

# ROOT を用いた SuperKEKB データアーカイバ用ユーザインタフェースの開発 DEVELOPMENT OF USER INTERFACE FOR SUPERKEKB DATA ARCHIVERS USING ROOT

廣瀬雅哉<sup>#, A)</sup>, 岩崎昌子<sup>B)</sup>  
Masaya Hirose<sup>#, A)</sup>, Masako Iwasaki<sup>B)</sup>

<sup>A)</sup> Kanto Information Service Inc.

<sup>B)</sup> High Energy Accelerator Research Organization (KEK)

## Abstract

We have developed the user interface for SuperKEKB archived data using ROOT. ROOT is an object oriented framework for large scale data analyses such as high energy experiments developed at CERN. The user interface based on ROOT is to access and analyze the archived data, the relational database stored by CSS and KEKBlog, for SuperKEKB. This ROOT based user interface also accesses the EPICS records. This paper describes the design of this user interface tool.

## 1. はじめに

高エネルギー加速器研究機構 (KEK) では、KEKB 電子・陽電子ビーム衝突型加速器を用いた B ファクトリー実験が行われてきた。現在、KEKB 加速器の更なる高輝度化を目的として、SuperKEKB 加速器の建設が進められている<sup>[1]</sup>。

現在、加速器に関する各種データの記録システムとして、CSS Archiver を用いたデータ蓄積システムの開発を進めている。そのユーザインタフェースとして、主に CSS (Control System Studio)<sup>[2]</sup>の使用を検討してきたが、詳細な記録データの解析を行うためには、CSS に各種の機能追加や、ソフトウェア本体の改造が必要であり、それらをユーザ側から行うのは困難であった。そこで、我々は、ROOT<sup>[3]</sup>を用いたデータアーカイバ用ユーザインタフェースの開発を行った。ROOT は CERN で開発された、高エネルギー実験、天文実験等、大型実験向けの解析ツールであり、データの解析機能の開発を、ユーザ自身で行うことができる。

今回開発したツールでは、CSS Archiver で蓄積されたデータの他に KEKBlog<sup>[4]</sup>や EPICS (Experimental Physics and Industrial Control System)<sup>[5]</sup>のレコードデータ等も扱うことができるように、デザインした。これらの詳細について、報告する。

## 2. データアーカイブシステム

KEKB 加速器運転時には、加速器運転に関連する各種データを記録するためのデータアーカイブシステムとして、KEKBlog を運用していた。KEKBlog は、KEK で開発されたファイルベースのデータアーカイブシステムである。また、アーカイブデータのユーザインタフェースとして、LogBrowser が使われてきた。LogBrowser は、KEK で開発された、SAD スクリプトで記述された解析アプリケーションである。LogBrowser は、KEKBlog からデータを取得し

て表示させる機能に加えて、それらのデータを解析することもできる。データ解析機能としては合計値出力、差分、台形・長方形積分、LPF・HPF が用意されている。

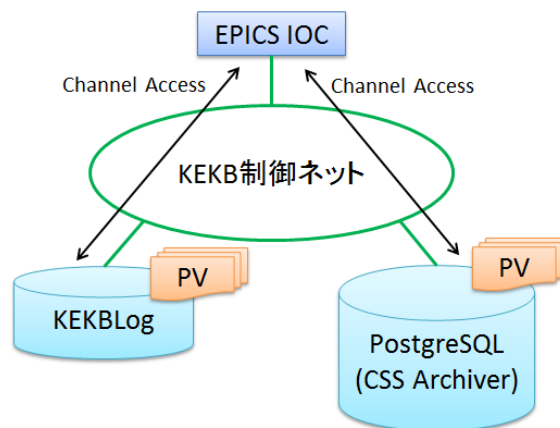


Figure 1: A data archive system of KEKBlog and CSS Archiver.

Archive Engine Group QS\_CRYO

Status: State Enabled

Channel	Connected	Mechanism	Current Value	Last Archived Value	Received Values	Channel Limit	Channel Avg	Channel Max	Capacity	Overrun
QS_CRYO.BEPI	Connected	10.00 sec scan; max. 60 repeats	2015/02/17 14:57:22.306077045 0 OK, OK	2015/07/28 19:47:47.825000000 0 OK, OK	1	0	0.0	1	6	0
QS_CRYO.COEM2	Connected	10.00 sec scan; max. 60 repeats	2015/02/17 14:57:22.306074102 0 OK, OK	2015/07/28 19:47:47.825000000 0 OK, OK	1	0	0.0	1	6	0
QS_CRYO.COEM3	Connected	10.00 sec scan; max. 60 repeats	2015/02/17 14:57:22.306075038 0 OK, OK	2015/07/28 19:47:47.825000000 0 OK, OK	1	0	0.0	1	6	0
QS_CRYO.CNTR10X	Connected	10.00 sec scan; max. 60 repeats	2015/02/17 14:57:21.106420878 -2.5 MINOR, LOW ALARM	2015/07/28 19:47:47.825000000 -2.5 MINOR, LOW ALARM	1	0	0.0	1	6	0
QS_CRYO.CNTR10X	Connected	10.00 sec scan; max. 60 repeats	2015/02/17 14:57:21.106421154 -2.5 MINOR, LOW ALARM	2015/07/28 19:47:47.825000000 -2.5 MINOR, LOW ALARM	1	0	0.0	1	6	0
QS_CRYO.BEPL	Connected	10.00 sec scan; max. 60 repeats	2015/02/17 14:57:22.198112899 0 OK, OK	2015/07/28 19:47:47.825000000 0 OK, OK	1	0	0.0	1	6	0
QS_CRYO.BEPL	Connected	10.00 sec scan; max. 60 repeats	2015/02/17 14:57:22.306078028 0 OK, OK	2015/07/28 19:47:47.825000000 0 OK, OK	1	0	0.0	1	6	0
QS_CRYO.BEPL	Connected	10.00 sec scan; max. 60 repeats	2015/02/17 14:57:22.306078028 0 OK, OK	2015/07/28 19:47:47.825000000 0 OK, OK	1	0	0.0	1	6	0
QS_CRYO.LIBSE1	Connected	10.00 sec scan; max. 60 repeats	2015/02/17 14:57:22.198112770 0 OK, OK	2015/07/28 19:47:47.825000000 0 OK, OK	1	0	0.0	1	6	0
QS_CRYO.LIBSE2	Connected	10.00 sec scan; max. 60 repeats	2015/02/17 14:57:22.306088485 1 OK, OK	2015/07/28 19:47:47.825000001 1 OK, OK	1	0	0.0	1	6	0

Figure 2: Web management screen in CSS Archiver.

一方、SuperKEKBでは、KEKBLogに加えて、開発用として、CSS Archiver を用いたデータ蓄積も行う予定である。ここで、CSS Archiver は、アーカイブデータを RDBMS に保存するが、SuperKEKB では PostgreSQL を採用する予定である。CSS Archiver で蓄積したデータを閲覧するユーザインタフェースとして、CSS が使用可能である。

KEKBLog と CSS Archiver の概念図を図 1 に示す。KEKBLog と CSS Archiver、どちらも KEKB 制御ネットワークに接続されている EPICS IOC 上のレコードに Channel Access (CA) を使用してアクセスし、それぞれデータを取得し、各ストレージに格納する。

また、CSS Archiver は図 2 に示すように、Web 上からアーカイブの動作状況やストレージ情報などを確認することができる。

### 3. アプリケーションの設計と仕様

CSS Archiver ではレコードデータに関する様々な情報を記録しており、それらの情報は CSS で表示することができる。CSS での表示例を図 3 に示す。

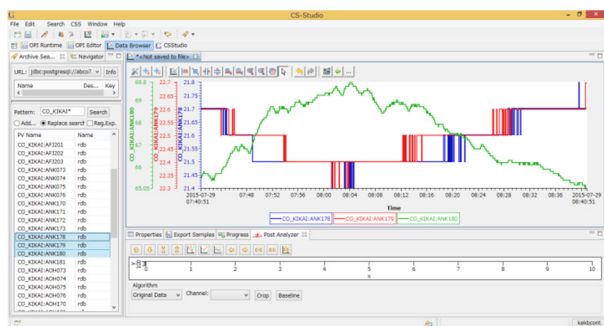


Figure 3: Graphical representation on the CSS screen.

しかし、CSS ではデータを解析する機能としてフィッティングや相関、FFT の 3 種類しか備えられておらず、大規模な解析を行うには不十分である。また、CSS は Eclipse をベースに開発されているため、ユーザが独自に機能を拡張するとなると、開発言語の他にプラグイン作成・追加、ビルド方法など様々な専門知識が必要になるため、開発の負担が大きくなってしまふ。

そこで、我々は高エネルギー実験等のデータ解析を目的に開発された ROOT をフレームワークとして用い、解析機能を充実させたユーザインタフェース「RooArcViewer」を開発した。開発言語は、C 言語を拡張したオブジェクト指向言語である C++を採用している。

本アプリケーションの概念図を図 4 に示す。ROOT がインストールされた Linux マシンから CSS Archiver や KEBBLog のデータを取得し、表示する。この他、EPICS レコードのデータは CA によりデータをモニタリングすることができる。PostgreSQL からのデータ取得には ROOT に用意された TSQLServer クラスを使用し、データを TObject クラスで保持す

るようになっている。TObject で保持しているデータは、ROOT で用意されているクラスを使用することで、グラフやヒストグラム、フィッティングなど様々な解析を行うことができる。

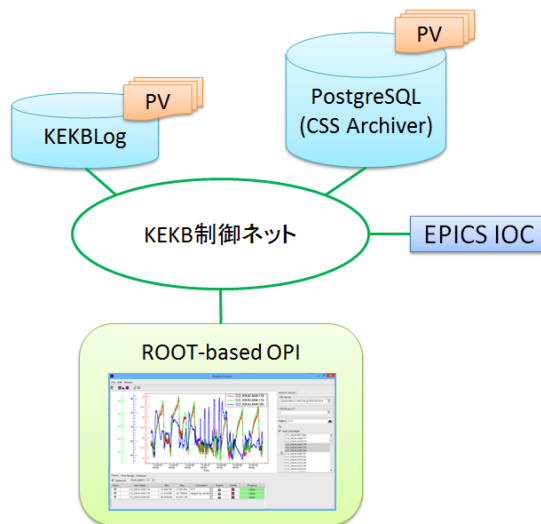


Figure 4: System schematic in ROOT-based OPI.

ROOT のバージョンは、2014 年 5 月からリリースされている 6.0 を採用している。ここで、画面デザインを図 5 に示す。左側にグラフを描画するためのキャンバス、右側にデータのセレクトボックス、下側に表示データの操作ツールを配置するといった、シンプルなレイアウトを採用した。

ユーザは取得先のデータベースと期間を指定し、閲覧したいレコードを選択し、プロットボタンを押下することで、データを見ることができる。ここで、現段階で実装している機能を表 1 に、動作例としてヒストグラム表示の画面を図 6 に示す。

また、キャンバスに描画されたヒストグラムなどを画像 (png、jpg) や PDF、PostScript など様々な形式で保存することができる。また、データファイル (.dat) や ROOT 独自のファイル (.root) などにデータを出力して保存することもできる。

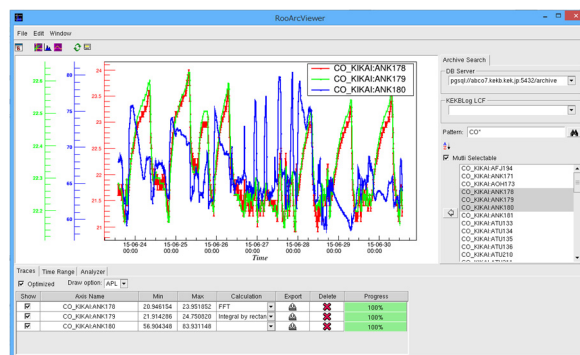


Figure 5: Main screen in RooArcViewer.

Table 1: Loaded Function List

対応データ	CSS Archiver KEKBlog EPICS Record
アナライザ	Histogram Graph Fitting Sum Difference Integral by rectangle and trapezoid FFT Correlation Subtraction Average
保存形式	画像保存 (png, jpg, PDF, PostScript, ...) データ保存 (dat, root)

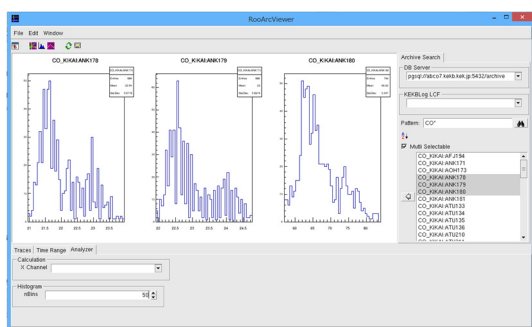


Figure 6: Histogram indication screen in RooArcViewer.

### 3.1. CSS との比較

新たなユーザインタフェースの開発にあたり、様々な視点から CSS と ROOT をベースとした本アプリケーションの比較を行った。

まず、データアーカイバ用ユーザインタフェースとして特に課題となるのがデータ取得速度である。これについては、浮動小数点を扱う単純なデータを用意し、PostgreSQL からデータを取得してから、グラフが表示されるまでの時間を比較した。比較結果を図 7 に示す。CSS は Java、ROOT は C++ と開発言語による違いはあるが、処理速度にさほど差は見られなかった。しかし、CSS は Java 仮想マシン (JVM) 上で実行されるため、メモリチューニング次第では、結果が変わる可能性がある。ただし、処理時間のほとんどは SQL 実行時間である。

次にユーザの拡張性について検討を行った。CSS では前述したように、ユーザ独自の拡張を行うことは難しい。一方、ROOT は C++ インタプリタ (CINT) が採用されており、C++ をスクリプト言語として使用することができるため、ユーザが C++ を理解していれば、ROOT の機能を拡張することもできる。

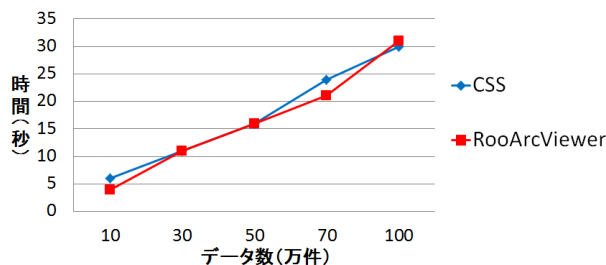


Figure 7: Comparison of the data acquisition rate of RooArcViewer and CSS.

ただし、C++ でプログラミングを行う場合は、Java に比べてやや注意が必要である。特に注意すべき点はメモリアクセスである。プログラミングを行う際、Java ではガベージコレクション (GC) が実装されているため、特にメモリを意識する必要はないが、C++ ではメモリ管理をプログラマ自身が定義しなければならないため、メモリリークやメモリ破壊等が起きやすい。

## 4. ROOT による解析

ROOT は、1995 年頃から CERN で開発されている、高エネルギー物理学実験のためのデータ解析フレームワークである。当初は開発言語として FORTRAN が採用されていたが、実験が大規模化するにつれソフトウェアの開発手法も変化し、C++ を用いたオブジェクト指向型のデータ解析環境が求められるようになった。ここでは、ROOT でデータ解析を行う場合の手法について簡易的に述べる。

ROOT には、データを扱うための基礎クラスとして、グラフ (TGraph) やヒストグラム (TH1) などが、あらかじめ用意されている。これらは全てオブジェクトクラス (TObject) を継承している。オブジェクトを描くための基礎となる「ホワイトボード」としてキャンバスクラス (TCavans) が用意されている。ここで、キャンバスにグラフを描画した例を図 8 に示す。

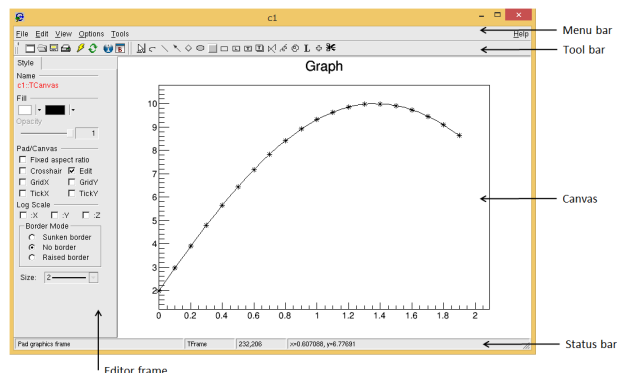


Figure 8: Drawn canvas in ROOT.

オブジェクトクラスはツリークラス (TTree や

TNtuple 等)として保持することができる。これにより、大量のデータをツリー構造で管理することができるため、作業の効率化を図ることができる。また、ツリークラスは ROOT の独自ファイル形式で保存することができ、こうすることで他の ROOT プログラムでも扱うことが容易である。このファイルは ROOT ブラウザ (TBrowser) で開くことができ、ヒストグラムなどを表示できる。その実行例を図 9 に示す。

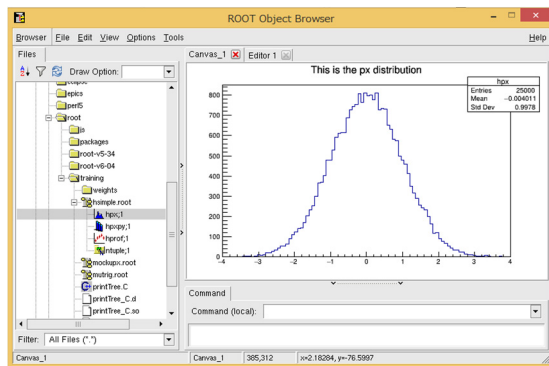


Figure 9: Indication of a ROOT file in TBrowser.

## 5. まとめと今後の予定

我々は、SuperKEKB データアーカイバ用ユーザインタフェースを、ROOT を用いて開発を行った。このソフトウェアにより、CSS では不十分であった解析機能の充実を図ることが可能となった。現在、操作性や安定性、機能性の面での向上を図っている。

また、CSS Archiver において、アーカイブデータの保存先に RDBMS を採用しているが、NoSQL データベースである Cassandra<sup>[6]</sup>を採用することも検討している。そのため、これらの対応についても検討を行う予定である。

## 参考文献

- [1] K. Akai, et al., "Design Progress and Construction Status of SuperKEKB", Proc. of IPAC12, pp. 1822-1824 (2012); <http://accelconf.web.cern.ch/AccelConf/IPAC2012/papers/uppr006.pdf>
- [2] CSS, <http://www.aps.anl.gov/epics/eclipse/>
- [3] ROOT, <https://root.cern.ch/drupal/>
- [4] T. T. Nakamura et al., "Data Archiving System in KEKB Accelerators Control System", Proc of the 10th ICALEPCS, Geneva, Oct. 10-14 2005, [https://accelconf.web.cern.ch/accelconf/ica05/proceedings/pdf/P1\\_077.pdf](https://accelconf.web.cern.ch/accelconf/ica05/proceedings/pdf/P1_077.pdf)
- [5] EPICS, <http://www.aps.anl.gov/epics/>
- [6] Cassandra, <http://cassandra.apache.org/>