Unified Reaction Structures for Astrophysics (URSA) - a new pipeline for nuclear data in astrophysics

Trevor Sprouse

Theoretical Division, Los Alamos National Laboratory

2022 NSAC Long-Range Plan Town Hall Meeting on Nuclear Structure, Reactions and Astrophysics
Nuclear Data Working Group

15 November 2022

Unified Reaction Structures for Astrophysics (URSA)

- **Problem:** nuclear physics is generally “hard-coded” into astrophysics modeling codes.
  - Difficult and/or impossible to adjust the nuclear data entering into simulations.
  - Has led to nuclear data for astrophysics being recorded in disparate data formats that are not readily transferred across simulation codes and research domains.
- **Solution:** Unified Reaction Structures for Astrophysics (URSA) codebase, a suite of tools designed to manage and query nuclear data in astrophysics applications.

URSA: design goals and implementation

- URSA approaches the problem of nuclear data in astrophysics as a two-step process:
  - Nuclear data selection and preparation
    - Out of the broader set of all nuclear data available to a user, select only those specific nuclei and nuclear properties appropriate to a specific application; prepare a single, self-consistent, and self-contained “nuclear data package” file
  - Nuclear data querying
    - Retrieve data from a nuclear data package file as-needed and when-needed over the course of a simulation.
- In particular, this approach formally separates the *implementation* of nuclear data in any given simulation from the details of the *nuclear data* itself.

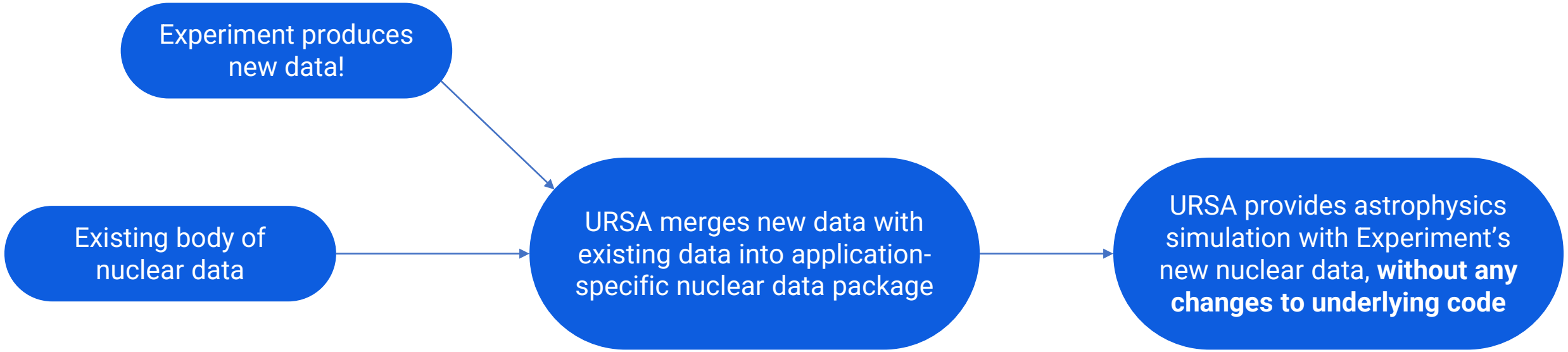
URSA: design goals and implementation

- The URSA codebase reflects this two-step approach:
  - Step 1 (nuclear data preparation) supported by a collection of tools written in Python
    - High-level programming language, chosen to specifically prioritize ease-of-use and minimize barrier of entry for end-users.
    - For most use-cases, customized nuclear data packages can be generated using only a few ($\lessapprox$10) lines of code.
  - Step 2 (in-simulation nuclear data querying) proceeds through a C(++) library that may be readily linked against in external simulation codes.
    - Written in a low-level, widely available language (C++)
    - Prioritizes computational performance, which is a major concern for many astrophysics applications.
    - C bindings => library can be implemented in codes written in a wide array of other programming languages as well (Fortran, Python, Julia, …)

URSA: design goals and implementation

- “Nuclear data packages” provide the connection between nuclear data specification (step 1) and querying (step 2) aspects of the URSA code.
  - Formatted using protocol buffers (Protobuf), which imposes a standardized, reproducible, and well-defined file structure (portable across computing platforms).
  - Data is serialized into a binary format that minimizes file size and dramatically reduces “load time” into the URSA library.
  - File format is sufficiently flexible so as to accommodate the diversity and complexity of nuclear data required by astrophysics applications.

URSA: An example use-case



URSA provides an effective and efficient workflow for immediately incorporating state-of-the-art nuclear data in astrophysics simulations, which will be especially important as FRIB (and other important experimental facilities) come on-line in the near future.

URSA: Current status

- URSA is generally prepared for initial open-source release
  - Basic tests for code stability, performance, etc. performed; community feedback from “production” usage will be extremely important for guiding future development.
- Following open source release, a concomitant publication detailing URSA will be submitted to the Journal of Open Source Software (JOSS).
- Has spawned complimentary development efforts within the broader nuclear astrophysics community, which several of the original project’s developers continue to lead / collaborate with.