Towards a new PDG computing system

J Beringer[1], O Dahl[1], K Jackson[2], C McParland[2], S Poon[2], D Robertson[2] and P Zyla[1]

[1] Particle Data Group, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Mail Stop 50R-6008, Berkeley, CA 94720, USA

[2] Computational Research Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Mail Stop 50B-2239, Berkeley, CA 94720, USA

E-mail: `JBeringer@lbl.gov`

Abstract. The computing system that supports the worldwide Particle Data Group (PDG) of over 170 authors in the production of the *Review of Particle Physics* was designed more than 20 years ago. It has reached its scalability and usability limits and can no longer satisfy the requirements and wishes of PDG collaborators and users alike. We discuss the ongoing effort to modernize the PDG computing system, including requirements, architecture and status of implementation. The new system will provide improved user features and will fully support the PDG collaboration from distributed web-based data entry, work flow management, authoring and refereeing to data verification and production of the web edition and manuscript for the publisher. Cross-linking with other HEP information systems will be greatly improved.

1. Introduction

The *Review of Particle Physics* [1] (known as the "PDG book") that is published every two years by the Particle Data Group (PDG) [2] is widely regarded as an indispensable resource for High-Energy Physics and beyond. In 1,422 pages, the 2010 edition includes 108 review articles and summarizes 2,158 new particle physics measurements from 551 papers, in addition to the 27,337 measurements from 7,749 papers that first appeared in previous editions.

The computing system that supports the worldwide PDG collaboration of over 170 authors in the production of this Review was designed more than 20 years ago. It has reached its scalability and usability limits and can no longer satisfy the requirements and wishes of PDG collaborators and users alike.

In this article we discuss the ongoing effort to modernize the PDG computing system. We summarize the scope of the project and discuss requirements, architecture and the technologies chosen for the implementation. We present examples of new features that are becoming available in the new system and summarize the status of the project.

2. Scope of the computing upgrade project

The computing system used by PDG until summer 2010 was originally built in the late eighties [3]. It was an extremely modern system that held up amazingly well over such a long period of time. In spite of several hardware and limited software updates, the system remained essentially a single-user system as illustrated in Figure 1. In particular, all data entry into the PDG database (from which the Review is produced) had to be done by a single editor. Obviously this scheme does not scale well to the present size of the PDG collaboration.

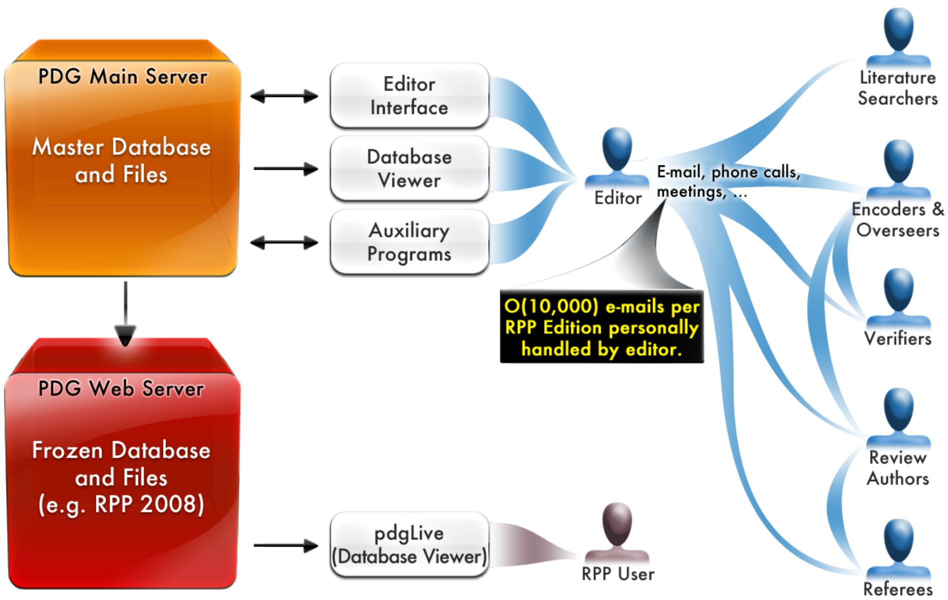

Figure 1. Conceptual view of the old PDG computing system. All data entry into the PDG database was done by the Editor, who received the relevant information from PDG collaborators by e-mail.

In 2008 a comprehensive project to modernize the PDG computing system was funded. The new computing system will support all areas of PDG work, including:

- Decentralized, web-based entry of measurement data by over 100 collaborators
- Data verification by hundreds of experts
- Authoring and refereeing of review articles, supporting over 130 review authors
- Monitoring of the progress in the production of the Review to assure timely publication
- Programs for data evaluation including complex fits, averages and plots
- Expert tools for the editor for creating the manuscript for a book of over 1,400 pages and the corresponding web pages and PDF files
- Tools for the HEP community (and beyond) to access, search and extract PDG data

The system is designed for long-term maintainability and will provide a suitable platform for future extensions. Such extensions might include, for example,

- A version of pdgLive running on smart phones (without requiring a data connection)
- Extending the PDG computing platform to support averaging groups
- User tagging of PDG content
- Enabling programmatic access to the PDG database outside of the PDG collaboration

3. Architecture

A conceptual view of the new PDG computing system is shown in Figure 2. In contrast to the old system, the new system supports decentralized, web-based data entry by each collaborator. All data needed for the publication of the Review is stored in the PDG master database and associated files. When an edition of the Review is published, the database contents are frozen and copied to the PDG web server and mirror sites, where they can be accessed by the public through web applications such as pdgLive [4].

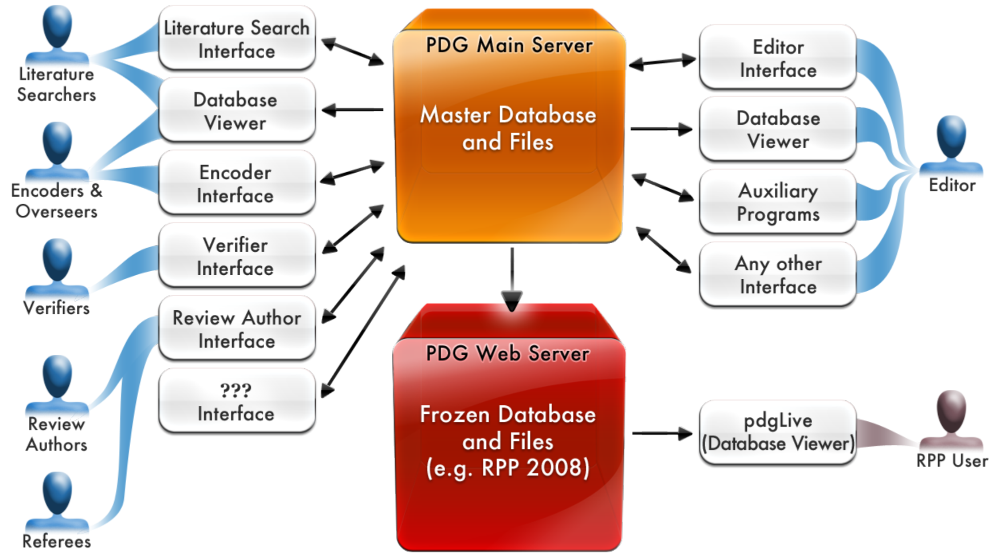

Figure 2. Conceptual view of the new computing system. Each collaborator can directly update the PDG database via a set of web applications tailored to his roles.

The new PDG computing system is based on a layered software architecture (Figure 3) that is built on top of a greatly modernized PDG database. All data from the old PDG database was imported into the new database. Except for legacy applications that were updated for the new database schema (they will eventually be phased out), all database access is handled through an API layer. This API layer contains most of the functionality needed by individual applications, including, for example, the processing of "PDG macros". PDG macros are used to separate scientific information from the details of rendering into different output formats and to specify how certain measured values should be updated based on new PDG averages of other quantities.

4. Technologies

The PDG computing system consists of two main groups of applications:

(i) Web applications provide convenient user interfaces to edit, update and view PDG data for collaborators and give the public access to PDG data.

(ii) Data analysis applications implement the necessary fitting, plotting, and other scientific data manipulations.

Separate APIs are provided for these two groups, using appropriate and well-established technologies in each case in order to ensure efficient development and long-term maintainability.

Web applications are implemented using a J2EE-based web application framework with Ajax-enabled web pages in order to provide user-friendly and highly interactive user interfaces. CSS and existing JavaScript code libraries are used where possible.

Data analysis applications need convenient access to ROOT [5] and other scientific code libraries that can more easily be called from a C/C++ based environment than from Java. In addition, for some applications, an interactive environment is highly desirable. Therefore Python with its existing integration with ROOT through PyROOT was chosen as the primary programming language for this part of the system.

Although separate APIs are implemented for the two groups of applications, the focus and functionality are rather different; therefore, this does not result in substantial duplication of API code. Where functionality from one API is needed in the other (particularly in the case

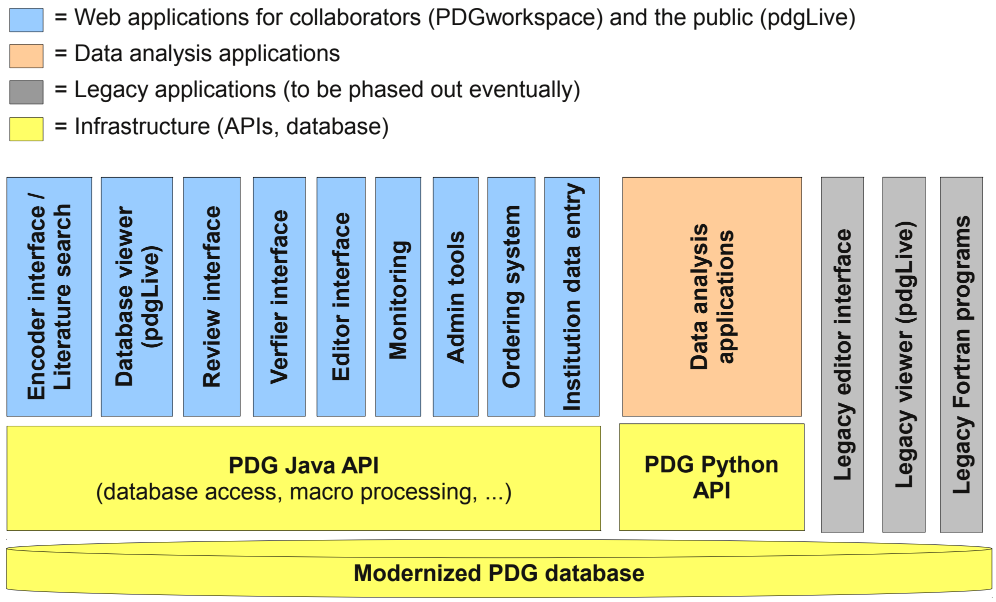

Figure 3. Layered software architecture of the new PDG computing system.

of PDG macro processing), it can be accessed through web services. Database access is based on industry-standard ORM (object-relational mapping) tools [6], accessing a PostgreSQL [7] database with currently 130 database tables.

A large amount of well-written Fortran legacy code has been developed by PDG over many years, both for data evaluation and for the production of the manuscript of the PDG book. It is not possible (nor necessary) to immediately replace all of this code, although eventually it will be phased out. Instead, a large fraction of this legacy code has been updated to work with the new PDG database and restructured as a library, so that it can be called from the Python framework and be reused in the new system where desired.

For a very large printed book with such an extensive need for equations and mathematical symbols TeX (or one of its variants like LaTeX) is an obvious and very well established choice. However, displaying math in a web browser quickly, in good quality, and in a platform-independent manner is a much more difficult task and requires special treatment for both visual rendering and speed. In this project the W3C MathML standard [8] was chosen for displaying math in web pages. In order to support older web browsers or slower computers, MimeTex [9] is used as a fallback solution.

5. Examples of new features

5.1. Distributed entry of measurement data

Decentralized data entry of PDG data requires web applications that are intuitive and easy-to-use with minimal training and that manage the workflow between different collaborators. In contrast to the PDG editor, most collaborators will use these data entry tools only on a casual basis. Therefore the web interfaces need to guide the user through the data entry process and shield him from the underlying complexity of the PDG database.

This is especially difficult when it comes to allowing users to define new particles and their branching ratios (or other new quantities to be listed in the Review), since a formalized representation of particles and other conventions need to be enforced. Therefore entering such

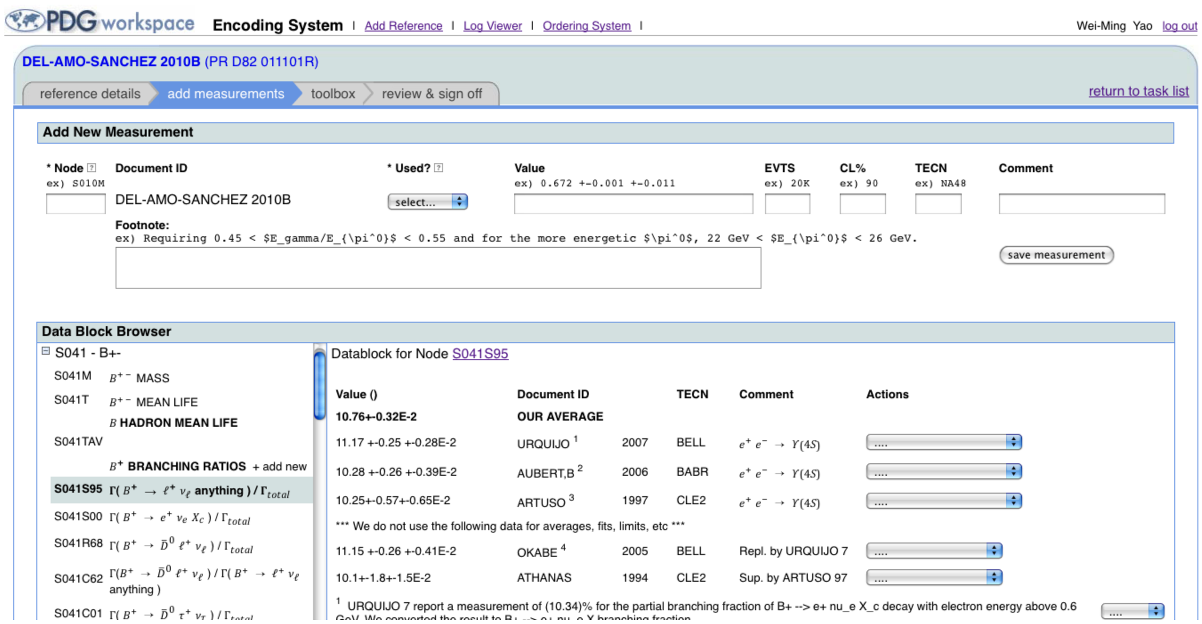

Figure 4. Data entry screen for entering new measurements into the PDG database. New measurements can be added using the fields at the top; existing measurements are displayed at the bottom.

quantities in ASCII form (using e.g. LaTeX notation) is not feasible. Instead, a solution where users can select quantities from hierarchical panels and drop them into place was adopted. This method allows users to easily define, for example, new ratios of existing branching fractions and to enter the corresponding measurement data.

The new computing system is designed to meet all of the above requirements. Figure 4 shows an example of a data entry page from the new system.

5.2. Programmatic access to the PDG database

Convenient and preferably interactive programmatic access to PDG data is highly desirable for a number of tasks, including exploring and plotting of PDG data, defining fits, and extracting data for use in other programs. The PDG Python API provides this kind of access, making it possible, for example, to extract and plot PDG data with just a few lines of Python code. For the time being, programmatic access to the PDG database is aimed at PDG-internal use only, but it might be made available outside of the PDG collaboration in the future as well.

5.3. Cross-linking with PDG Identifiers

PDG Identifiers are strings that can be used as permanent external references to PDG data items. They can be used to link to pdgLive web pages or to retrieve data using the PDG Python API. Both particles and their properties such as mass or decay modes with averages, fit results, and the corresponding list of published measurements, as well as complete review articles can be referenced. For example, “Q007TP” is the PDG Identifier for measurements of the top quark mass. PDG will publish an authoritative list of PDG Identifiers that, once defined, are guaranteed not to change.

PDG Identifiers can be used to link to pdgLive web pages or to retrieve data using the PDG Python API. They can also be used to cross-link PDG with other HEP information systems. For example, by storing PDG Identifiers of relevant PDG data in the metadata of articles catalogued by INSPIRE [10], support for cross-platform queries such as "What data does PDG have about the topic of this article?" (when looking at an article in INSPIRE), or "What are the latest papers on this topic?" (when looking at data in pdgLive) can be implemented. Work on such cross-linking between INSPIRE and pdgLive is in progress.

6. Status

Given the constraints from the ongoing production of the *Review of Particle Physics* and the migration of the existing PDG data into the new database schema, a first release of the new system (termed "V0 release") was deployed as early as possible. The V0 release includes the modernized PDG database, the PDG APIs, updated legacy applications running on the new database, and an alpha release of the application for entering measurement data (encoding interface). The latter is by far the most difficult and complex application. It's implementation relies on the same building blocks that will be used by the other applications. After extensive validation requiring, among other tests, the production of an identical manuscript of the PDG book in the new system, the V0 release was successfully deployed into PDG production work in August 2010.

The remaining year of the three year project is primarily devoted to the completion of the different end-user applications based on the infrastructure available in the V0 release of the new computing system.

7. Conclusions

A comprehensive modernization of the PDG computing system based on a modern, modular, web-based architecture implemented using industry-standard tools is in progress and expected to be completed on schedule in Fall of 2011. An initial version of the new system has been validated and is now used for the ongoing PDG work.

While the primary goal of the project is to address the needs within the PDG collaboration, the new system provides a platform where innovative new features can be implemented in the future. The priority given to any such new features will depend on the wishes from the wider High-Energy Physics community, and any suggestions or comments from the community are highly appreciated (please send your input by e-mail to **suggestions@pdg.lbl.gov**).

Acknowledgments

We thank K.S. Lugovsky and S.B. Lugovsky (COMPAS Group, Institute for High Energy Physics, Protvino, Russia) for essential prototyping for the present project, and for their long-term contributions and support of the earlier PDG system. This work is supported by the Director, Office of Science, Office of High Energy and Nuclear Physics, the Division of High Energy Physics of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231, and by the U.S. National Science Foundation under agreements PHY-0652989 and PHY-0966691.

References

[1] K. Nakamura et al. (Particle Data Group), *J. Phys. G* **37**, 075021 (2010)
[2] Particle Data Group, `http://pdg.lbl.gov`
[3] T.G. Trippe, Comput. Phys. Commun. **33** (1984) 229
[4] `http://pdglive.lbl.gov`
[5] `http://root.cern.ch`
[6] Hibernate (`http://www.hibernate.org`), JPA (`http://www.oracle.com/technetwork/articles/javaee/jpa-137156.html`), SqlAlchemy (`http://www.sqlalchemy.org`)
[7] `http://www.postgresql.org`
[8] `http://www.w3.org/Math`
[9] `http://www.forkosh.com/mimetex.html`
[10] `http://www.projecthepinspire.net`