

## RELATIONAL DATABASE SYSTEM FOR J-PARC LINAC AND RCS

H. Sako<sup>1</sup>, N. Kamikubota<sup>2</sup>, H. Sakaki<sup>1</sup>, H. Takahashi<sup>1</sup>, H. Yoshikawa<sup>1</sup>, M. Kawase<sup>1</sup>, and K. Watanabe<sup>1</sup>  
<sup>1</sup>*Accelerator Group, JAERI, Tokai-mura, Ibaraki, JAPAN*

<sup>2</sup>*High Energy Accelerator Research Organization, Tsukuba-shi, Ibaraki, JAPAN*

### ABSTRACT

LINAC and the Rapid Cycling Synchrotron (RCS) in the J-PARC accelerator complex consist of a huge number of accelerator machine components. They produce extreme high power beam of 1 MW. To minimize radiation due to beam loss, systematic control of machine parameters of components as the single accelerator system is required. RCS is further complex system where each beam pulse of 25 Hz is injected either into the Material and Life Science Facility (MLF) or into the 50 GeV Main Ring (MR). In order to store these complex control parameters as the only data source shared by the whole control system and accelerator components, a relational database (RDB) system has been developed.

The RDB system consists of three kinds of RDB; a machine database which records static machine parameters, a data acquisition database which records data taken with the data acquisition system, and an operation log database which records history of operation parameters.

The machine database stores static machine information about control system of each accelerator component. One of the main goals of the database is to generate EPICS configuration files automatically. A JAVA code has been developed and tested successfully. The machine database also stores geometry information and conversion parameters between device parameters and physics parameters.

The overall RDB system is designed to have close connection to the high level accelerator operation applications and online simulators. The high level application can save and restore operation parameters with the operation log database. The same set of parameters is used to generate simulation input files together with the geometry data from the machine database. The generator code in JAVA has been developed for XAL and Trace3D, and SAD simulation codes. The code utilizes XAL framework [5] developed in SNS. By conversion between device and machine parameters from the machine database, the high level application can directly compare operation parameters of the machines with simulations.

The design and the current status of the RDB system are discussed.

### INTRODUCTION

In the control systems of J-PARC LINAC and RCS, a unified RDB system has been adopted to store and maintain various control parameters, machine parameters, as well as commissioning and simulation parameters. Considering cost free feature, a large population of users, and expected rapid future developments, PostgreSQL has been adopted. In this paper, we present design and development status of the RDB system, focusing mainly on Machine DB and Commissioning DB.

### DESIGN OF THE RDB SYSTEM

Base design principles of the RDB system required by J-PARC control system are as follows.

1. Simple maintenance procedures of machine and control parameters, especially, capability of automatic generation of EPICS definition files.
2. Record of synchronized data in RDB with powerful analysis and search capability.
3. Close connection to high level applications.

Item 1 is necessary to maintain a huge number of interfaces, IOC's, and machines with limited manpower. This will be realized in Machine DB shown below. Item 2 is very important to analyze data from huge number of machines to investigate tuning parameters to reduce beam loss as low as

possible. This will be solved by DAQ DB. Item 3 is very important for efficient commissioning with quick operation of devices and comparisons of measured data with simulations.

The overall RDB system has been designed in close connection to the control application software as shown in Fig. 1.

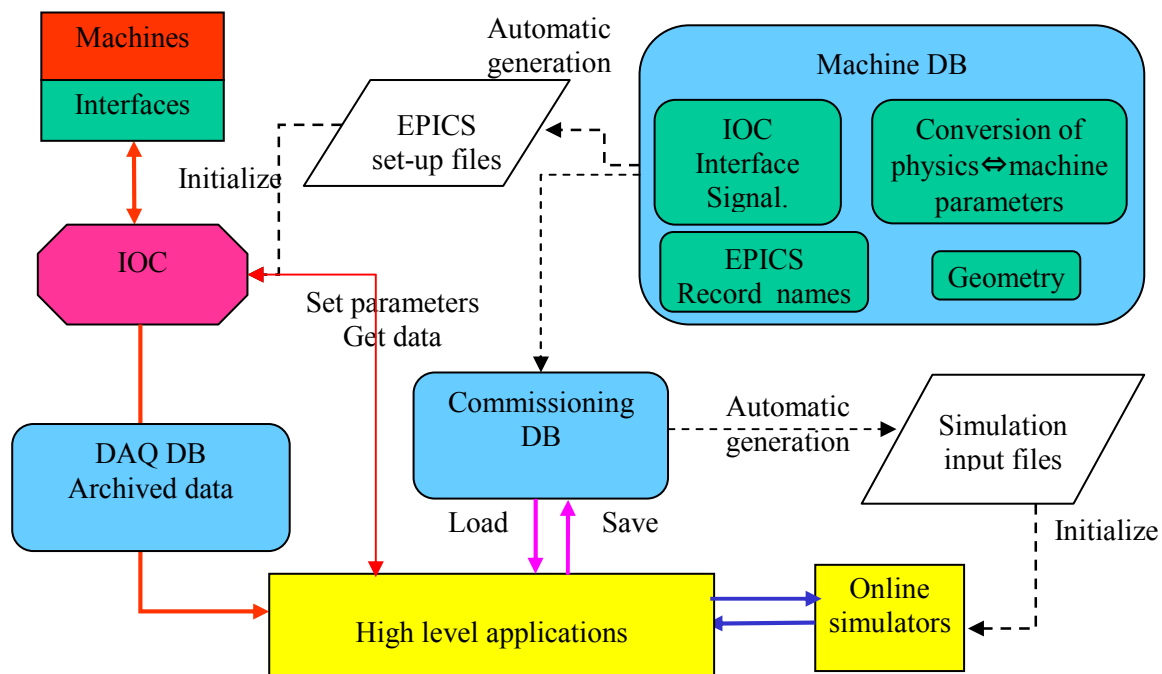


Fig. 1: Relation of RDB system with high level applications

Machine DB stores device parameters and configuration parameters for the control system. Information of signals used in the control system is most important. Most of devices except for a part of beam monitors are controlled based on EPICS. EPICS records for operation, interlock, and setting and monitoring analog/digital data are defined for various accelerator devices. Each signal is also mapped to each address of interfaces. We have developed common table structures to store parameters for different kinds of interfaces, PLC, VME, and EMB-LAN interfaces. A Java code has been developed to generate EPICS configuration files for these interfaces. EPICS record names and their relational links to power supplies and machines are used to configure high level control applications. For commissioning, it is planned to use a Java based SAD script interpreter (JAD) [4], which takes input files which are automatically generated by the Machine DB including such links. Geometry information further helps for visualizing lattice of machine component for high level accelerator control software. Information of accelerator machine components is also important to keep track of replacement of devices and to obtain key parameters for commissioning or simulations. Especially, position and dimension information of each component is used for simulation.

DAQ DB stores history of online data from various accelerator components such as beam monitors, magnets, RF components, and vacuum devices. A very important point for beam monitor data is synchronization. We are going to store data into RDB synchronized with a common shot ID for all monitors and other devices. This is important especially for RCS, since at 25Hz the beam is injected from RCS into two different beam transport lines to Main Ring (MR) and to Material and Life science Facility (MLF). For each destination, beam parameters in RCS may be different.

Recording data into distributed files, not into RDB is easy to achieve enough data collection speed, while it is a painful work to collect all the data from different locations and check synchronization, and data search speed may be slow. Compared to this scheme, our scheme to record synchronized data into RDB is very powerful in terms of search and analysis.

Main amount of our data comes from beam monitors. The total amount of data is expected to reach several GB per day in RCS. A direct recording system to RDB of such amount of data is a challenge, therefore intensive bench tests have been performed [1,2].

Commissioning DB stores the following information.

- Simulation related parameters (Simulation DB)
- Device operation parameters and simulation input parameters (Operation Log DB)
- Geometry of beam-line devices in the orbit coordinate and intermediate geometry information transformed from alignment information in Machine DB
- Coefficients for unit conversion functions from/to device parameters and physical parameters

Simulation DB stores simulation parameters of each beam-line device (magnets, monitors, RF's, and so on) as well as beam parameters. Geometry information transformed from Machine DB will also be stored. Since geometry data has not been available inside Machine DB yet, all the data required to generate simulation input files is temporarily stored in Simulation DB. In the future it is planned to relate measured geometry information from Machine DB to Commissioning DB, utilizing "dblink" user function available for PostgreSQL.

Operation Log DB aims at storing history of operation parameters of beam-line devices. Currently, it is used only to store simulation input parameters to generate simulation input files. In the future, a tool to read snapshot of all the current operational parameters via EPICS and record them to Operational Log DB is necessary. Also a mechanism to store fudge factors which absorb inconsistency between simulation parameters and device operation parameters may be required in order to share the same set of operation parameters in controlling devices and running simulations.

In summary, we explain designed procedure to control devices with a high level application shown in Fig. 1. First, we load a basic set of operation parameters from Commissioning DB. After some modifications of parameters, we generate simulation input files from the parameter set. Then, we run an online simulator to calculate beam parameters and orbits through the lattice and if required also perform beam matching procedure. The modified set of parameters is then recorded to the Commissioning DB, and at the same time the parameter set is finally applied to accelerator devices via EPICS. After running accelerators, we analyze data recorded in DAQ DB in the high level application, and compare the data with the calculations by the simulation. Finally the results may be fed back to the next set of operation parameters.

The initialization and configuration of the system in Fig. 1 is also based on parameters from the RDB system. EPICS record set-up files are generated from Machine DB for each IOC. IOC's are initialized with these files. Control applications are initialized with EPICS record lists as well as links to power supplies and accelerator components stored in Machine DB. Conversion functions between physics and device parameters will be used by a unit transformation software. A design of the unit conversion scheme is currently under development.

## STATUS OF DEVELOPMENT

Here, we show status of development and prototype tests for each kind of database.

### *Machine DB*

We have designed and constructed tables for each type of EPICS record (*e.g.* ai and ao types), to generate EPICS record definition files for PLC, VME, and EMB-LAN interfaces. We are currently storing data to these tables for magnets and vacuum devices for L3BT. Using generated record definition files, we plan to perform control tests for these devices in October 2005 for L3BT. Note that this automatic generation requires a unified interface for EPICS device supports for PLC, VME, and EMB-LAN, which have been developed by us. In parallel to these activities, we have been collecting basic machine information and control signal information for all the machine components for LINAC and RCS. Collection of data for main components will be done in October 2005, and then redesign and reconstruction of tables are planned.

For machine maintenance, we shall take a dual table configuration for each device, namely a configuration with a role table and an entity table. The entity table stores information for all

individual machines characterized with their own serial numbers. The table includes also spares and broken devices as well as currently used devices. The role table represents data for relations to other devices and geometry configurations, which are unchanged by replacement of real devices. By changing links between the two tables, one can update machine information and record logs of replacement of devices easily.

We have developed EXCEL based template files to collect data common to each device. We have made two kinds of templates; a hardware template which includes parameters of machine components, and a signal template which includes signal information. We distributed the template files to each machine group, who is responsible to collect the data.

### DAQ DB

We have developed two schemes for the data acquisition to RDB. For beam monitors of RCS, we require synchronization at each 25 Hz pulse with a common shot ID. We transfer data from the monitors at 25Hz to the reflective (shared) memory which is configured as a ring memory. Then, a block of data accumulated for 50 pulses is transferred and inserted into the RDB. In this scheme, we have performed a bench test and succeeded in recording 25 Hz synchronized data to RDB for the expected number of beam position monitors in RCS. Also, we have developed a tool to read EPICS channel data and write it into RDB directly. A more efficient logging scheme by event notification with “CaMonitor” function in EPICS is under development. Detailed information of the tests is described in Refs. [1, 2].

### Commissioning DB

We have developed a prototype of Simulation DB including Operation DB for simulation. We need to combine information of geometry with operation parameter of devices from the operation log DB. For this process, transformation of the coordinate system from J-PARC system into the orbit system is done. In the future, inclusion of geometry errors due to alignment measurements is planned. We have also developed a JAVA tool to read the DB information and generate simulation input files for XAL online simulator developed in SNS [5]. Currently, basic machine components (quadrupole and bending magnets, beam position monitors, and RF gaps) for LINAC and RCS are included. We have also developed JAVA tools based on XAL framework, which transform the input XAL file into three kinds of simulation input files; Trace3D, SAD, and MAD (see Fig. 2). Detailed comparisons of Twiss parameters and closed orbits have been performed with qualitative agreements between the four simulation codes with the same set of input parameters form the Simulation DB (see Fig. 3).

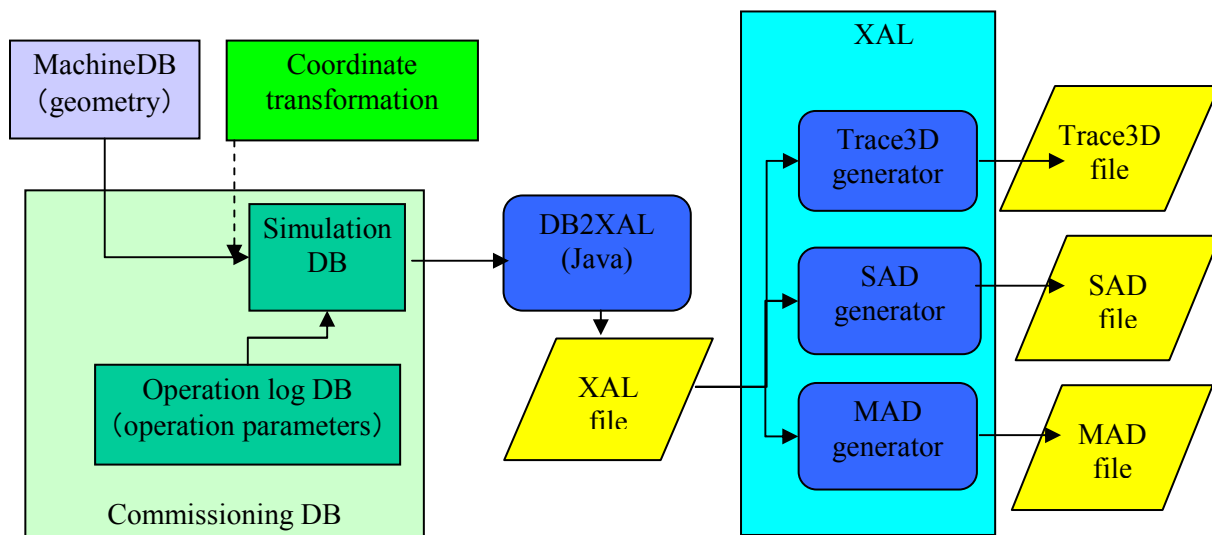


Fig. 2: Scheme for producing simulation input files

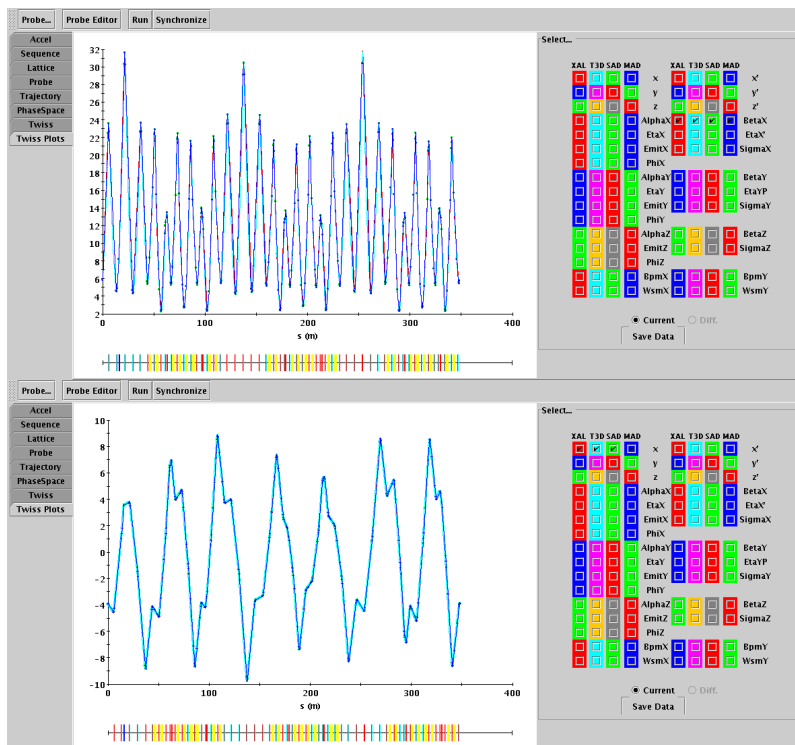


Fig. 3: Comparison of the beta functions (top), and closed orbits (bottom) in the horizontal direction for XAL, TRACE3D, SAD, and MAD.

We have also tested successfully save and restore functions of Operation Log DB using JAD and a GUI application, generation of TRACE3D simulation file, and magnet controls, in LINAC-MEBT1 test bench at KEK in Feb. 2005 [3].

#### Other plans for RDB tools

The I/O interface to RDB is in general not trivial for most of users, unless they have deep knowledge about RDB structures. For commissioning, we may use an application tool which has only EPICS channel access interface without an interface to RDB. In such case, a RDB-EPICS gateway may be a good solution. A prototype RDB-EPICS gateway is currently being developed. It is based on Portable Channel Access Server (PCAS) and modified to read and write data from/to a relational database table instead of EPICS runtime database. A user can read, write, and monitor data of each EPICS record via normal EPICS commands, whereas actual data is stored in a RDB table.

Another request from users is a unit conversion of device parameters between device units and physical units. For instance, one may set or read parameters of a magnet in magnet field (T) instead of in current (A). In such a case, several strategies can be considered. KEKB adopted a scheme to perform transformation in the EPICS record level. In the IOC which controls a device, the unit transformation is also done. We have tested this scheme using LINAC-DTQ magnet with the EMBLAN interface on April and June 2005 [3]. The test has been successful, but we encountered the following problems. In this scheme, modifications of transformation functions require IOC reboot. Another disadvantage of this scheme is complexity of the EPICS links and record definitions, where automatic generation of record definitions is very hard. To solve these problems we are now planning to develop a channel access server. The server reads a raw EPICS channel connected directly to a device, transforms the raw value into a physical value in a transformation function, and sets the physical value to a physical record which is created by the server. The opposite operation is also possible, namely writes a physical value to a physics record and a raw value obtained from inverse transformation is set to the device record. Also, monitoring of physical values must be possible.

There is yet another way, namely doing transformation inside application. However, this requires exactly the same transformation functions for each application. In J-PARC, several different applications may be used and this method may not be recommended.

## CONCLUSIONS

We have designed the overall scheme of the RDB system in combination of control applications. The RDB system has several key functions; automatic generation of EPICS record definition files, direct recording of synchronized DAQ data into RDB, and usage of Commissioning DB for high level applications. Machine DB, DAQ DB, and Commissioning DB have been confirmed to achieve key functionalities and performance. Final production of the RDB system and collection of data will start soon.

## REFERENCES

- [1] H. Takahashi *et al.*, “Data Acquisition System for J-PARC 3GeV RCS”, these proceedings.
- [2] H. Takahashi *et al.*, “Summary of 3GeV RCS Control System (2)”, The 2<sup>nd</sup> Annual Meeting of Particle Accelerator Society of Japan and the 30<sup>th</sup> Linear Accelerator.
- [3] H. Sako, *et al.*, “RDB System in J-PARC LINAC and Its Application to Commissioning”, The 2<sup>nd</sup> Annual Meeting of Particle Accelerator Society of Japan and the 30<sup>th</sup> Linear Accelerator Meeting in Japan, Saga, July 2005.
- [4] H. Ikeda, *et al.*, “Development of a SAD Script Interpreter with Java”, The 2<sup>nd</sup> Annual Meeting of Particle Accelerator Society of Japan and the 30<sup>th</sup> Linear Accelerator Meeting in Japan, Saga, July 2005.
- [5] C. K. Allen, *et al.*, “a Novel Online Simulator for Applications Requiring a Model Reference”, Proceedings of ICALEPS2003, Gyeongju, Korea