DATABASE DEVELOPMENT FOR NSLS-II ACCELERATOR DATA MANAGEMENT*

J. Choi†, T. Shaftan, BNL, Upton, NY 11973, USA

Abstract

NSLS-II is developing a database which will be used for the accelerator data management. The information related to the operation is the main target at this stage. Also, various documents are being collected to provide easy access and installing workflow management is under consideration. The database will have web-based interfaces to communicate with the users. This paper overviews the database structures and required functionalities.

INTRODUCTION

Even though it is very useful and even indispensable for efficient operation of a complex accelerator like NSLS-II, collecting all relevant data and defining the relations between them is a huge task and requires big scale resources including a long time.

The undergoing database project is strictly limited by the available resources and is far from such a full scale development. Instead, as the first stage, we are targeting the full-functioning system for the operation related data and their interfaces, which can directly help to improve machine performance. That means we are taking the strategy that even if only part of the data are available we make them be used conveniently by constructing the relations and interfaces as much as possible. Then, as the data scopes are added, we connect the new data to the existing structure. Therefore, as well known, a more efforts should be invested for the useful interfaces than constructing the database itself. Furthermore, the resources are very limited, we should utilize convenient tools that help make the whole system even though the absolute amount of data is not so big at this stage.

OPERATION RELATED DATA

The operation related data for the first stage are listed as follows.

- Magnet and power supplies

  Each magnet's mechanical and electrical specifications.
  classification according to the design model.
  The connection to power supplies and the cable specifications.
  Power supplies' electric specifications and connected magnets.
  Magnet field measurement data and corresponding unit conversion.

- Physics parameter of operational lattices

  Magnet set-point currents in physical units for the operational lattices .(bare lattice, lattices with damping wiggle(s) (DW) closed)
  Twiss parameters at each element positions.

- Operation status

  Live status Beam current, lifetime, ID status, orbit, tune, synchrotron radiation

  History Variations of optics, orbits, and magnet set-point currents

- Operation related logics and procedures]

  Logic of active interlock
  Logic of top-off injection
  Start-up procedures
  Shutdown procedures
  Lattice tuning procedures

- Specifications of subsystems

  RF, IDs, diagnostics, control, vacuum

- Survey and alignment

  Accumulation of survey data and comparisons

- Maintenance tracking

  Regular and temporary maintenance procedure
  Error report and troubleshooting procedure

- Documents

  Design reports, drawings, technical reports

As mentioned, the first stage is focussed in improving the beam quality and beam availability. Even though not finalized, the database schema based on the above list is shown in Fig. 1.

DEVELOPMENT TOOLS

The developers understand well that the constructing organized database is critical just as the basic starting point. Providing good interfaces that make the data useful is sometimes more important and requires a lot more efforts.

Fortunately, various useful tools with public licenses are available which are not just useful but indispensable in developing the whole system from the data collection to graphical user interfaces. Especially, Python-Django provides convenient libraries which establish systematic connection between data and web pages. With these tools, decent web pages are being developed with limited resources.

The tools used in the development are listed as follows.

- MySQL [1]: Database

* Work supported by DOE contract No: DE-SC0012704
† jchoi@bnl.gov

ISBN 978-3-95450-147-2

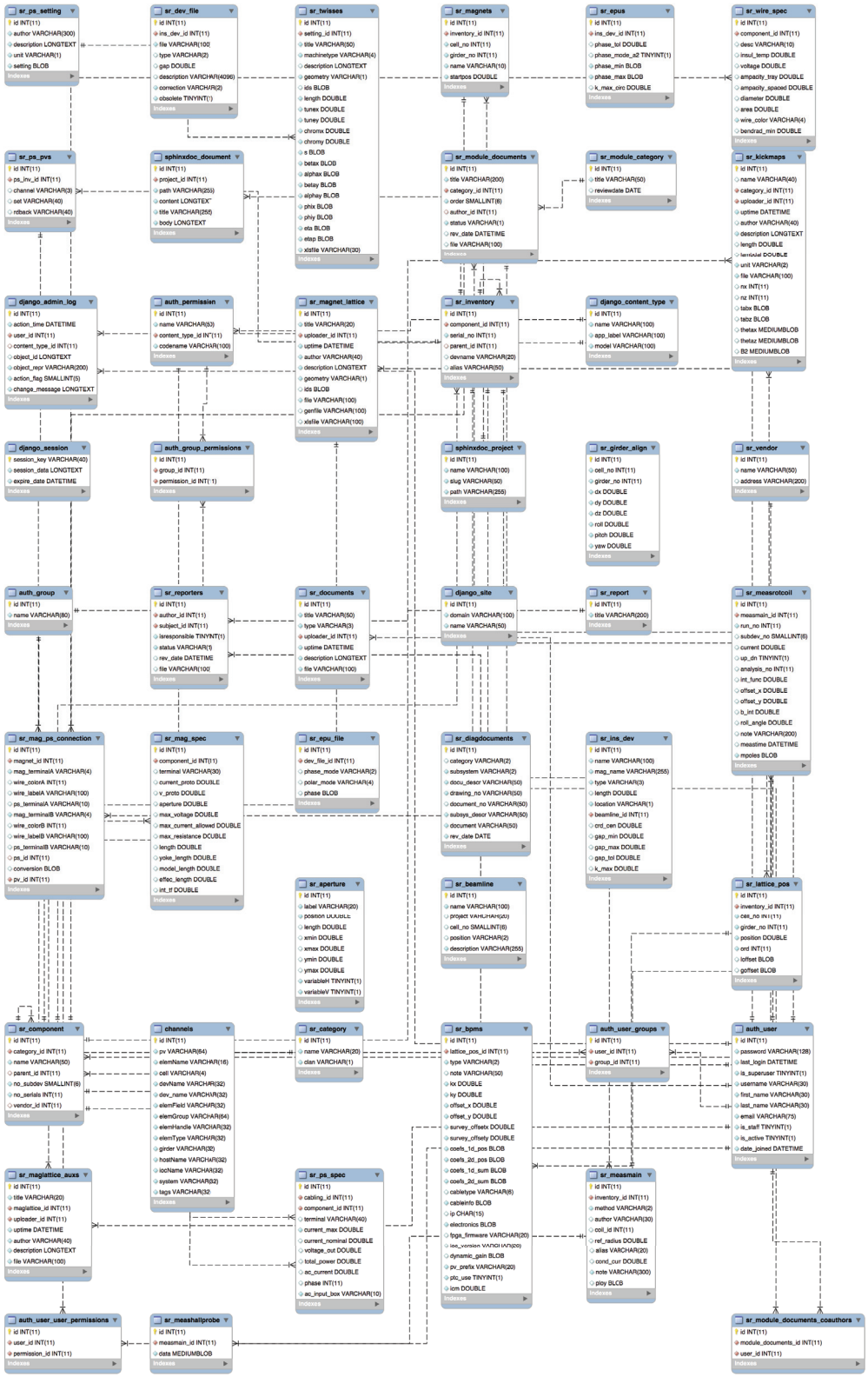

Figure 1: NSLS-II accelerator data management system schema.

- Apache2 [2]: Web Server
- HTML, CSS, Canvas [3] : Web Page
- JavaScript (jQuery) [4] : Client Application
- Python-Django [5]: Server Development
- Python-Sphinx [6] : Documentation
- Highcharts [7] : Web Plot
- Haystack [8], Whoosh [9]: Search

Even with the small amount of data, we try to make scale-independent well designed high-level software with these tools, so that expanding the database needs few corrections in softwares.

SOME ISSUES

In this section, we show some issues which need to be addressed.

Documents: From the start of NSLS-II project, huge amount of various documents have been generated and accumulated. Collecting and verifying their validness are already huge task and identifying the relations cannot be done in a reasonable time. Accordingly, at first, we just collect them and make them accessible through the web interfaces. For that purpose, we made a software which converts the document files, once collected, into PDF format and generate interface addresses. For the spread sheet documents like MS Excel, where PDF format is almost useless, we made another software which generates the HTML documents having the same cell formats as the original documents. Using theses tools, we can make the documents, once verified, instantly available from the web browsers.

Operation status: The ideal database for the operational status will be gathering data from the accelerator control system directly or from the archive system and showing the status in real time. However, because of the security, the control system cannot be communicated directly from outside and manual processing is unavoidable. Even though the manual methods can be improved to some extent, showing the real live status will not be realized.

Maintenance tracking: This maintenance tracking system would be the most useful part and, at the same time, the most difficult part to develop. Even though the basic steps are simple like, open-assign-plan-work-verify-close, many aspects should be involved in the workflow. Fortunately, there are excellent examples like CATER of SLAC [10], and we should study in full depths the existing systems as well learn the experiences of NSLS-II before deciding the database structure and workflow logics.

REFERENCES

[1] MySQL, https://www.mysql.com

[2] Apache2, https://httpd.apache.org

[3] HTML, CSS, http://www.w3schools.com

[4] jQuery, http://jquery.com

[5] Django, https://www.djangoproject.com

[6] SPHINX, http://sphinx-doc.org

[7] HIGHCHARTS, http://www.highcharts.com

[8] Haystack, http://haystacksearch.org

[9] Whoosh, http://bitbucket.org/mchaput/whoosh/

[10] C. W. Allen, S. Anderson, R. Erickson, W. Linebarger, J. C. Sheppard, and M. Stanek, "Opportunistic or Event-Driven Maintenance at the Stanford Linear Accelerator Center," SLAC-PUB-7424, 1979.

ISBN 978-3-95450-147-2