ELSA REVEALED: PHILOSOPHY AND PURPOSE, STUDENT-LED DESIGN

L.D.V. Neakrase, T.D. Pagán, T. Arnold, L. Huber, N. Chanover
Planetary Data System, Atmospheres Node, Dept. of Astronomy, P.O. Box 30001, MSC 4500, New Mexico State University, Las Cruces, NM 88003-8001.

Introduction. In an effort to aid derived data and/or single-time data providers, the PDS Atmospheres Node (ATM) has been developing a new online tool, the Educational Labeling System at Atmospheres (ELSA). ESLA provides an online web environment to build PDS4 bundles for simple data while teaching users about PDS4 structures and organization.

Under the PDS4 archiving standard, datasets are organized by an Information-Model-driven hierarchy:

Bundle → Collection(s) → Basic Products

Every registered product consists of a metadata label and the product itself (data, document, inventory files, context products, etc.). The labels are stand-alone XML files that contain all the necessary information about the companion files, including linkages to related files. All products must have unique identifiers (logical identifiers – LIDs) that are used to register and reference every product under this system. Logical Identifiers are implemented through the use of Uniform Resource Names (URNs) and for NASA will have the form:

urn:nasa:pds:bundle_id:collection_id:product_id

There are formal and informal rules for constructing unique, valid URNs for PDS4. Although these rules are laid out in PDS documentation (e.g., [1] *PDS Data Preparers Handbook*, §2.5, p.6-7), some data providers have found this portion of the data archiving process to be confusing and cumbersome. To simplify and streamline this process, ELSA is designed to assist data providers in building PDS4-compliant metadata labels for their archives using an intuitive and educational web-based interface.

Philosophy and Purpose. More than just a tool for creating labels, ELSA employs a philosophy for working with data providers borne out of Atmospheres Node experience working with newer, less-experienced data providers, many of whom are new to the PDS and are tackling the archiving task for the first time. ATM is adapting our personal experience with the parts of the PDS4 standard into the ELSA system.

Because of the integrated nature of the URN-LID system in PDS4, it is important to build most of the associated labels all at once from the top down. This can sometimes be confusing for new PDS4 users. The hierarchical layout of a PDS4 bundle builds off of the LID initially setup for the bundle file, continuing down to the file/product level.

Alongside the internal referencing for the bundle hierarchy are context references that provide pointers to PDS-curated product descriptors for system-wide references for investigations, hosts (spacecraft and facilities), instruments, telescopes, and targets. Each context product has a unique LID that can sometimes be difficult to produce. The goal of ELSA is to provide an easy interface for selecting what a user wants and then correctly populating it in the system of bundle labels.

The desired end product of the ELSA application is a populated, cross-referenced system of bundle labels with hard-to-find, tough-to-remember references provided through a web interface to allow users to explore and become familiar with the PDS4 standard whilst constructing viable label ***templates*** for a complete PDS4 bundle. The goal is to have an easy way to edit and create files that can eventually be used for generating large numbers of labels through a data archive generation pipeline, or for making individual label files as needed.

Process and Mechanics. ELSA at ATM has been a student-driven process and its development is a testament to the hard work and devotion of the ATM undergraduate workers that have developed the software from the ground up. Initially started as a JavaScript and PHP project, student involvement led the project to adopt a Python/Django framework that has proven to be easier for the students to learn and develop new modules. ELSA development has moved into object-oriented design to increase the ability to build modularly, which fits with an expanding architecture and the students' busy academic schedules. Object-oriented design also allows more stability and control for editing and updating the software.

Design principles employed by ATM for the initial design of ELSA included:

- *Educational:* include instructive text to guide users and provide links to further resources.
- *Plug and Play:* allow a more tactile approach to input, allowing users to input values, select from drop-down menus, etc.
- *Uniformity:* include a uniform look and feel across all aspects of the application

with rules set in place for how new sections should look.
- *Modularity:* allow subsections and new functionality to be added or modified on the fly.

ELSA is maintained locally at ATM and is also posted to our internal GitHub site while we beta-test the software and address functionality issues. Although ATM designed ELSA based on our node's specific experience with atmospheric data providers, it could be adapted for more general audiences. Once the first fully operational build of ELSA is functional online, we intend to post to more publicly facing GitHub repositories.

Current Status. Users for ELSA are required to set up login credentials for a free account with ATM that allows users not only access to ELSA, but also allocates disk space at ATM to create bundles within the online environment. ELSA is currently set up to build bundles including all collections pertinent for completion of a bundle. Because of requirements and strong recommendations from ATM, ELSA automatically includes the default document, context, and XML schema collections as part of the bundle. These collections are generated inheriting the values for the URN LIDs based on user input for the naming of the bundle. The user (data provider) has the ability to add data collections and soon a host of other supporting collections (browse, geometry, calibration, etc.). Product-level files can then be added to each of the collections with inheritance of selected context product options. Currently, ELSA is designed to handle tabular data (binary, delimited, and fixed-width tables). Inclusion of some of the PDS4 local dictionaries is under development alongside the inclusion of basic arrays as upcoming valid data types. Internal beta-testing is ongoing at ATM with clean-up and editing occurring Spring 2021. Public beta rollout is tentatively scheduled to take place by the time of this conference (June 2021).

Future Work. Our current goals are to be able to allow label design for arrays, which in turn could be composited with already functional tables to move towards FITS file capabilities (which are common files in atmospheric science).

ATM plans to have ELSA rollout potentially other services We currently use the same development environment to create Python/Django-driven online review forms to help with science peer reviews of data sets submitted for archiving. With further refinement and interfacing ELSA's capabilities could be expanded to include full data submission allowing data providers to design labels, download the templates, apply them to their data, and then easily upload them through the same account system through the ELSA environment. We also plan to offer helpful tutorial walkthroughs to show users how different parts of ELSA work and what expected output should look like in the ELSA environment.

Conclusions. ATM's efforts to design an educational tool for an often-overlooked group of data providers, namely novice or first-time providers, that can also be used by veteran data providers will help PDS move into a new era of PDS4 archiving. ELSA will allow data providers an intuitive, straightforward option for designing and producing archive bundles for submission to PDS.

References. [1] "The PDS4 Data Provider's Handbook" (2020), V1.15.0, https://pds.nasa.gov/datastandards/documents/dph/current/PDS4_DataProvidersHandbook_1.15.0.pdf .