The Spallation Neutron Source (SNS) is a pulsed neutron source that uses the most power in the world for use in scientific research and industrial development. In order to achieve the production of neutrons at the target using 1.4 megawatts (MW) of operating power, the SNS Beam Instrumentation Group (BIG) monitors the accelerator performance. Proper beam quality and transport will ensure high availability of neutrons for user experiments. To facilitate the monitoring of the accelerator at multiple levels, from the control room to executive summary reports, a reporting system was first envisioned in a LabVIEW display [1], [2]. A prototype was developed by Madhan Sundaram, Neutron Science Systems Software Engineer at Oak Ridge National Laboratory (ORNL) using MySQL, Personal Home Page (PHP), Asynchronous JavaScript and Extensive Markup Language (AJAX) scripting. The prototype demonstrated the ability to refine the initial reporting capability of a database using a common web interface for viewing. The prototype was developed further during the summer of 2007 to incorporate Java programming for data acquisition and to interface with an Oracle database cluster recently introduced at SNS. The reporting system continued to be developed at ORNL by Jeff Patton, Software Systems Administrator, and Douglas J. Edwardson, Scientific Systems Programmer, both within the Oracle database and with the modification of the user interface to use Adobe's Flex Technology. The revamped reporting system accesses data stored in the database and implements Java classes incorporating pre-defined and user-defined analytic functions. The reporting system will be the foundation for analyzing data taken throughout the accelerator. It provides a simple and intuitive user interface that is cross-platform and easily accessible for the design, viewing and distribution of reports throughout SNS or outside the SNS facilities (via e-mail).

**Development of Accelerator Data Reporting System and Its Application to Trend Analysis of Beam Current Data**

**MARIANO J. PADILLA AND WILLEM BLOKLAND**

**ABSTRACT**

Detailed ongoing information about the ion beam quality is crucial to the successful operation of the Spallation Neutron Source at Oak Ridge National Laboratory. In order to provide the highest possible neutron production time, ion beam quality is monitored to isolate possible problems or performance-related issues throughout the accelerator and accumulator ring. For example, beam current monitor (BCM) data is used to determine the quality of the beam transport through the accelerator. In this study, a reporting system infrastructure was implemented and used to generate a trend analysis report of the BCM data. The BCM data was analyzed to facilitate the identification of monitor calibration issues, beam trends, beam abnormalities, beam deviations and overall beam quality. A comparison between transformed BCM report data and accelerator log entries shows promising results which represent correlations between the data and changes made within the accelerator. The BCM analysis report is one of many reports within a system that assist in providing overall beam quality information to facilitate successful beam operation. In future reports, additional data manipulation functions and analysis can be implemented and applied. Built-in and user-defined analytic functions are available throughout the reporting system and can be reused with new data.

**INTRODUCTION**

The Spallation Neutron Source (SNS) is a pulsed neutron source that uses the most power in the world for use in scientific research and industrial development. In order to achieve the production of neutrons at the target using 1.4 megawatts (MW) of operating power, the SNS Beam Instrumentation Group (BIG) monitors the accelerator performance. Proper beam quality and transport will ensure high availability of neutrons for user experiments. To facilitate the monitoring of the accelerator at multiple levels, from the control room to executive summary reports, a reporting system was first envisioned in a LabVIEW display [1], [2]. A prototype was developed by Madhan Sundaram, Neutron Science Systems Software Engineer at Oak Ridge National Laboratory (ORNL) using MySQL, Personal Home Page (PHP), Asynchronous JavaScript and Extensive Markup Language (AJAX) scripting. The prototype demonstrated the ability to refine the initial reporting capability of a database using a common web interface for viewing. The prototype was developed further during the summer of 2007 to incorporate Java programming for data acquisition and to interface with an Oracle database cluster recently introduced at SNS. The reporting system continued to be developed at ORNL by Jeff Patton, Software Systems Administrator, and Douglas J. Edwardson, Scientific Systems Programmer, both within the Oracle database and with the modification of the user interface to use Adobe's Flex Technology. The revamped reporting system accesses data stored in the database and implements Java classes incorporating pre-defined and user-defined analytic functions. The reporting system will be the foundation for analyzing data taken throughout the accelerator. It provides a simple and intuitive user interface that is cross-platform and easily accessible for the design, viewing and distribution of reports throughout SNS or outside the SNS facilities (via e-mail).
METHODS

The reporting system consists of five processes or methods, also shown in Figure 1. Each process is defined here. 1) Report Definition is the interface where the user defines the data to be acquired, how to process the data into the desired results and how to format and publish the results, 2) Data Collection is the process that takes the data from the accelerator, 3) Database Storage is the process of storing data within an optimized data storage system to facilitate retrieval and processing of data, 4) Report Processing and Logic is the process that takes the data stored in the Oracle database and transforms it into the results that will be used in the report and 5) Report Delivery is the process that formats the processed data into a report (pdf, image, flex) and delivers as defined (email, database or web server).

The report definitions, the processing functions, or the report formatting can be reused to create another report. The database stores the report definition, the raw data, the results and the final report. The database also links the processes together and is thus central to the reporting system. Figure 2 shows how these five processes have been implemented. The next sections explain in further detail the function of the five processes.

Report Definition

The report definition includes graphical report display choices, published data from the accelerator in the form of a list of process variables (PVs), and a selection of Oracle Structured Query Language (SQL) code. The report logic is also defined in this section and consists of basic pre-defined Java classes, such as simple scalar addition, multiplication or division, and specialized user-defined Java classes. Each report definition is saved in the database and can be used as a template for future reports. Users will be able to select previously defined reports, to modify or re-use in their own report.

Data Collection

Data collected from the accelerator is archived within the Oracle database and used by the reporting system. The data storage uses Oracle record definitions which are already present, both for archiving and reporting, so that an existing data collection mechanism can be used. Also, as data is archived, future reports can recall historical data, thus extending the flexibility of the reporting system. Manipulated or analyzed data is also stored in the Oracle database, allowing future reports to re-use analyzed data.

Database Storage

At the center of the system is the Oracle database. The reporting system takes advantage of the power of the database cluster and is projected to grow along with the archiving needs. Oracle also provides pre-built analytic functions, such as average and standard deviation. In addition to performance, using Oracle as a central storage for all data, raw and processed, allows reports to re-use existing data without duplication. Although SNS chose Oracle because of its availability, the reporting system does not depend on Oracle and can be used with other databases.

Report Processing and Logic

The reusable report logic contains Oracle SQL statements for standard deviation, average, and other statistics reporting functions. It also includes user-defined Java programming language classes that can be chained for execution within a report to further manipulate data. The Oracle SQL statements are stored in the database, while the user-defined Java classes are stored as Java class files. Users can create new logic even when they are not producing reports. The Java logic is stored and made available for any report definition. Upon selection of the Java logic, the report engine will execute the logic for processing the data.

Report Delivery

The report delivery publishes the results defined in the delivery options such as PDF, PNG export, and/or e-mail delivery of the PDF or PNG files. All reports can be viewed through the web.
browser or e-mailed to the user. Links for web viewing can also be delivered via e-mail or used in Microsoft Word documents, PowerPoint presentations, or PDF documents to actively link back to a graph or report. Each report delivery definition made in the Report Definition process will be available to be re-used for new reports, allowing the user to easily define multiple reports and select the distribution methods already defined.

**Reporting System Elements**

The reporting system’s elements, as shown in Figure 2, represent all the technologies integrated to work together. For each report element there is a correlation for the processes described in the previous section.

**Report Definition:** The Watchman report definition element is a web interface where the authenticated user defines the PVs, date range and data summarization for the report. The interface is an intuitive flash web page designed using Flex. Flex implements Adobe’s Flash technology to incorporate a simple, efficient, and elegant interface design. Figure 3 shows the data section of the Watchman user interface, where users select the PVs for their report. Figure 4 shows the chart section of the Watchman user interface, where the users define chart options such as line graph, line color, number of axes, number of series and labels. In the chart section, additional data parameters are also defined, such as date range, data averaging and custom Java functions to be applied to the data prior to charting. The custom Java functions can be chained to implement a sequence of operations and specific order of execution. Figure 5 shows the final page of the report definition where report metadata, such as name, description and publication file type are defined. In the current implementation the publication options are not available.

**Data Collection:** Readings from the sensors located throughout the SNS accelerator facility are published using the Experimental Physics and Industrial Control System (EPICS) [3]. The EPICS front-end computers, which are also known as input-output-controllers (IOCs), provide network access to the data in the form of PVs via the Channel Access (CA) Network Protocol [4]. The IOC configuration determines when sensors are read and when PV updates are published to the network. To allow for network delivery delays, PVs are time-stamped at the IOC, so clients can determine when the data was originally measured. PVs also carry status information that might, for example, indicate sensor read errors. The EPICS Channel Archiver is a CA client program that logs received PV updates (value, time stamp, status and meta information such as engineering units) to provide a data history [5]. While the original archive toolkit uses custom binary data files for performance reasons, current developments store the PV data history in a Relational Database RDB, which facilitates better access for reporting tools. The data collected for the initial reporting system tests were of type float, provided by the IOCs at about 1-second intervals. The SNS accelerator timing is determined by neutron chopper speed and synchronization to the 120 V AC line frequency, which results in a 1.002 second period between most samples.

**Database Storage:** The database uses multiple table definitions and data field definitions within the database known as schemas. Depending on summarized data or raw data needed for the report, the data is stored within the appropriate schema in the Oracle database. The reporting system accesses the raw data from the channel archive schema and makes it available to the data and diagnostics reporting system schema for producing pre-defined reports. Within the data and diagnostics report system schema,
processed or summarized data can be stored for reuse in future reports. Oracle executes statistical functions at the cluster level, making Oracle not only a central storage element of the reporting system, but also a processing element.

**Report Processing and Logic:** The report engine consists of several Java classes that query data from the Oracle database, execute user-defined Java classes, and implement distribution options. The report engine can call the Momentum viewer to display the chart, or create PDF or PNG graphic exports to be e-mailed or stored in the file system. The report engine tier of the reporting system runs on a production web server and is written entirely in Java programming language. It runs continuously and produces reports by launching independently running processes (program threads) which execute report-generating Java programming classes defined by the user.

These Java programming language classes defined by the user are integrated into the report engine. Figure 6 shows a sample user-defined Java class that generates the beam current monitor (BCM) trend analysis report. Other user-defined Java classes can be implemented to create additional customized reports which may then be reused in other reports or chained as part of a list of processing steps to achieve desired data manipulation within the reports.

**Report Delivery:** The Momentum web page is a common Flex-based web page where users can view their reports and charts. In the current version of the Momentum viewer, charts are displayed based only on the defined report logic. The analytic functions and Java logic have already been implemented by the report engine prior to charting the data. All data already processed, as per the user's instructions, is then delivered to the Momentum viewer as a chart. The viewer has the option to export to PDF.

**BCM Trend Analysis Report**

One of the reporting templates, a BCM Trend Analysis report, is designed by graphing raw data collected at 1 Hz intervals and averaged every 60 records. The data collected is the charge of the particle beam as measured by the various BCMs at given times. The data is averaged as denoted as sequence $\{s^k_i\}$, representing a moving average sequence [6]. Given a sequence $\{y_i\}_{i=1}^N$ with $y_i$ being the charge measured at time $t_i$, an $n$-moving average is a new sequence $S^k_0 = \{s^k_i\}_{i=1}^N$ defined from the $y_i$ by taking the average of subsequences of $n$ terms.

$$s^k_i = \frac{1}{n} \sum_{j=0}^{n-1} y_{i+j}$$

Figure 7 shows the decimated average charge for several BCMs as functions of time or multiple sequences $S^k_i$ with $k$ the index identifying a BCM. Any trends in this graph are hard to see because the signals vary widely from one BCM to another. Recalibration of the data is necessary to clearly see the trends.

The BCMs sequences $S^k_0$ are calibrated by taking the average of the sequence from one pre-selected BCM and multiplying each sequence from the other BCMs with this reference average divided by the average of each BCM sequence. These averages are not taken over the whole sequence but just over that part of the sequence where the beam was stable and no modifications were made to the accelerator. The results are new sequences $S^k_m$ with $k$ the BCM index and $R_k$ the scaling factor for BCM $k$

$$S^k_0 = R_k S^K_0$$

where $K$ denotes a pre-selected sequence used as a baseline, $i=1$, the first sequence sample and $M$ the number of sequence samples used for the averaging

$$R_k = \frac{1}{M} \sum \left( S^k_i s^K_i \right)^{-1} S^k_i$$

A decimation of sequences $S^k_m$ is graphed as shown in Figure 8. In contrast to sequences $S^k_m$ that are shown in Figure 7, the trends and variations are easily spotted in Figure 8.
RESULTS

The reports designed thus far have obtained good results that can be correlated to events within the accelerator operations and to configuration changes of instrumentation. Since the reporting system is designed to encompass more than one aspect of reporting for the entire accelerator, three different results are presented here.

Noise

In the chart in Figure 9, the BCMs in the Medium-Energy Beam Transport (MEBT) and High-Energy Beam Transport (HEBT) sections of SNS, MEBT_Diag:BCM02:Q, MEBT_DiagBCM11:Q and HEBT_Diag:BCM32:Q show more noise and larger standard deviation than the rest of the BCMs. The greater standard deviation in the report for the MEBT BCM is probably due to low frequency noise affecting the charge calculation. A different grounding scheme is being investigated to alleviate this problem. The noisy signal in HEBT BCM32 occurs because of its close proximity to the Ring devices. Because of the known grounding and noise issues, BCM02 and BCM11 from the MEBT section of the accelerator and BCM32 from the HEBT section are excluded from the BCM Trend Analysis Report. Thus, grounding and noise issues are easily identified in the report.

Event Correlation

In the data shown in Figure 8, which was collected on June 13, 2008, the transformed BCM Trend Analysis report shows a change in the trend from 3:00 AM to 3:06 AM. The change, as later reported in the Accelerator Electronic Logs (E-log), correlates with an entry explaining the movement of stripping foil position near the I-dump section of the accelerator. Ring_Diag:BCM_D09:Q, RTBT_Diag:BCM02, RTBT_Diag:BCM11:Q, RTBT_Diag:BCM14:Q and RTBT_Diag:BCM25:Q display a difference of approximately 1.0 x 10^-7 during the period of June 13, 3:00 AM to June 13, 4:00 AM.

Systematic Error

In a theoretical model for calibration of BCMs, all graphs should lie within a standard deviation of 1.0 x 10^-7, however, the current graphs have a deviation of approximately 8.0 x 10^-7. This report indicates possible imperfections in the monitors. The results also indicate that the BCM calibration can be improved.

DISCUSSION AND CONCLUSION

The reporting system is the result of a collaboration of technology and individuals. The technologies meld together as the integration of the report extends from EPICS to Oracle and out to the user via a web interface. The integration of ongoing development, such as the RDB Data Archive and Oracle, makes the reporting system possible and will provide further feedback for data archiving needs. For example, as reports are generated and used over a period of time, trends within the reports will provide guidance on these needs. If the reports require only summarized data, then the data acquisition can be further customized to collect at the adjusted interval, or the data can be summarized for long-term archiving.

The purpose of the RDB Data Archive is to archive all data from the accelerator for all defined PVs. The reporting system makes use of this technology, rather than developing a new one, in order to assist the developer in providing feedback and in refining the development of the RDB Data Archive, as well as integrating existing and new technology to be part of the overall reporting system. Though the RDB Data Archive has milestones and goals outside the reporting system, the integration of the technology was the best fit for the reporting system.

The Java logic implementation allows users to specify specific logic for data manipulation. Additional signal filtering routines can be implemented to extend the analysis of the reporting system. The ability to extend the reporting system's analysis tools by writing Java classes gives the reporting system a flexibility and reusability that will grow. An example is the BCM Analysis Report.

As is the case with much software, the results presented here show that the BCM Analysis Report needs to be enhanced. The analysis for Figures 7 and 8 shows a higher overall charge during the time period of 3:00 AM to 3:06 AM. The results of the data
transformation should have reflected this fact. The transformation applied uses the average of the entire series, thus when the Ring and RTBT BCMs were adjusted, the adjustment should be calculated over the period of 3:00 AM to 3:06 AM to get the proper average adjustment. The variation in the report coincides with operator adjustment of the stripping foil in the ring section of the accelerator; therefore a lower overall average charge should be represented after the adjustment. The BCM Transformation Report succeeded in representing an abnormality within the operations of the accelerator; however, an additional parameter in the algorithm is needed to represent data more accurately. The additional parameter needed is to better define the multiplicative factor used in the transformation. Currently it is a ratio of the averages between the reference series and the series to be transformed. The reference series average is taken over the entire dataset and thus reflects an overall average, rather than the section where the discrepancy occurred. Although the transformed chart reported good results for the trend analysis, the additional parameters will allow for the definition of the timeframe for the ratio. Instead of calculating averages over the entire timeframe of the chart, the user will be able to define the averages for only a section of the series’ timeframe. This additional parameter will allow the report to better reflect the true value of the factor to be applied to the calibrations between BCMs. The true value takes into consideration the beam loss changes.

The reporting system is currently used only with BCM data; however, future implementation will include neutron detector data to generate a Contact Radiation Decay Report and Beam Loss Monitor (BLM) data to provide summaries of total beam loss throughout the accelerator.

ACKNOWLEDGMENTS

This research was conducted at the SNS Facilities at ORNL. I thank the U.S. Department of Energy, Office of Science for allowing me to participate in the Science Undergraduate Laboratory Internship program to further my studies and understandings within my field of studies. I would like to extend a special thanks to my mentor, Dr. Willem Blokland, for his guidance, teaching and patient mentoring. I would like to extend my thanks to Dr. Kay Kasemir of the Research Accelerator Division at SNS, Jeff Patton of the Neutron Science Directorate Computing Integration, and Douglas Edwardson of the Neutron Science Directorate Computing Integration for their invaluable support, contribution and assistance.

REFERENCES