PROVEX – APPROACHABLE DATA PROVENANCE TRACKING FOR SPACE MISSIONS

H.Steinlechner[1], S. Pichler[*], T. Kohout[2,3], D. Korda[2], T. Ortner[4], C. Traxler[*], and G. Paar[5]

[1] *VRVis Zentrum für Virtual Reality und Visualisierung Forschungs GmbH, Vienna, Austria, hs@vrvis.at*
[2] *Department of Geosciences and Geography, University of Helsinki, Helsinki, Finland, tomas.kohout@helsinki.fi*
[3] *Institute of Geology of the Czech Academy of Sciences, Prague, Czech Republic*
[4] *thomas@ortner.fyi*
[5] *JOANNEUM RESEARCH Forschungsgesellschaft mbH, Graz, Austria, gerhard.paar@joanneum.at*

ABSTRACT

Data understanding is crucial throughout space missions and inherent in science workflows. To understand and reproduce scientific results it is important to trace them back to the data-sources and derived products at all times. With heterogeneous data-sources, large data-sets, complex data acquisition and processing methods, however, it is challenging to keep track of heterogeneous instrument data and complex derived products such as 3D reconstructions. We present *PROVEX*, a provenance tracking framework which provides a principled approach to collaborative data and workflow management targeting space science applications. *PROVEX* provides the data and workflows as graph visualizations that the user can interact with. We showcase the application with use-cases from geological interpretation based on 3D reconstructions and mineral analysis using spatially mapped multi-spectral imagery. Testing emphasizes use cases from preparations for the Hera mission [4].

Key words: Provenance, Notebooks, Hera, 3D-GIS, Visualization.

1. MOTIVATION

Large-scale heterogeneous instrument data from cameras, spectrometers, and other chemical and physical instruments is immanent to space missions and plays a central role in space science. Analysis such as geological interpretation crucially depends on the understanding and consistency of the data products.

Given the central role of measurements and processing it is important to trace back the relevant data when drawing conclusions. For example, when integrating new instrument, calibration data or improved data-processing methods it is important to identify which findings from derived products need to be re-evaluated. The information needed to trace back each computation, interaction or analysis step is often referred to as *provenance information* and needs to be taken care of as an integral component of the data pipeline and analysis workflow.

Tracking provenance data in data-centered approaches is not new, yet it is not easily applicable to the particular setting of space missions for two reasons: (1) dealing with large 3D reconstructions and heterogeneous instrument data is challenging, (2) provenance data alone is not enough – provenance data needs to be tracked and documented through all phases of data acquisition and -analysis and be *approachable* and *repeatable* for scientists. In this work we present the PROVEX framework, which allows to coherently track provenance data in space missions. The approach is outlined in Fig. 1.

2. BACKGROUND

PROVEX focuses on managing planetary 3D reconstructions [7] as well as instrument data and maintaining scientific results in a tractable and approachable manner. The data in this context is generated and provided by *PRoViP* (Planetary Robotics Vision Processing) [8]. For visualizing 3D data products, the *PRo3D* [1] (Planetary Robotics 3D Viewer) is used by *PROVEX*. *PRo3D* is a multi-platform open source tool maintained under GitHub [6]. *PROVEX* integrates a viewer to visualize the *PRoViP* data and *PRo3D* to visualize and interact with the 3D data products in order to inspect the tracked data in an approachable way and to perform new analysis.

We demonstrate PROVEX for the following scientific use cases, in preparation of Hera [4] science operations and in discussion with the Hera Working Group 4 (WG4) *"Data Analysis, Exploitation, Interpretation"*: (1) Geological interpretations of planetary 3D reconstructions and (2) Qualitative and quantitative analysis of mineral compositions (chemical material compound analysis using multi-spectral image data) of astronomical objects. All images and analyses of Dimorphos are based on the Dimorphos shape model g_01960mm_spc_obj_dimo_0000n00000_v003.obj from the *DART repository*.

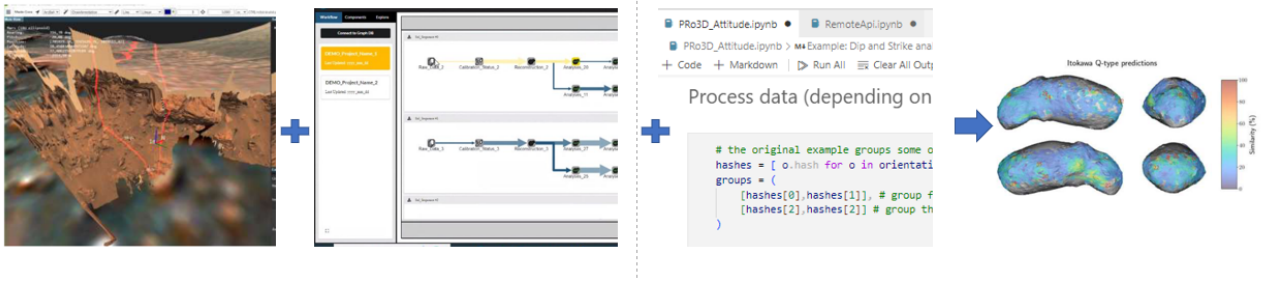


Figure 1: *PROVEX* integrates 3D data (left) with provenance tracking in a provenance graph (middle left) interoperating with Jupyter notebooks for scientific analysis (middle right) and tracks the generated output (right) within the provenance graph [3]. Processing, analysis and interpretation steps can be analyzed at all times, processing parameters and key interactions can be inspected, adjusted and coherently tracked in the system.

3. PROVEX TECHNICAL REALIZATION

PROVEX is being developed to track analyses workflows for Hera. It is a web-based application that establishes a connection to a graph database that stores manually and automatically tracked data to reproduce and visualize previous workflow states. The data types include images and corresponding meta data files that contain information about the underlying data that was used to reconstruct Dimorphos (see Fig.10), PRo3D files to display and interact with the reconstructed model and varying documentation file formats to present the findings.

The provenance graph and the files corresponding to graph nodes are displayed in the web application. The graph visualizes how files relate to each other and additionally represents the chronology of processing steps encoded in the graph layout.

The following subsequent sections discuss PROVEX in more detail. Section 3.1 presents the PROVEX architecture, the product types that are displayed in the graph and features that are implemented in PROVEX. The web application is summarized in Section 3.2. Section 3.3 addresses how to interact with PROVEX within PRo3D, and Section 3.4 covers the integration of PROVEX in concrete data-science workflows.

3.1. Architecture

The PROVEX architecture consists of a graph database, a web application to interact with the data and the data-science platform Jupyter notebooks [2]. An overview of the PROVEX components and how they are connected is given in Fig. 2.

Images and calibration data are processed by the meta data engine based on PRoViP-exported and complementary imported data, and the resulting meta data and the 3D reconstruction are ingested into the PROVEX database (DB). PROVEX uses Neo4j [5] to store the graph data. The initial data upload can be done in a separate user interface that iterates over a given folder structure and uploads the images and camera data to a DB by providing the address of the server that interacts with the DB as depicted in Fig. 3.

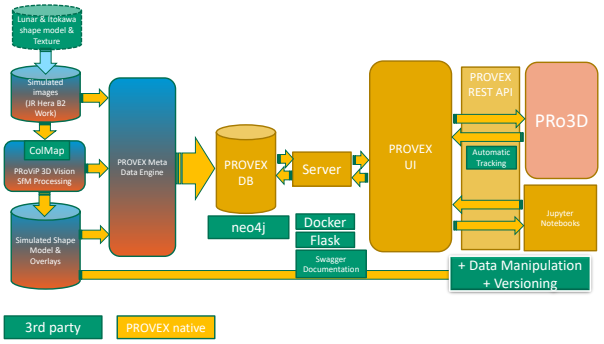


Figure 2: An overview of the technical components: All data products tracked by the *PROVEX Meta Data Engine* are stored in the *PROVEX database (DB)*. The server retrieves and adds new data. The *PROVEX user interface (UI)* visualizes the tracked data and establishes a connection with PRo3D to load and store PRo3D data from, respectively to the PROVEX DB. The jupyter notebooks can either communicate with the PROVEX DB using the PROVEX UI interface or directly the server interface. Native *PROVEX* components are marked in yellow.

The PROVEX user interface (UI), is discussed in more detail in Section 3.2, interacts with the PROVEX DB via a flask server. The server provides REST API endpoints to request graph data, add new nodes and edges and to update edge data. The server can be set up locally or the user can alternatively connect to the PROVEX DB. Therefore the user has to connect to the DB using the PROVEX UI, shown in Fig. 4.

The PROVEX UI also communicates with PRo3D which provides REST API endpoints to set a scene, to get scene data and to extract and apply annotations. PROVEX UI and PRo3D additionally communicate over a websocket connection, such that PRo3D can actively broadcast updates done by the user. These updates are automatically

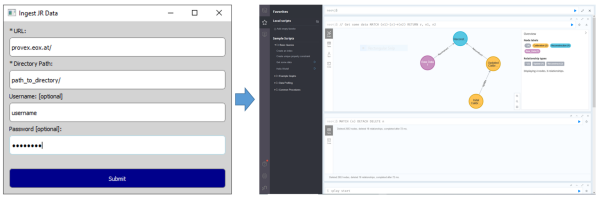


Figure 3: The data for the 3D reconstruction is uploaded with the data injestion user interface (left) to Neo4j (right).

Figure 4: EOX connection requires credentials (top). Graph data is displayed after a successful connection is established (bottom).

or semi-automatically sent to PROVEX to be stored to the PROVEX DB. This is discussed in more detail in Section 3.3.

For more complex analysis workflows we intergrate Jupyter Notebooks. The results can either directly be uploaded using the flask server API or manually using the PROVEX UI.

3.2. PROVEX web application

Fig. 5 shows the layout of the web application:

1. The editor panel shows the content of the selected node

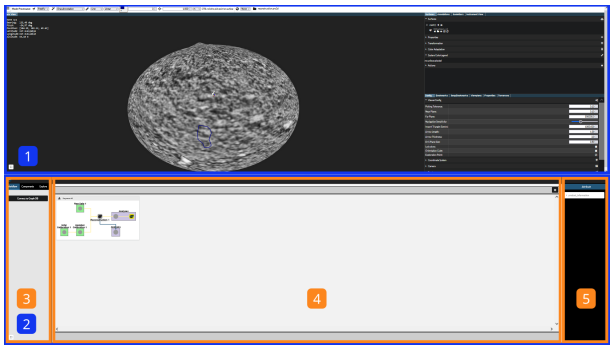


Figure 5: The PROVEX user interface with (1) the editor panel, (2) the graph panel, (3) the graph interaction panel, (4) the graph viewer panel and (5) the attribute panel

2. The graph panel depicts the graph and embedded panels to interact with the graph and to analyse the graph
3. The embedded graph interaction panel implements the tabs *Projects*, *Scripts*, *Explore* to load and select a specific project, to apply scripts and to explore the graph.
4. The embedded graph viewer panel contains tiles that show multiple workflows based on the project
5. The attribute panel shows meta data connected to the selected node

The graph includes item nodes and collection nodes. Collections function as semantic orientation for the user and support the user to focus on important aspects as collections can be collapsed and extended interactively.

We specified following item node types with the respective file types:

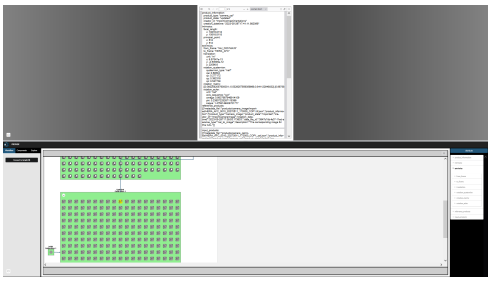
- Raw Data: Image
- Calibration: JSON
- Reconstruction: PRo3D
- Analyses: PRo3D
- Presentation: Document — 3D Object — Notebook

We further specified the following collection node types:

- Raw Data Collection
- Calibration Collection
- Analyses Collection
- Presentation Collection

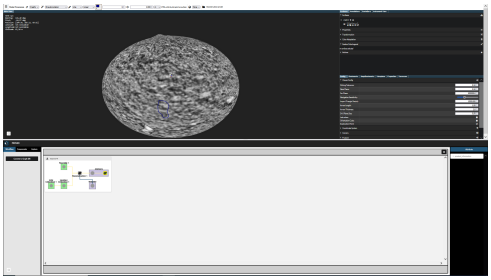
Raw Data nodes represent images that are used for the reconstruction, calibration nodes represent the calibration data in a json format, the reconstruction node stores the 3D mesh embedded in a PRo3D scene, analyses nodes are analyses based on the previous reconstruction performed in PRo3D and the presentation nodes are files of the type image, PDF or .ipynb (jupyter notebook). Presentations show the findings of an analyses. Reports can be automatically generated using the download icon in the graph viewer panel. The automatic report stores an image of the current graph and extracts some meta-information, e.g. the names of the contributors from the graph.

Fig. 6 shows four different node selections. In figure a) an item node with the type *Calibration* is selected. In figure b) a collection node with the type *Raw Data Collection* is selected. Figure c) shows the UI with a selected *Analyses* item node and figure d) a selected *Presentation Collection* node.

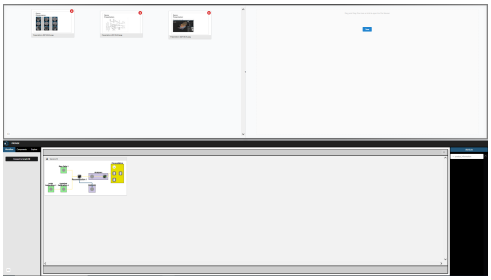


a) Selecting of a calibration item node

b) Selecting of a raw data collection node



c) Selecting of an analyses item node



d) Selecting of a presentation collection node

Figure 6: PROVEX UI with varying node selections.

3.3. Provenance tracking in PRo3D

Generally, a *PRo3D* scenes consists of reconstructed surfaces, possibly some measurements or annotations on the surface and visualization settings such as viewport, color settings etc. All this data can be saved as *PRo3D* project file. Generally, those files could simply be stored as a payload for provenance nodes. While this is a viable approach, we opted for a tighter integration of *PRo3D* scenes and *PROVEX*. To this end, we extended *PRo3D* with support for provenance tracking which comes in two flavors:

1. User-driven creation of provenance 'snapshots': A provenance snapshot is a state of the *PRo3D* scene which can be saved and restored at all times. Whenever the user made progress which should be stored for future analysis or exploration, the snapshot needs to be triggered manually. For each snapshot the system automatically maintains and updates the underlying provenance data.

2. Automatic provenance tracking which operates in the background and records interaction steps automatically. This approach is based on heuristics which detect unnecessary changes but creates a snapshot for potentially relevant user-interactions. As an example, consider changes of the viewport in the scene. As long as no annotations are created, capturing and storing all those exploratory interactions is unnecessary. To prevent storing those, subsequent redundant interactions collapse to a single node avoiding clutter. For explicitly intended snapshots, users can always fall back to manual snapshots. The creation of the snapshot and maintaining the provenance data works just like (1).

From a user perspective, automatic tracking can be enabled and disabled and also in automatic mode it is possible to trigger manual snapshots actively. An example provenance graph for a simple scene with one single polygon annotation is given in Fig. 7.

In order to distinguish between coarse-grained provenance information and fine-grained provenance on a per-scene level in the *PROVEX* UI, we introduced another node type called *collapsable node* which contains a *PRo3D* scene and its provenance graph as a sub-graph. This allows to collapse *PRo3D* nodes and expand them as needed when looking at the scene in more depth.

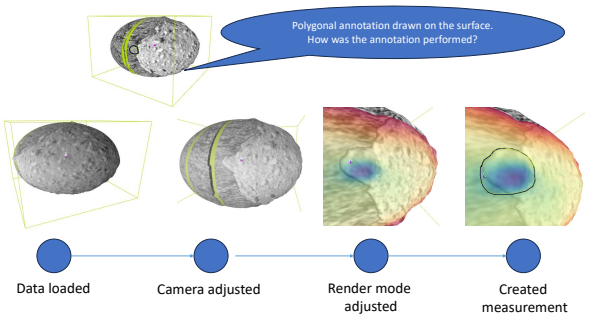


Figure 7: The provenance information for a polygonal measurement on the Dimorphos model which was recorded automatically. Nodes represent relevant modifications of the viewport, changes of visualization properties (e.g. use elevation coloring) and measurement interactions. The graph also reveals that the measurement was performed using the elevation map. By clicking on nodes, each step can be reproduced at all times. All annotation elements available in *PRo3D* get tracked and the provenance information is stored to the database transparently to the user.

3.4. Support for data-science workflows

While *PRo3D* provides workflows for geological analysis on 3D surfaces, numerous other tools exist for particular use cases. For this reason, in *PROVEX* we provide a technology independent interface for third-party tools and programming environment.

To this end, *PROVEX* exposes a web API which can be used from most programming environments and thus can be integrated in many tools. The API is provided as a `REST` service and provides functions such as:

- Load node data such as 3D reconstructions
- Query node data
- Query 3D annotation data
- Spatial and Geospatial queries on 3D reconstructions

This allows to access PROVEX data fluently from the outside which is particularly interesting when heterogeneous data (e.g. spatial, geospatial and raw instrument data) needs to be aggregated or combined.

PROVEX also provides Python wrappers for the REST interface to conveniently work from within common python environments, even from Jupyter notebooks [2]. An interactive session showing the interactive exploration of 3D data in the *PRo3D* panel and a Jupyter notebook session which analyses and visualizes the data interactively is depicted in Fig. 8.

Another use case comes into play when different methods need to be compared systematically and evaluated on data

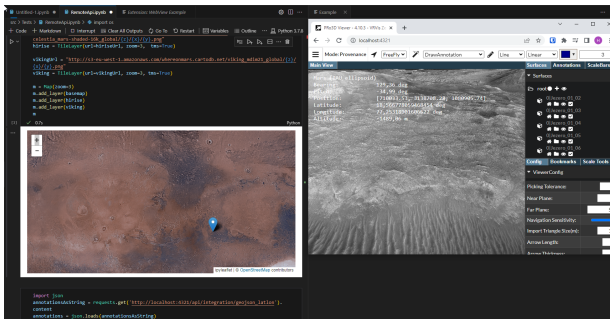


Figure 8: A Jupyter notebook session which shows the communication with the *PROVEX* API which processes 3D annotations being made in an interactive *PRo3D* session. The python *ipyleaflet* [11] package is used to visualize the annotations in 2D maps. Note that all data is synchronized automatically.

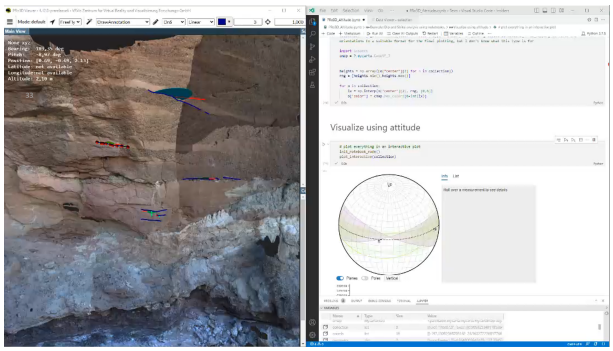


Figure 9: Geological interpretation and Jupyter integration of Dimorphos reconstruction (right).

stored in the PROVEX ecosystem. By using a Jupyter notebook which queries the *PROVEX* data, runs experiments and creates plots and graphs, evaluations and their conclusions can be made reproducible. The notebook itself can be stored as a *presentation node* in the provex system itself.

4. SHOW CASES AND MOTIVATIONAL APPLICATIONS

We evaluated our system using two motivating examples. In section 4.1 we show how notebooks can be used to compare and evaluate algorithmical methods. In section 4.2 we show how spatial and abstract queries can be used to extract surface properties, find interesting areas and export the results to other tools.

4.1. Notebooks for comparing strike and dip measurements

As an example for using *PROVEX* and its notebook interface for the evaluation of different methods solving particular problems, consider the analysis of geological

strike and dip measurements used for planar geologic features. For actually performing *strike and dip* measurements on 3D reconstructions plane orientations need to be computed [10]. While *PRo3D* has built-in support for this task, it is limited to simple *strike and tip* measurements without statistical quantification or other advanced use-case specific use-cases. Without *PROVEX*, for quantitative analyses for example, it is necessary to rely on import and export functionalities. More tightly integrated, by using the notebook library, different methods for computing *strike and dip* planes can be compared and plotted for example using the *attitude* library [9].

The results of such comparisons can be seen in Fig. 9. By storing the computational notebook in the *PROVEX* database, the evaluations can be repeated and comprehended at all times.

4.2. Provenance for complex spatial and abstract data queries

While 3D reconstructions typically come with an albedo map, for scientific use-cases additional data layers come into play, for example:

- Elevation
- Gravitational forces
- The reconstruction accuracy
- Instrument data such as hyper-spectral image data
- Chemical properties or material compositions

In order to support additional data-layers, in *PROVEX* we extended *PRo3D* with support for multiple texture layers. Example renderings of the Dimorphos 3D reconstruction can be seen in Fig. 10.

This feature becomes particularly interesting, when different mission phases, or particular analysis and measurements results need to be compared and versioning of data products with accumulating analysis and interpretations comes into place. The ‘copy Annotations’ in *PROVEX* is designed to support exactly those scenarios. Suppose that annotations and measurements have been performed on a reconstruction of an early stage in the mission. Later, when new reconstructions are available it might be interesting to map old annotations onto newer reconstructions. By clicking ‘copy annotations‘ on a *PRo3D* node containing old reconstructions and annotations in the provenance graph and pasting them onto another reconstruction, an implicit dependency is added to the provenance graph and both results appear simultaneously in the same scene while retaining the provenance information. The user can then proceed with modifying the annotations from either initial status, and their outcome can be followed back to their joint source. An example is given in Fig. 11.

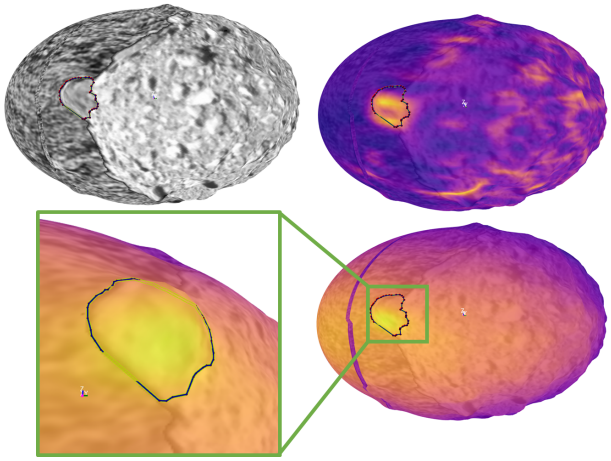


Figure 10: Rendering of the Dimorphos 3D reconstruction in *PRo3D* (top left), the slopes visualized are using a color scheme.

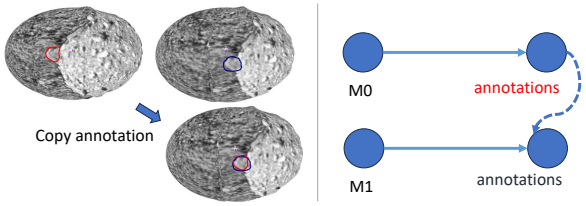


Figure 11: By using the ‘copy Annotation‘ feature, annotations performed on an older dataset can be added to a new reconstruction for comparison (left). The operation is also reflected in the provenance graph (right).

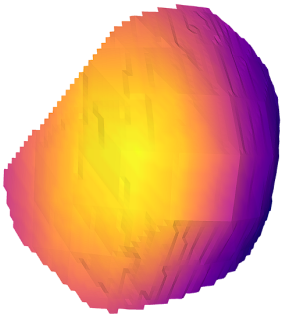


Figure 12: The geometry extracted as wavefront .obj file from the polygon measurement in Fig. 10

In order to support scientists with analysis of particular spatial regions on 3D reconstructions, the *PROVEX* API provides functions for extracting 3D regions from the reconstruction. An example cutout of a measurement on Dimorphos (see Fig. 10) is shown in Fig. 12.

The API has functions for querying, as for example:

- Geospatial bounding boxes
- 3D polygons projected onto the surface
- All vertices within a range, provided by any data layer (e.g. elevation).

The queries can be invoked directly from the Jupyter notebook environment to perform further computation.

Such functionality is useful when source code of the notebook is saved as presentation nodes in the provenance graph. Given new reconstructions in later phases of the mission, the nodes containing the notebook can be loaded again and the analysis can be run on a different reconstructions which allows to quantitatively compare analysis results. Taking this idea even further, figures and plots for publications can be regenerated whenever new data becomes available.

5. CONCLUSIONS AND OUTLOOK

In this work we presented a systematic approach to maintain provenance information for input and derived products and extended *PRo3D* with support for fine-grained provenance tracking. Based on a common API, it is possible to interact with analysis results programmatically using the Jupyter notebook environment for quantitative analysis of geospatial and abstract instrument data. The *PROVEX* approach will be followed-up in forthcoming discussions with the Hera WG4 Team in terms of viable scientific use cases and their provenance support. Important future research directions are the comparison of data from different mission phases, the investigation of more complex geospatial queries for the Hera mission, and the collaboration aspect.

ACKNOWLEDGEMENTS

JR and VRVis PROVEX development and Hera participation is funded by ESA Contracts `4000138386/22/NL/GLC/my` and `4000141262/23/NL/GLC/my`. VRVis is funded by BMK, BMDW, Styria, SFG, Tyrol and Vienna Business Agency in the scope of COMET - Competence Centers for Excellent Technologies (879730) managed by FFG. David Korda and Tomas Kohout are supported by Academy of Finland ICT project no. 335595 and within institutional support RVO 67985831 of the Institute of Geology of the Czech Academy of Sciences. We thank the Hera WG4 Team Members and Hera ESA Project Scientist Michael Kueppers for their ongoing support, as well as the DART and DRACO Teams for provision of the Dimorphos shape model and DART impact images.

REFERENCES

[1] R. Barnes, S. Gupta, C. Traxler, T. Ortner, A. Bauer, G. Hesina, G. Paar, B. Huber, K. Juhart, L. Fritz, B. Nauschnegg, J.-P. Muller, and Y. Tao. Geological Analysis of Martian Rover-Derived Digital Outcrop Models Using the 3-D Visualization Tool, Planetary Robotics 3-D Viewer—PRo3D. *Earth and Space Science*, 5(7):285–307, 2018.

[2] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing. Jupyter Notebooks – a publishing format for reproducible computational workflows. In F. Loizides and B. Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87 – 90. IOS Press, 2016.

[3] D. Korda, T. Kohout, K. Flanderová, J.-B. Vincent, and A. Penttilä. (433) Eros and (25143) Itokawa surface properties from reflectance spectra. 675:A50, jul 2023.

[4] P. Michel, M. Küppers, A. C. Bagatin, B. Carry, S. Charnoz, J. De Leon, A. Fitzsimmons, P. Gordo, S. F. Green, A. Hérique, et al. The ESA Hera mission: detailed characterization of the DART impact outcome and of the binary asteroid (65803) Didymos. *The planetary science journal*, 3(7):160, 2022.

[5] Neo4j. Neo4j - The World's Leading Graph Database, 2012.

[6] T. Ortner. PRo3D - Planetary Robotics 3D Viewer, 2023.

[7] G. Paar, T. Ortner, C. Tate, R. G. Deen, P. Abercrombie, M. Vona, J. Proton, A. Bechtold, F. Calef, R. Barnes, et al. Three-Dimensional Data Preparation and Immersive Mission-Spanning Visualization and Analysis of Mars 2020 Mastcam-Z Stereo Image Sequences. *Earth and Space Science*, 10(3):e2022EA002532, 2023.

[8] G. Paar, T. Ortner, C. Traxler, R. Barnes, M. Balme, C. Schröder, and S. G. Banham. Preparing 3D vision & visualization for ExoMars. In *Proceedings of 16th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA 2022)*. ESA/ESTEC, 2022.

[9] D. P. Quinn. The attitude package. `https://pypi.org/project/Attitude/`, 2019. [Online; accessed 05-September-2023].

[10] D. P. Quinn and B. L. Ehlmann. A PCA-Based Framework for Determining Remotely Sensed Geological Surface Orientations and Their Statistical Quality. *Earth and Space Science*, 6(8):1378–1408, 2019.

[11] Q. Wu. Leafmap: A Python package for interactive mapping and geospatial analysis with minimal coding in a Jupyter environment. *Journal of Open Source Software*, 6(63):3414, 2021.