

HANDLING PDS4 EVOLUTION IN THE PSA. J. Saiz¹, I. Barbarisi¹, R. Docasal¹, C. Rios¹, A. Montero¹, C. Laantee¹, J. Osinde¹, J. Ruano¹, S. Besse¹, J. Arenas¹, B. Merin¹, C. Arviset¹, ¹ESA/ESAC, Camino Bajo del Castillo s/n, Urb. Villafranca del Castillo, 28691 Villanueva de la Cañada, Spain, jsaiz@sciops.esa.int.

Introduction: The Planetary Science Archive (PSA) [1] is the central repository where products from all planetary missions of the European Space Agency (ESA) are stored, following the standards given by the Planetary Data System (PDS).

While legacy missions such as Giotto, Huygens, Venus Express and SMART-1, the Rosetta mission, currently in post-operations phase, and the still operational Mars Express, use the former PDS3 standard, newer missions like ExoMars 2016, ExoMars RSP, BepiColombo and Juice use or will use PDS4 [2].

PDS4 versioning: Adopting PDS4 as the standard for new missions while being compatible with previously existing PDS3 products in the same archive is managed with a common data model that matches the equivalent concepts of both standards.

Additionally, PDS4 follows its own evolution. Roughly every six months, a new version of the PDS4 standard is released by the PDS. This implies evaluating a number of options, both by the missions as well as by the archive:

- Which PDS4 version do we use?
- How often do we migrate to the last version?
- Do we upgrade all existing products?
- If each mission decides to use a different PDS4 version, how do we manage them in the PSA?

To facilitate this decision making to data providers by not imposing additional constraints, the PSA has built a flexible layout where various PDS4 versions are supported simultaneously: not only diverse information model versions between missions, but also different versions along the lifetime of a single mission.

PDS4 model Java API: In order to isolate the *bundle generator*, the *ingester* and *updater* modules of the PSA from the evolution of the PDS4 standard, version independent Java interfaces have been created to model the various PDS4 entities. These interfaces are implemented by adapter classes for every PDS4 information model version, which make the appropriate translations to the corresponding schemas, with a minimal overhead.

Although this common API could be slightly modified every time a new IM version is required to be incorporated, the impact on client code is much less drastic than by exposing the generated JAXB [3] classes directly. This approach leads to a reasonably stable and robust codebase for the PSA processes that handle PDS4 data of the supported missions.

References: [1] Besse, S. et al., (2017) *Planetary and Space Science*; [2] Data Design Working Group (2018) *PDS4 Concepts*; [3] E. Ort and B. Mehta (2003) *Java Architecture for XML Binding (JAXB)*.