Beam Physics Note
TRI-BN-19-16
July 30, 2019

`HLA-beamData`: An Elementwise Database for Passive Beam Data Accumulation

Olivier Shelbaya

TRIUMF

Abstract: The structure of a passive beam-data collection database, operating within the TRIUMF-HLA framework is presented. This database is intended to accumulate beam diagnostic information, concurrent with use of high level applications, providing a naturally growing set of information for accelerator performance studies and analysis.

Introduction

Is the operations e-log too cluttered? This is a recurrent question members of the Operations, Beam Delivery and Beam Physics groups. The combination of its heavy subscription by support groups and its use a task tracking tool for shift operators makes structured information storage difficult, particulary for accelerator research. As the volume of beam observation data increases, parasitic use of the operations e-log clutters their workspace. In addition, any accelerator data can only be accessed via the e-log search function, which is not designed for this purpose.

This document explores the methodology in which control room HLAs may serve the dual purpose of enabling operational duties while simultaneously saving information to a structured storage. In this way, continuous accumulation of this data, concurrent with the regular tuning and troubleshooting duties of operators, can be made to provide physicists and coordinators access to beam and machine data, which may be used for accelerator studies and analysis, etc..

Keeping in mind the diverse base of proposed users, any data storage scheme which involves an overly complex or intricate infrastructure may passively discourage usage. In particular, reliance on third party APIs may impose a steep learning curve upon users who simply wish to access information for machine and facility studies. As such, it is advantageous to attempt to define the simplest, most robust framework to enable passive, continuous beam-based information collection, storage and sharing, through the HLA framework.

Requirements

This section presents a collection of ideas that will guide development of the HLA-`beamData` information storage system. These are not intended as strict or fundamental rules, but rather as broad guidelines meant to encapsulate the essence of what is desired.

Data Storage and Access

Forcing users to install third party software of any kind is considered impermissible. There must be a browser based option, in line with the remainder of the HLA suite of web-based applications. The web frontent will allow coordinators or scientists to access beam information and produce outputs consistent with their preferred working tools. Raw data from beam diagnostic devices will be output in the form of `ascii`-files, as these can easily be used by a variety of data analysis programs.

In parallel to manual frontend access from a browser, an indexing database such as `redis` will act as a global data index, containing time- and beam- stamps for each stored beam measurement file. This will enable access of raw datafiles from other HLAs.

HLA-Specific Requirements

The HLA-`beamData` repository must be fully modular and require no modifications to existing applications or databases. For instance, elaborating a scheme which requires modification of existing HLA infrastructure, for example the `/acc` database, is not be permissible. This would, in effect, set a precedent regarding retroactive modifications on existing applications, wasting time and effort.

Instead, information collection must take place in a transparent and self realizing manner. Concretely, this requires the introduction of a few input and output subroutines, which will provide a standard method of accessing stored beam data via the HLA framework. The remainder of this document details the conceptual operations that must be carried out to realize this idea.

Transparency to Operators

The saving of diagnostic information must occur automatically through use of HLAs. As such, applications that deal with beam diagnostic collection (e.g. `snapshot`, `tomography`) must not require additional user steps. As an application runs through its sequence, collected diagnostic information must be saved in a manner that does not introduce additional perceptible workload, such as an additional saving procedure, that may either slow the application or be forgotten.

HLA Ownership

This proposed database remains the sole property and responsibility of the HLA project group, or any successor entity, which will exercise full operational discretion on its evolution. In particular, proposed additions, modifications or structural changes, in addition to future developments, remain the exclusive prerogative of the HLA project group, or any successor entities, without justification.

Scientific Integrity

Any and all information that is committed to this database is considered to represent scientific data. As such, it must be regularly backed up on more than one physical machine. **It is not permissible to edit data on the database/file repository in any way after it has been saved.** Users may edit downloaded data as they please.

Elementwise Data Storage

The `beamData` structure is schematically represented in Figure 1. To maximize flexibility and ease of use, raw diagnostic data is stored in `ascii` formatted datafiles, stored in folders bearing the name of the collecting diagnostic device. In parallel, using a `redis` database, indexed information relating individual measurements to beam properties and time is stored.

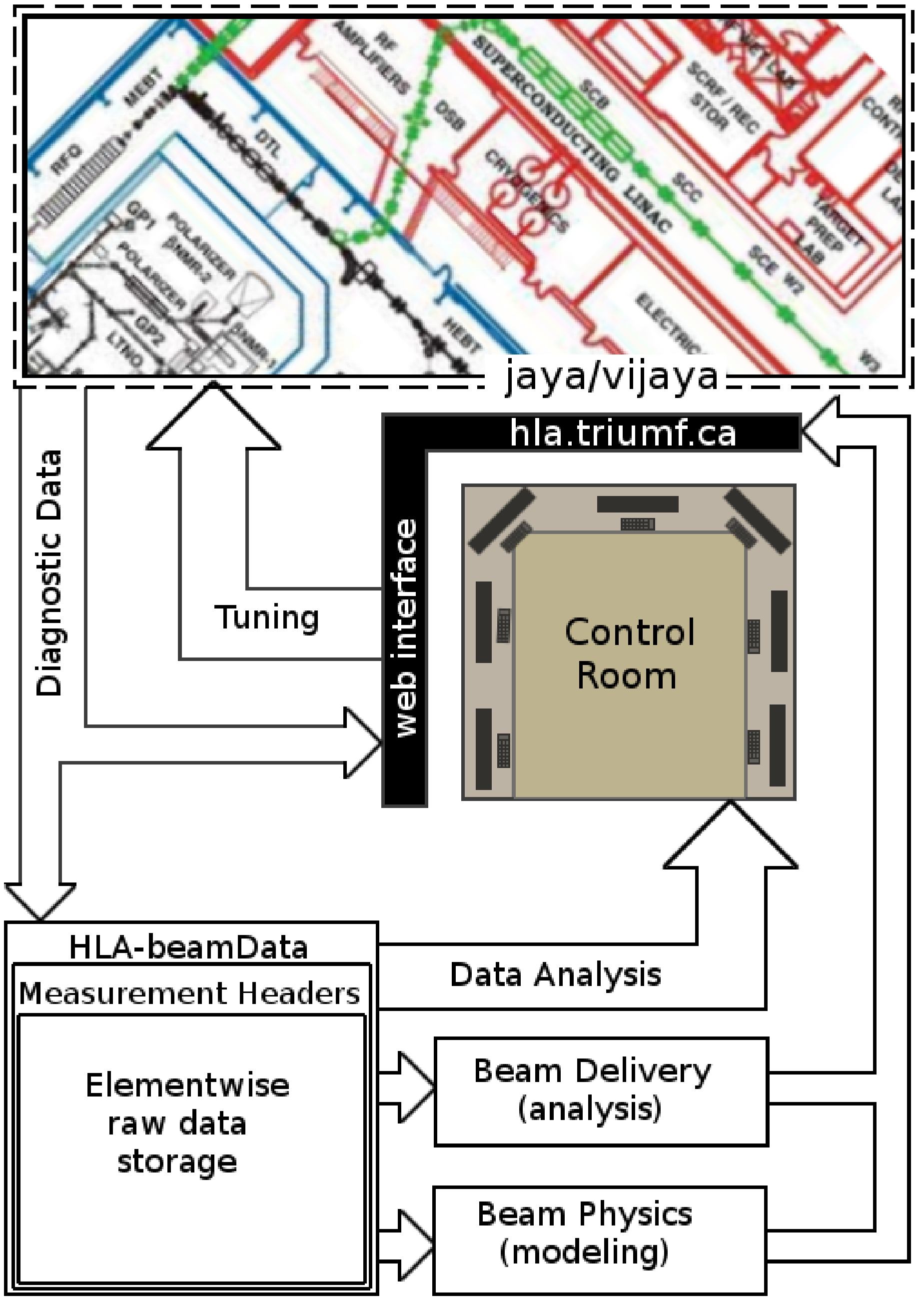

Figure 1: Schematic representation of the relationship between the Control Room (Operations), the HLA interface and its web-applications which enable tuning, diagnostic data collection, storage and analysis by the Beam Physics and Beam Delivery groups, which use the information to refine accelerator models and HL-applications. At the top, the jaya/vijaya communication interfaces are shown, providing a direct read/write bridge between the HLA server and the control system. Operators and coordinators may also access this information for data analysis and future application development.

The `beamData` storage structure is defined as follows:

- All raw diagnostic data is attached to the diagnostic device (element) which was used for its collection, by generating folders corresponding to each element, such as Faraday Cups or RPMs.
- raw data is stored in the form of `ascii` files in the element folder.
- Each `ascii` file contains one single measurement.
- Each `ascii` file receives a key-name, which is the date at which the measurement was performed, with the format `YYYYMMDD_HHmmSS.dat`.
- A database contains **measurement headers**, which consist of beam- and time-stamps pointing to each element involved in the measurement.
- Files and measurement headers are automatically generated as soon as the measurement is complete.

For instance, a transmission snapshot accomplished via the HLA-`snapshot` application produces a measurement header entry in the database. This entry contains beam information, provided by the `snapshot` application, for example:

Example Measurement Header

```
time="2019-07-20 11:35:27" beam="22Ne" source="olis" dest="Prague"
sourceAQ="5.5" mebtAQ="5.5" hebtAQ="5.5" eDTL="1.50*MeV/u"
cupSequence="IOS:FC6,ILT:FC33,ILT:FC49,MEBT:FC5,MEBT:FC9,HEBT:FC5"
rpmSequence="ILT:RPM33,MEBT:RPM5,HEBT:RPM5"
```

In the above example, a transmission snapshot is taken from OLIS to the Prague (HEBT1:MB0) magnet, for ^{22}Ne^{4+}, containing both Faraday cup and RPM scans. As such, the raw diagnostic readback for each cup and RPM would be stored in folders corresponding to the devices, for example `/beamData/elements/IOSFC6/20190720_113527.dat` contains IOS:FC6 data:

Example Datafile 20190720_113527.dat (IOS:FC6)

```
2019-04-20 11:35:27 3.68e-09
2019-04-20 11:35:28 3.21e-09
2019-04-20 11:35:29 3.44e-09
2019-04-20 11:35:30 3.53e-09
2019-04-20 11:35:31 3.72e-09
```

In the above example, the transmission snapshot involved a 5 second averaging of the beam current. All samples used to compute the average are stored in the raw datafile under the respective element folder, enabling post-processing. For the above IOS:FC6 data, the averaged current assuming 5 samples 1 second apart is 3.516 nA. Using the corresponding measurement header, all necessary beam and path information are available for post-analysis.

The above proposed structure will allow for natural interfacing with HLAs, which can easily access the database headers through appropriate subroutines. In addition, the `ascii` formatting of individual files does allow for manual access to files by users who do not wish to use the database structure. Indeed, by looking up the operations e-log timestamp associated with any given HLA measurement, the files from each element can be found by searching the folder structure for the key-name, which is simply the date of the measurement.

Further, while storing measurements in individual files will over time lead to folders with perhaps many thousands of files, it avoids accumulated performance limitations that would arise from storing all diagnostic data in single files, which would grow over time and have to be parsed. By using single files per measurement, the computational burden of accessing the data is always the same.

Conclusion

The present note outlines a passive data storage strategy which will provide access to a naturally growing set of beam based measurements, performed in a noninterfering manner via the HLA platform. In this way, facility and accelerator studies and monitoring may be performed in a minimally intrusive manner, reducing the need to schedule beam-delivery interrupting measurements for both the Beam Physics and Beam Delivery groups.

Acknowledgements

The author wishes to thank Thomas Planche, Spencer Kiy, Paul Jung and Rick Baartman for useful discussions and patience while elaborating the present idea.