

Development of User Interface for Accelerator Components Database

Kazuhiko Watanabe^{1,A)}, Hiroshi Yoshikawa^{B)}, Hironao Sakaki^{B)},
Hiroki Takahashi^{B)}, Masato Kawase^{B)}, Makoto Sugimoto^{C)}

^{A)} Total Support System Corporation

3-10-11 Funaishikawaekinishi, Toukai, Naka, Ibaraki, 319-1116

^{B)} Japan Atomic Energy Research Institute

2-4 Sirane, Sirakata, Toukai, Naka, Ibaraki, 319-1195

^{C)} Mitsubishi Electric Control Software Corporation Ltd

1-1-2 Wadasaki, Hyougo-ku, Koube, 652-8555

Abstract

In J-PARC, we manage the accelerator devices data using the relational database. It is used for managing physical value of all devices. We request the database to generate EPICS records, and the managed data must always be the latest. Some data is treated by operator who doesn't have enough experience of database input. If they feel the input is difficult, the input of the database stagnates. In the worst case, the database might not be used.

This paper reports the database GUI for operator who doesn't have enough experience of database.

加速器コンポーネントデータベースのユーザーインターフェイス開発

1. はじめに

J-PARCでは機器の物理情報、及び制御情報をデータベース（以下DBとする）により管理することが決定し、その構築が行われている。このDBは、加速器構成機器のデバイス値を物理値に変換するパラメータなどの管理や、EPICSレコードの自動生成などJ-PARC制御のコアとなる部分で、常に最新の正確なデータに更新され続ける必要がある。

このDBで管理されているデータの更新は各装置の担当者が行うため、DBに関する知識・経験を十分に持たない人もこのDBを操作することになる。このような背景から、装置担当者がDBの知識を持たなくても容易に利用可能なグラフィカルユーザーインターフェイス（以下GUIとする）がなければ、装置担当者がDBを活用することを敬遠するようになり、最悪の場合は全く活用されない状態に陥るであろう。

本報告は、DBの使用経験が乏しい装置担当者が効率的に活用できるDBとなることを目的としたGUIの作成に関するものである。

2. 設計

使いやすいインターフェイスとするためには、まず、わかりやすいものでなければならない。

そのために、以下の点に留意する。

- ① 一度に表示する操作項目は必要最小限とする。
- ② 操作の階層は浅く。
- ③ 少ない操作で目的を果たす。

これらを実現するために、まず、DBに対して行わ

れる操作の抽出・整理を行う。

2.1 DB操作の抽出

まず、今回のGUIに関して、データの編集作業は行わないことを前提とした。理由は以下の2点である。

- ① DBのテーブル構成がまだ明確に固まっていないため、テーブル構成の変更によって編集画面もその都度変更になってしまう恐れがある。
- ② 現在、機器や制御信号の情報はEXCELファイル形式で収集しているが、今後もしばらくはEXCELファイルでの情報収集がメインとなると予想され、データの更新はファイルのインポートのみとした方が、都合が良い。

データの編集作業は制御グループが指定したフォーマットのEXCELファイルにより行うものとし、編集機能は省略する。

編集機能が省略されたことにより、DBへのデータ追加・更新はすべてEXCELファイルから入力することになる。J-PARCは信号だけでも数万点あると思われる、これを人の手によりDBに入力していくのは非常に非効率的である。このため、EXCELファイルを直接インポートする機能が必要である。

次に、DBのGUIである以上、DBに入力されたデータを閲覧する機能は欠かせない。このとき、ユーザーは、多くの場合すべてのデータが見たいわけではなく、任意の条件により絞り込んだデータを必要としているはずである。そこで、希望する条件によ

¹ E-mail: kaz_wa@linac.tokai.jaeri.go.jp

りフィルタリングし、それに関連するデータの中で必要な項目だけを取り出せる機能が必要となる。

また、ここで取り出したデータは、そのままではデータベースへの入力順となっているので、並び替える必要がある。並び替えはフィルタリングにより取り出された項目によるソート以外に、機器の配置順でのソートができることが望まれる。

ここまでで、ユーザーが希望するデータの表がGUI上に表示されるが、ユーザーの多くは閲覧するだけではなく、データのチェック、編集、印刷など、別の目的がある。そこでこの表をオフラインでも利用できるように、EXCELファイル形式でエクスポートする機能も必要である。

よって、操作として必要な項目は、インポート、フィルタ、表示項目選別、ソート、エクスポートということになる。

このうち、フィルタに関しては、その条件を設定するため、設定画面が必要となる。この設定画面はフィルタボタンをクリック後、サブウィンドウとして表示することにする。設定されたフィルタ条件はウィンドウ上方に表示する。

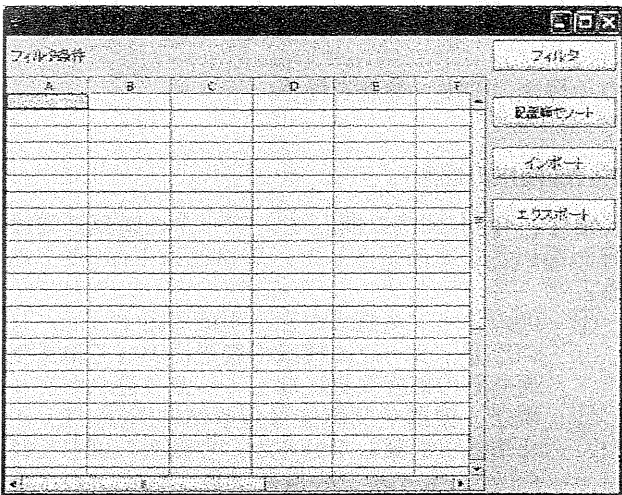


図1. メインウィンドウ

図1は実際に作成したメインウィンドウである。メニューは、実際に右手でボタンを押すことイメージしたとき、押しやすい位置となる右側に配置し、操作性を考慮して、大き目のボタンにした。

2.2 利用価値を高める機能

ここまでは単純に操作画面の使いやすさについて述べたものであるが、DBの操作画面としてより重要なものは

- ① データのリレーションを活用し、リンクをたどることで目的のデータへの到達時間を短縮する。
- ② 出力されたファイルが利用価値の高いものであること。

という点である。これらの目的にあったフィルタ及びソート方法を

実装する必要がある。

2.2.1 ソート

ユーザーがソートをかける理由の多くは、以下のようなものと推測される。

- ① 表にある機器の物理的配置順を知りたい。
- ② 表にある機器の中で同一種類の機器をまとめ、データに抜け・誤りがないかチェックしたい。

よって、必要なソート処理は、配置によるソートと表の各項目によるソートということになる。

項目によるソートは、ユーザーの方々がEXCELの操作に十分慣れているということもあり、EXCELでのソートと同様にカラムのヘッダをクリックすることで昇順、降順、ソートなしを切り替えられるようにした。

2.2.2 フィルタ

ユーザーから要求されるフィルタは、絞込みに使用したい項目がDBの項目のほぼすべてになる可能性があり、条件設定の画面を工夫しなければ項目を選択するだけでも大変な労力となりうる。

そこで、まずは、フィルタに何を求めているのか、整理する。

フィルタに使用する項目は大きくハードウェア情報とソフトウェア情報に分けられる。

まずはハードウェア情報について考える。

フィルタリングに使用する項目は物理的エリア、グループ、加速構造、機器タイプ、直列に接続される機器のグループ、ケーブル配線ルート、制御盤等が収納されるラックが考えられる。そこで、ハードウェア情報からフィルタリングに使用する項目は上記のものに限定する。

次にソフト情報の中で、フィルタリングに使用する項目は、データ形式、EPICSかNon-EPICSか、IOC、コントローラタイプ (PLC、VME、CAMAC、EMB-LANなど) が考えられる。ソフトウェア情報も上記の項目に限定し、これ以外の項目はフィルタ項目から除外する。

これらの項目と、AND、OR、NOTを使って条件を設定しフィルタをかけることで、希望するデータを選別できるようにする。実際のフィルタ条件設定ウィンドウを図2に示す。

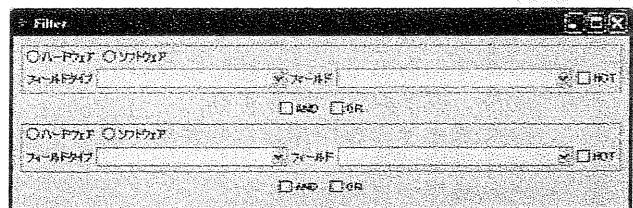


図2. フィルタ条件設定ウィンドウ

また、図3のように、セルを選択し右クリックすることで、そのセルの内容によってフィルタリングできる簡易フィルタも用意する。

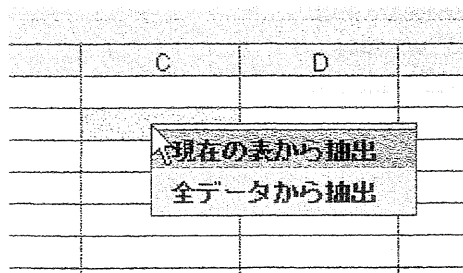


図3. 簡易フィルタ

2.2.3 表示項目選別

フィルタリングされた項目に関連付けられた項目は多岐におよび、すべてを表示すると非常に見辛い表になってしまう。そこで、表示する項目を選別し、ユーザーが必要とするシンプルな表を出力できるようにする。このとき、指定された項目が存在しない機器が混在していることも考えられる。このような場合は「Not Use」が入るようにし、すべての機器に対し表示項目をそろえる。

2.2.4 インポート

データの入力は先にも述べたとおり、EXCELファイルを、インポート機能を使用して読み込み、DBに入力する。このEXCELファイルのフォーマットはJ-PARC制御グループが用意したものを使用する。

2.2.5 エクスポート

フィルタやソートを使用し要求する情報をウィンドウに表示させた後、EXCELファイルとして取り出す。エクスポート後、EXCELファイルの生成に使用したフィルタ条件は、再度利用する可能性がある。そこでエクスポートする際も、表の先頭にフィルタの条件を付加する。

3. 開発環境

GUI開発には、開発言語としてJAVAを選択した。J-PARCでは、運転用コンソールや、の多くはLinuxを使用したものになると思われる。しかし、機器情報の入力や、編集、資料作成などはWindows上で行われる。よって、どちらのプラットフォームからでも同様にDBにアクセスできる必要がある。JAVAを使用してGUI作成を行えば、まったく同じ操作性をWindowsにもLinuxにも提供できる。

DBには無料でありながら高機能であるPostgreSQLを使用した。

また、JAVAプログラム中のDBにアクセスする部分

にはHibernateというオブジェクト/リレーショナルマッピングツールを使用することにより、オブジェクト指向というJAVAの利点を損なうことなくプログラミングすることができた。

4. まとめ

機器、および信号の情報が集まるに従い、入れるべき情報の種類も増え、DBの完成度を上げるため、テーブルの構造は度々変更せざるを得なくなる。本報告のようなGUIならばテーブルの変更をプログラムで隠蔽することで、ユーザーは変更点を意識することなく操作することができる。

しかしながら、このままでは、DBの恩恵を十分に享受しているとは言い難い。

現在でも、DBに入力されているデータは二次元の表一つで表現するには無理があるものが多い。この問題を解決するにはすべてのテーブル構造を完成させ、それらにスムーズにアクセスでき、なおかつ、J-PARCの機器情報管理という業務に最適な構造を持ったGUIを用意する必要がある。

今後は、さらなる機器情報の分析を進め、効率の良いテーブル構造の検討と、J-PARCの運営に欠かせないツールとなりうるGUIの設計を行っていく。