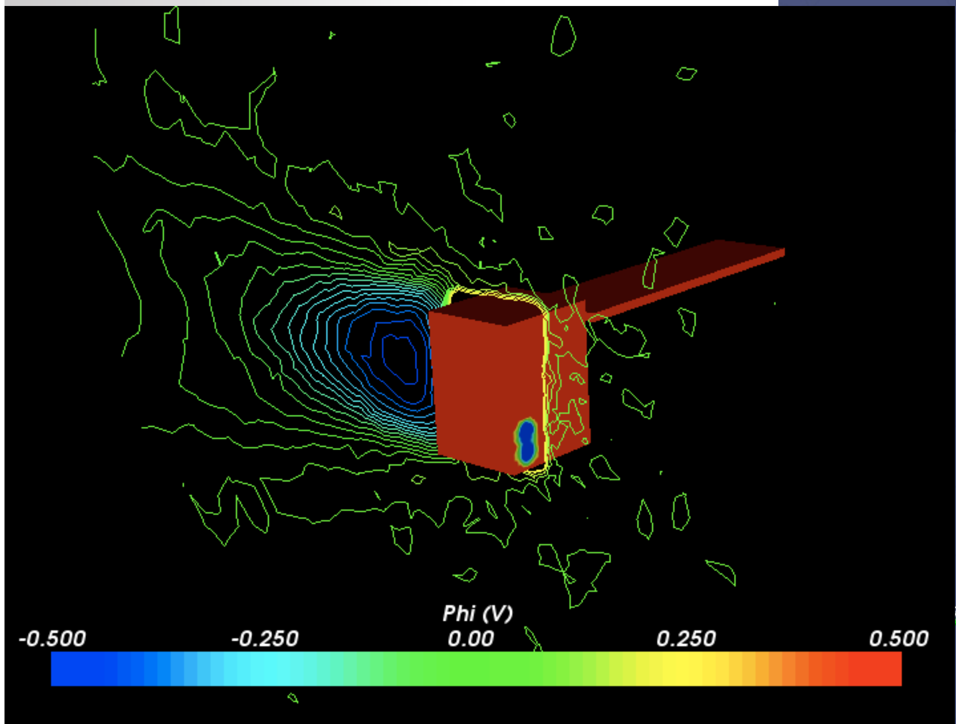
- Multi-data analysis and filtering

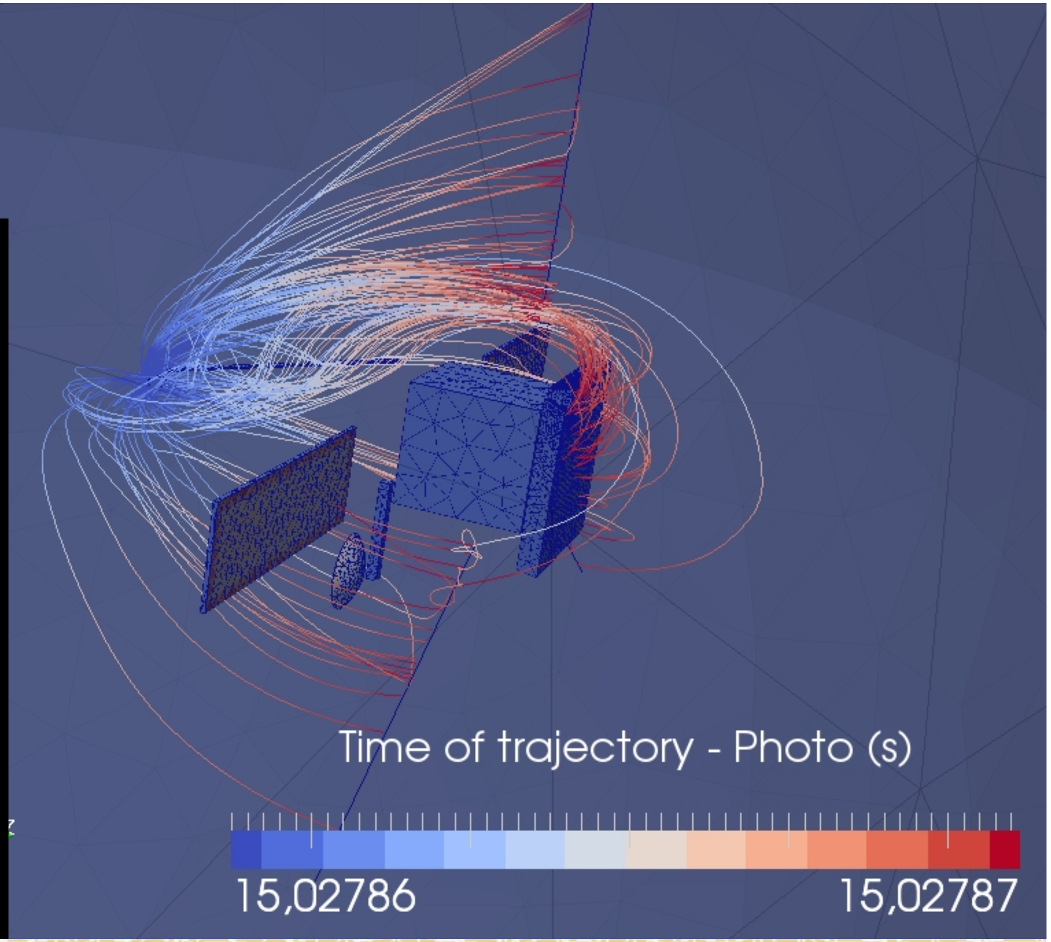


SPIS-UI (17/24)

User interface overview – Data mining

- Tailored post-processing
  - Particles tracking
  - Probing
  - Pre-built analysis pipelines



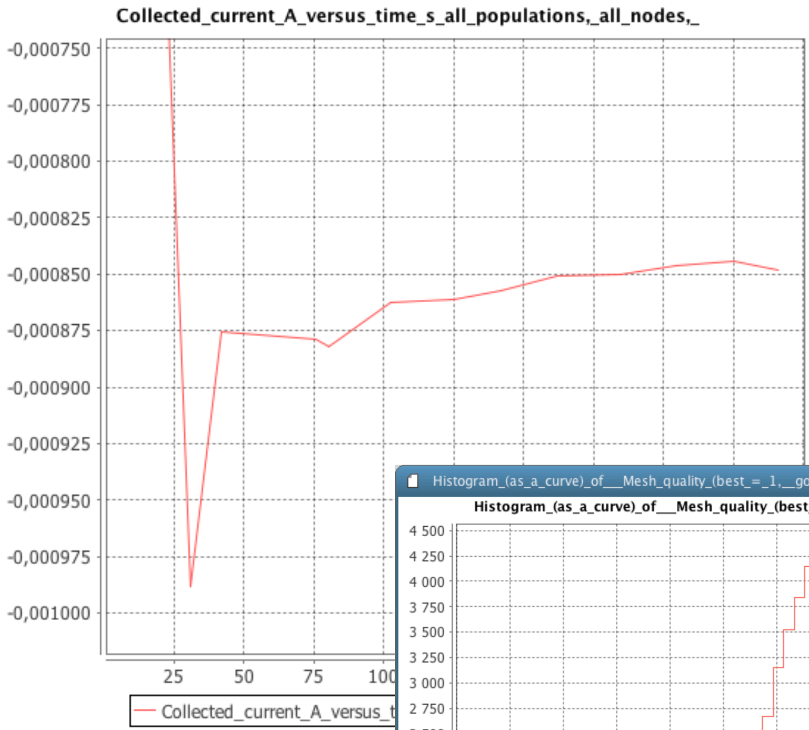


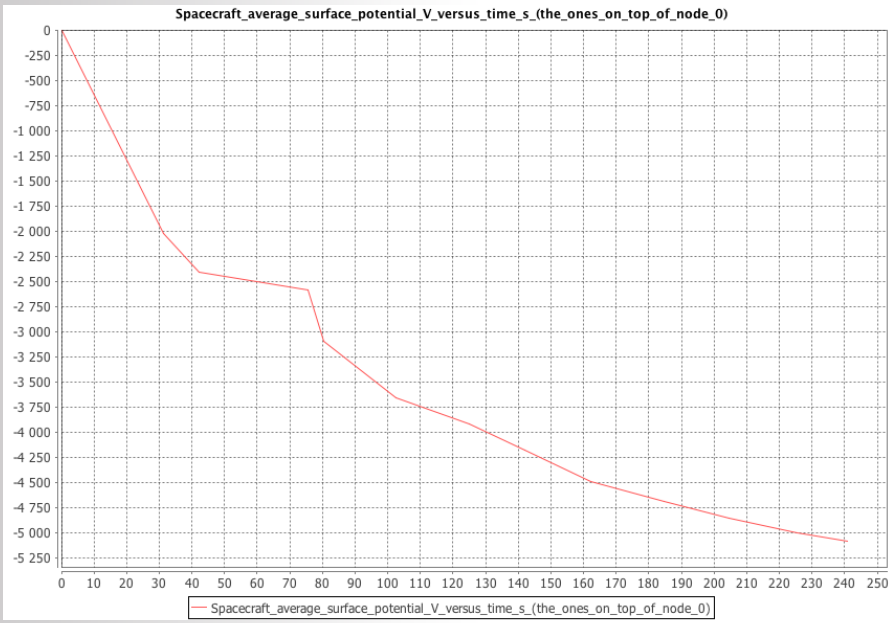
Artenum, Paris
Science & Groupware

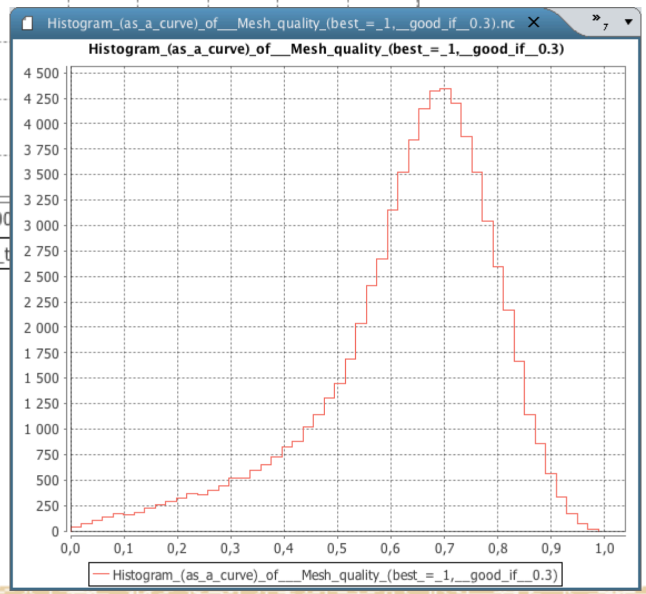
SPIS-UI (18/24)

User interface overview – Data mining

- Rich 2D post-processing capabilities, based on JFreeChart 2D plotting functions
  - times series
  - histograms



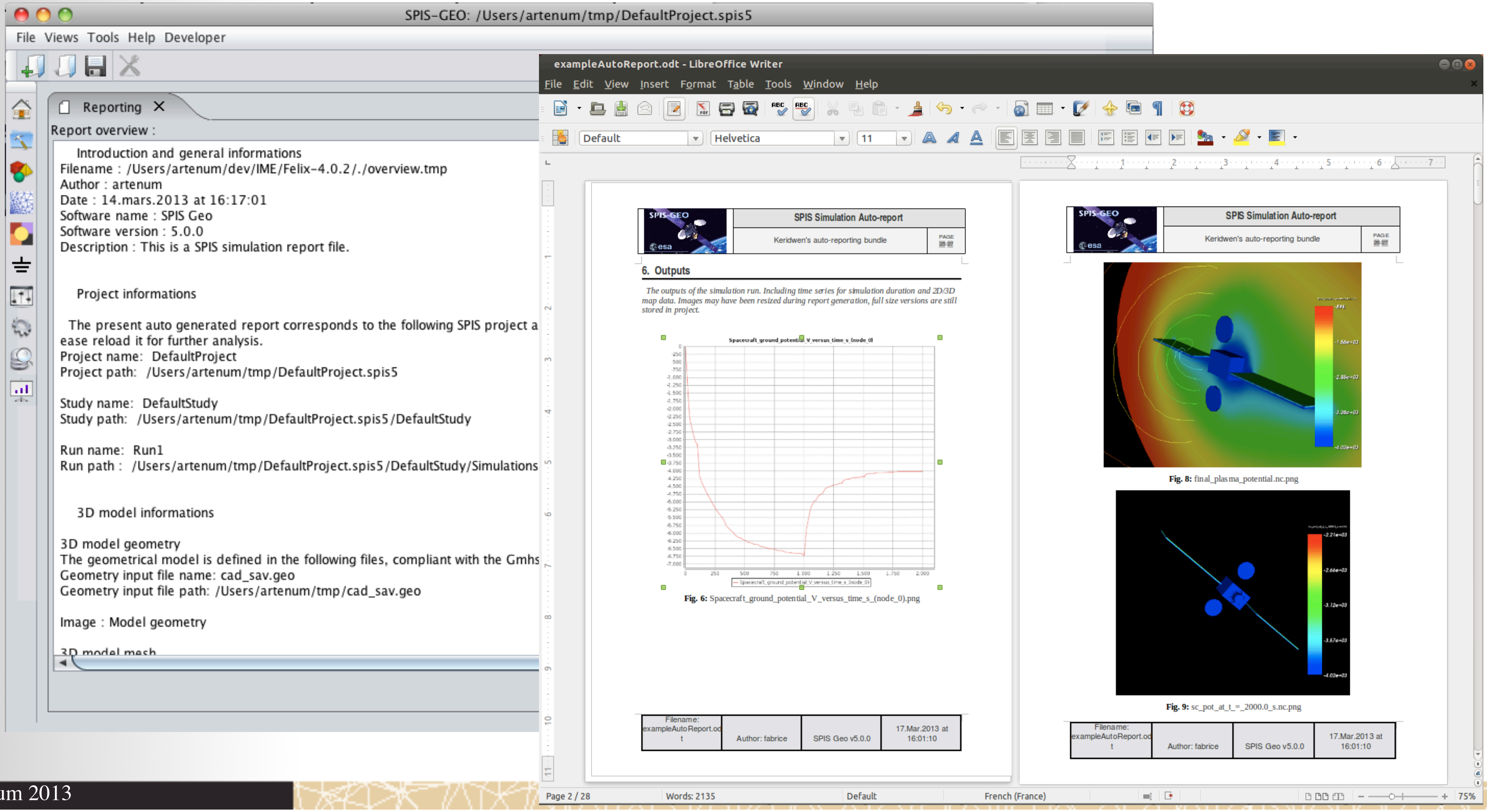




SPIS-UI (19/24)

User interface overview – Reporting

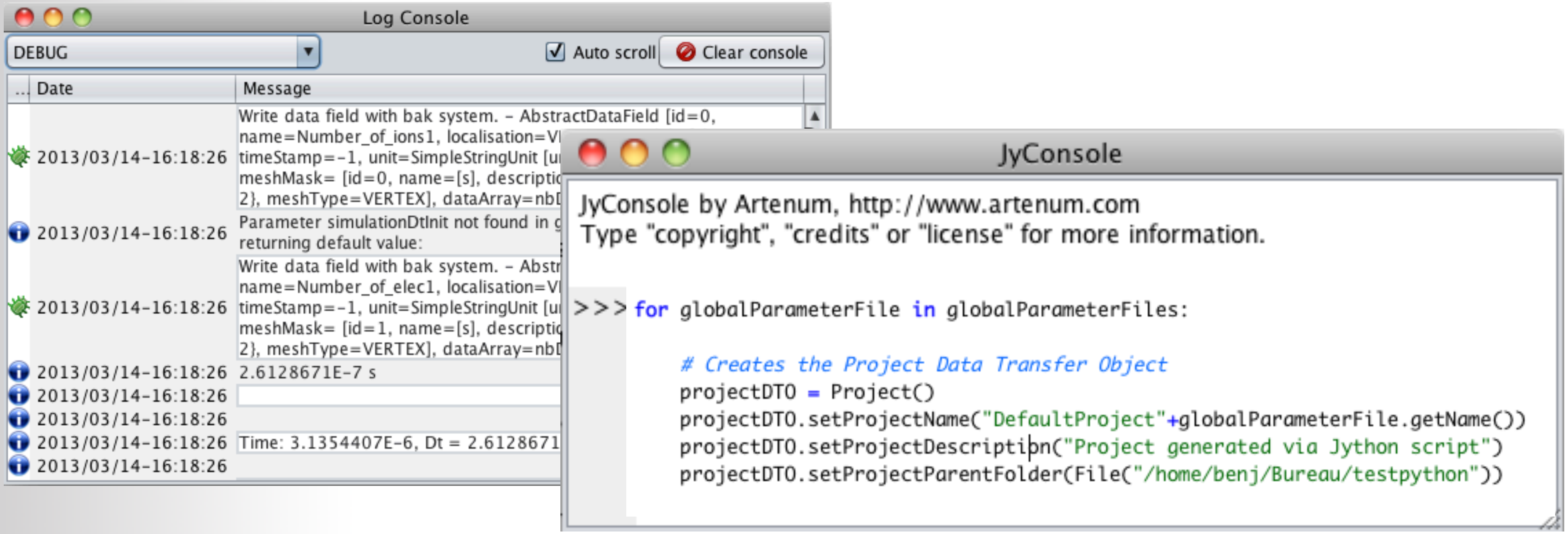
- New reporting feature: automatically generated OpenOffice reports



SPIS-UI (20/24)

User interface overview – Log console and scripting

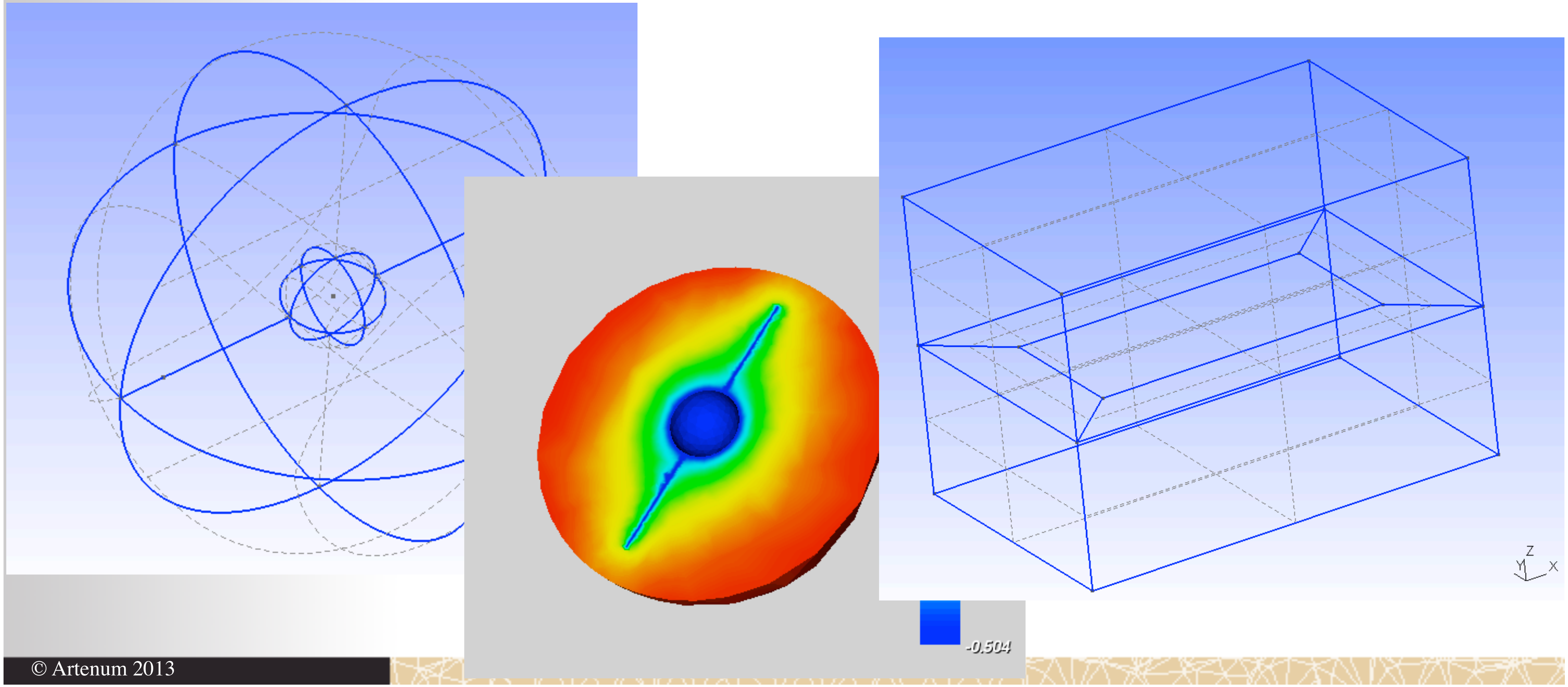
- New log console displays shows detailed information facilitating maintenance of the software and bug fixing
- SPIS 5 is still fully controllable with Jython scripts
- Batch mode allows to run SPIS in headless mode (i.e. without its graphical user interface
- There are scripts provided to perform parametric studies (e.g. variation of global parameters or material characteristics)



SPIS-UI (21/24)

User interface overview – Advanced functions

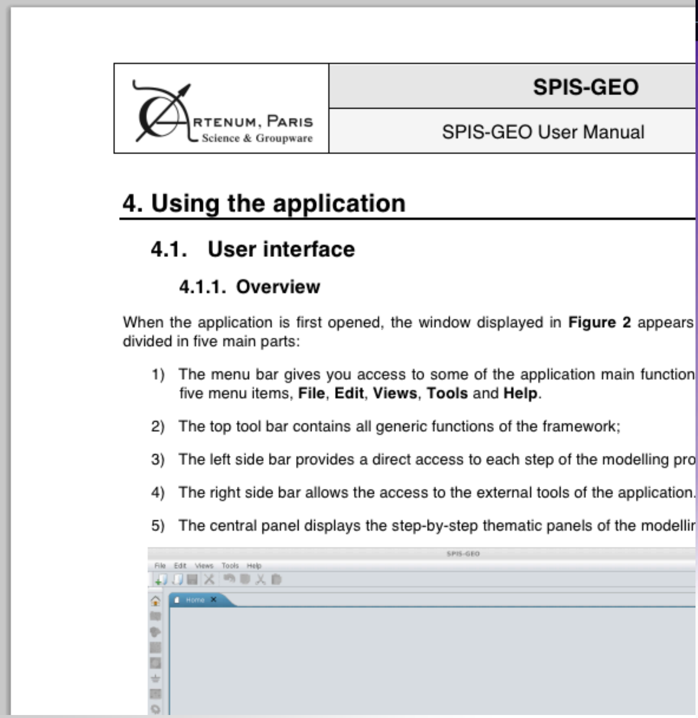
- Modelling of thin elements
  - 1D thin elements: wires, booms, antennas...
  - 2D thin elements: solar arrays, sails, antennas...



SPIS-UI (22/24)

User interface overview – Documentation

- Direct access to the user documentation from the user interface
- Comprehensive documentation (more than 400 pages) for both the user interface and the numerical kernel
- The user manual describes the features in detail (70 pages): read it!



ARTENUM, PARIS Science & Groupware	SPIS-GEO
	SPIS-GEO User Manual

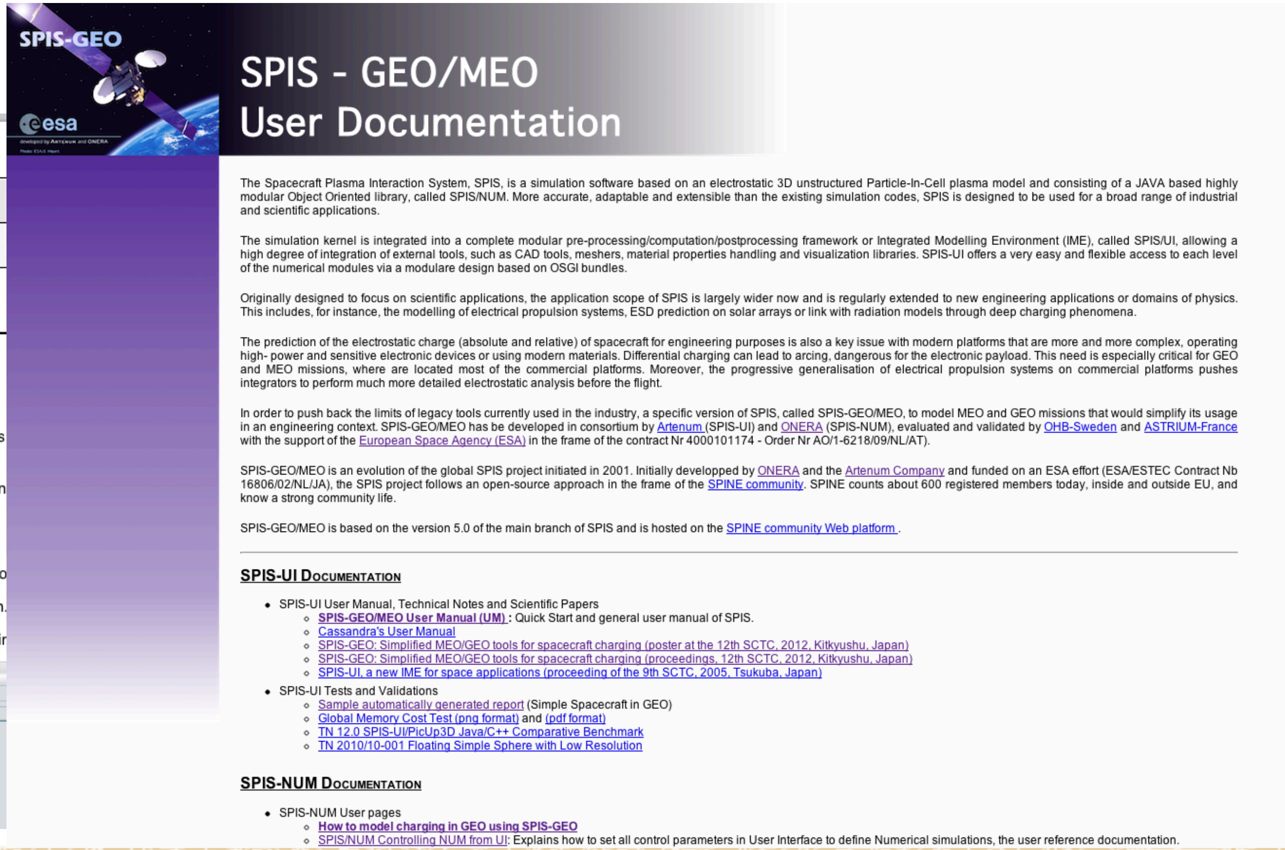
4. Using the application

4.1. User interface

4.1.1. Overview

When the application is first opened, the window displayed in **Figure 2** appears divided in five main parts:

1) The menu bar gives you access to some of the application main function five menu items, **File**, **Edit**, **Views**, **Tools** and **Help**.
2) The top tool bar contains all generic functions of the framework;
3) The left side bar provides a direct access to each step of the modelling pro
4) The right side bar allows the access to the external tools of the application.
5) The central panel displays the step-by-step thematic panels of the modellir



SPIS - GEO/MEO User Documentation

The Spacecraft Plasma Interaction System, SPIS, is a simulation software based on an electrostatic 3D unstructured Particle-In-Cell plasma model and consisting of a JAVA based highly modular Object Oriented library, called SPIS/NUM. More accurate, adaptable and extensible than the existing simulation codes, SPIS is designed to be used for a broad range of industrial and scientific applications.

The simulation kernel is integrated into a complete modular pre-processing/computation/postprocessing framework or Integrated Modelling Environment (IME), called SPIS/UI, allowing a high degree of integration of external tools, such as CAD tools, meshers, material properties handling and visualization libraries. SPIS-UI offers a very easy and flexible access to each level of the numerical modules via a modulare design based on OSGI bundles.

Originally designed to focus on scientific applications, the application scope of SPIS is largely wider now and is regularly extended to new engineering applications or domains of physics. This includes, for instance, the modelling of electrical propulsion systems, ESD prediction on solar arrays or link with radiation models through deep charging phenomena.

The prediction of the electrostatic charge (absolute and relative) of spacecraft for engineering purposes is also a key issue with modern platforms that are more and more complex, operating high- power and sensitive electronic devices or using modern materials. Differential charging can lead to arcing, dangerous for the electronic payload. This need is especially critical for GEO and MEO missions, where are located most of the commercial platforms. Moreover, the progressive generalisation of electrical propulsion systems on commercial platforms pushes integrators to perform much more detailed electrostatic analysis before the flight.

In order to push back the limits of legacy tools currently used in the industry, a specific version of SPIS, called SPIS-GEO/MEO, to model MEO and GEO missions that would simplify its usage in an engineering context. SPIS-GEO/MEO has be developed in consortium by Artenum (SPIS-UI) and ONERA (SPIS-NUM), evaluated and validated by OHB-Sweden and ASTRIUM-France with the support of the European Space Agency (ESA) in the frame of the contract Nr 4000101174 - Order Nr AO/1-6218/09/NL/AT).

SPIS-GEO/MEO is an evolution of the global SPIS project initiated in 2001. Initially developped by ONERA and the Artenum Company and funded on an ESA effort (ESA/ESTEC Contract Nb 16806/02/NL/JA), the SPIS project follows an open-source approach in the frame of the SPINE community. SPINE counts about 600 registered members today, inside and outside EU, and know a strong community life.

SPIS-GEO/MEO is based on the version 5.0 of the main branch of SPIS and is hosted on the SPINE community Web platform .

SPIS-UI Documentation

- SPIS-UI User Manual, Technical Notes and Scientific Papers
  - **SPIS-GEO/MEO User Manual (UM)** : Quick Start and general user manual of SPIS.
  - Cassandra's User Manual
  - SPIS-GEO: Simplified MEO/GEO tools for spacecraft charging (poster at the 12th SCTC, 2012, Kitkyushu, Japan)
  - SPIS-GEO: Simplified MEO/GEO tools for spacecraft charging (proceedings, 12th SCTC, 2012, Kitkyushu, Japan)
  - SPIS-UI, a new IME for space applications (proceeding of the 9th SCTC, 2005, Tsukuba, Japan)
- SPIS-UI Tests and Validations
  - Sample automatically generated report (Simple Spacecraft in GEO)
  - Global Memory Cost Test (png format) and (pdf format)
  - TN 12.0 SPIS-UI/PicUp3D Java/C++ Comparative Benchmark
  - TN 2010/10-001 Floating Simple Sphere with Low Resolution

SPIS-NUM Documentation

- SPIS-NUM User pages
  - **How to model charging in GEO using SPIS-GEO**
  - SPIS/NUM Controlling NUM from UI: Explains how to set all control parameters in User Interface to define Numerical simulations, the user reference documentation.

SPIS-UI (23/24)

User interface overview – Project structure

- SPIS projects are saved in a hierarchical file structure
- Significant performance improvement in loading/saving of projects
- Extensive use of standard file formats (XML, NetCDF, etc.)

```
DefaultProject.spis5                 ----> Root directory
 ├──── model.xml
 └──── DefaultStudy                  ----> Study directory
        ├──── Preprocessing          ----> Preprocessing directory for the study
        │      ├──── Datafields       ----> Pre-processing DataFields
        │      ├──── ElectricalCircuit ----> Internal circuit settings directory
        │      ├──── Geometry         ----> Geometrical models directory
        │      ├──── Groups           ----> Groups and properties settings dirertory
        │      ├──── Mesh             ----> Mesh directory
        │      └──── Properties       ----> Properties directory
        │
        └──── Simulations            ----> Simulation directory
               └──── Run1            ----> Run directory
                      ├──── GlobalParameters ----> Global parameters directory
                      ├──── NumKernel        ----> Spis-Num specific directory
                      └──── OutputFolder     ----> Output data directory
```

SPIS-UI (24/24)

Improvements summary

- Simplified user interface that lowers the learning curve
- More homogeneous interface and integration of third party components
- Better traceability and simplified release mechanism
- Improved overall code quality and maintainability
- The use of industry-wide standards (Java, Maven, OSGi, etc.) facilitates community contributions
- Significant performance improvements (e.g. project loading/ saving)
- True modularity that greatly improves SPIS evolutivity
- Will benefit to all future versions of SPIS (SPIS-Science, SPIS-Dust, AISEPS, etc.)

Adapted numerical models

Presentation by ONERA

Artenum, Paris
Science & Groupware

Validation campaign

Presentation by OHB-SE

Conclusion & perspectives

- Transition to SPIS 5 is a significant improvement for SPIS-GEO and all future versions of SPIS
- Keridwen 2 offers a modern architecture to SPIS and paves the way to better interoperability with other space environment tools
- Simplified and more homogeneous user interface reduces the learning curve for new users
- Improved numerical models are more adapted to industrial users needs
- Validation campaign lead by end users and not by developers
- Real-life comparisons show promising results
- Further analyses should be done

- SPIS-GEO is released and available on SPIS website http://www.spis.org
- Beginning of the 6-months maintenance phase: feedback is welcome!

Conclusion & perspectives

Perspectives

- Regarding the SPIS-UI IME:
  - Good example of application of Keridwen as tailored IME
  - Good basis for interoperability with other modelling tools
  - Good basis for other models/domains of physics
  - Ready for a future distributed architecture
  - Possibility to integrate collaborative tools

- Regarding SPIS-NUM:
  - Improved models able to address industrial needs
  - Performance optimisation