LABELS MADE SIMPLE, ELSA TEACHES STRUCTURE: LESSONS LEARNED ANEW

L.D.V. Neakrase, T.D. Pagán, K. Sweebe, J. Karl, L.F. Huber, N. Chanover, R. Beebe, V.I. Martinez, A. Hyder
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 providers, the PDS Atmospheres Node has developed a new online tool, the Educational Labeling System at Atmospheres (ELSA). ELSA provides an online web environment to build PDS4 bundles for simple data while teaching users about PDS4 structures.

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 PDS 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. Many PDS documents for data providers outline the rules for these identifiers. New providers or providers with limited experience with PDS4 may need to create these unique references for funded research projects, but may be intimidated by the complexity of the PDS4 archiving standard. ELSA is designed to aid in this task.

ELSA, Python, and Django. ELSA initially began as a student project developed with Java and PHP. Continued student activity required an easier way to maintain the code. The decision was made to leverage our previous coding experience for PDS3-to-PDS4 migration efforts with Python to expand our capabilities in aiding data providers with label support. A natural extension of this was to incorporate Python's Django libraries to create a web-interface for creating and editing label templates. The goal for ELSA is therefore to provide a way for novice and expert data providers to quickly create viable label templates. Django allows Python to be used to develop web applications, interfacing HTML with Python. This provides an easy framework for ELSA bypassing the need for Java or PHP, allowing multiple generations of undergraduate students to quickly pick up the coding language to maintain and manipulate the application.

Django's tools for the development of various web applications help compartmentalize the aspects of a bundle creation. Using the Python library *lxml*, undergraduate students are able to divide the PDS4 label template into unique components. Each component is handled by a web application designed around the formal and informal rules of the Information Model. This allows the students to create a more comprehensive and user-friendly interface.

Features and Capabilities. Potential data providers and users of ELSA are greeted with an initial log in screen. The initial account allows users to create a free login giving them space to play with PDS4 bundles within our system. The next screen prompts the user to create a new bundle or continue working with existing bundles from previous sessions. The Atmospheres Node encourages top-down bundle development. Data providers need to understand that submitting an archive to the PDS is more than handing over a few data products and requires the creation of an archive bundle.

ELSA's *Build* application allows users to build PDS4 templates for an entire bundle. This includes setting up initial internal references for their bundle, including creating the LID patterns to be used throughout, designating and creating the initial set of collections they plan to have, basic editing of many parts of the bundle label and selecting what types of data they plan to include.

The selected collections in the bundle provide stubs of templates for generating the collection labels where the user can add context referencing, internal and external referencing for documentation nearly completing each needed collection label.

Within each collection, the individual product label templates can also be started. Document collections allow users to select the type of documents and include consistent context referencing. Data collections, depending on the type of data selected in the bundle creation section, allow users start with appropriate data product templates, import in the necessary context referencing and allow for editing of column field headers. Currently ELSA only allows tabular data but includes fixed-width, delimited, and binary tables.

Expandability and Future Plans. Because of the modular approach taken in the development of ELSA, continued augmentation of its features is now possible. The decision to use Python/Django architecture ensures ease of use for future student software developers to quickly learn and compose modules for specific functionality. The modular nature of the ELSA development leveraging the Django application architecture allows future modules/applications to be added seamlessly to the core of ELSA, augmenting its functionality.

Specific future capabilities will include support for other data types beyond table formats. Simple arrays are next in our development schedule. With the inclusion of arrays and simple images, we are beginning to develop a simple interface for the incorporation of local data dictionaries, starting with the PDS4 discipline dictionaries such as Display, Geometry, and Cartography (in increasing complexity order). Efforts with migration of PDS3 data to the PDS4 archiving standard have allowed our students to become increasingly familiar with including dictionary support in label creation. Using this experience we aim to continue the ELSA philosophy of building a web-interface to simplify the process specifically for first-time, novice users.