

PLAN TO DEVELOP OVERALL STATUS SCREENS FOR J-PARC ACCELERATOR COMPLEX

Norihiko Kamikubota^{#,A)}, Noboru Yamamoto^{A)}, Nobuhiro Kikuzawa^{B)}, Masato Kawase^{B)}, Masatoshi Adachi^{C)}, Masahiko Tanaka^{C)}, Susumu Yoshida^{D)}

A) J-PARC Center / KEK (Tokai campus), 2-4 Shirakata Shirane, Tokai-mura, Ibaraki, Japan, 319-1195

B) J-PARC Center / JAEA, 2-4 Shirakata Shirane, Tokai-mura, Ibaraki, Japan, 319-1195

C) Mitsubishi Electric System and Service Co., Ltd., 2-8-8, Umezono, Tsukuba, Japan, 305-0045

D) Kanto Information Service (KIS), 8-21 Bunkyo, Tsuchiura, Ibaraki, Japan, 300-0045

Abstract

J-PARC LINAC started beam operation in Nov., 2008, followed by J-PARC MR in May 2008. Recent operations have been aiming for stable beam delivery to experimental facilities: MLF, Hadron, and Neutrino. Operators control J-PARC accelerators using GUI applications. Due to historical reasons, current GUI applications are not optimized for operators. This report discusses that what ideal GUI applications for J-PARC operators are.

オペレータ視点のJ-PARC 加速器統合画面開発プラン

1. はじめに

J-PARC加速器は、平成18年11月にリニアック (LINAC) に初めてビームを入射し、平成19年10月に3GeVシンクロトロン (RCS)、平成20年5月に主リング (MR) のビーム運転を始めた。現在は、実験施設 (MLF, HD, NU) へのビーム安定供給期になってきた^[1]。



コントロール室のPC端末(オペレータ用画面)

図1 J-PARC 中央制御棟コントロール室

J-PARC加速器は、オペレータが中央制御棟のコントロール室のPC端末で起動するプログラム (GUI画面) を操作することで運転している (図1)。オペレータがGUI画面で加速器を運転するにあたり、現在以下の問題がある。

- (1) LINAC, RCS, MRのGUI画面は別々のグループが開発した。画面のLook-and-Feelはばらばらで、起動可能な端末も分かれている。オペレータは、分散した複数の端末で多数の画面を操作しなければならない。
- (2) (主にMR) 建設期に機器グループが専門的調整のために開発した制御画面が、そのままオペ

レータにも使用されている。ビーム供給運転に不要な細かすぎる情報、誤操作で加速器が止まり得る危険なボタンがある。オペレータは、わかりにくい画面で慎重な操作が求められる。

- (3) 加速器が停止した時どこに問題が生じたか、あるいは停止に至る前の警告レベルの情報を、オペレータや加速器スタッフに知らせる統一的なシステム (Alarm画面) がない。

本稿では、オペレータが使用する加速器GUI画面について議論する。第2章では現状を述べ、第3章ではオペレータ視点からどういう画面・システムが望ましいか、さらに第4章では開発すべき加速器統合画面の位置づけや整備計算機環境について検討する。

2. オペレータ用加速器画面の現状

2.1 3 J-PARC加速器のGUI画面

加速器 (LINAC, RCS, MR) の制御システムは、異なる設計思想の基に別々のグループが開発した。ただし、いずれもEPICS^[3]ベースの制御システムである。開発の時期も数年のずれがある。GUI画面の開発ツールや動作端末の情報を表1に示す^[2]。

表1: 3 加速器のGUI画面の環境

	LINAC/JAEA	RCS/JAEA	MR/KEK
基本	Java, medm	Java, medm	medm, edm
高機能アプリ	XAL (Java)	SAD	SAD, python, ROOT
端末	PC (RHEL)	PC (RHEL)	Thin Client

LINAC、RCSの基本GUI画面の開発はJAEAが担当した。Java言語での開発を主とし、一部でEPICS標準ツールであるmedmを使用した。高機能アプリケーション

ション (HLA = High Level Application) 環境には、LINACは米国ORNLの加速器SNSで開発されたXAL^[9]、RCSはKEKBからSADを導入している。RCSのアプリケーションは[5]も参照されたい。LINAC、RCSとも、GUIアプリケーションはある程度決まったPC端末 (RHEL linux) で少数のみ起動することが前提になっている。

MRはKEKが開発を担当したが、基本GUI画面の開発にはEPICS標準のmedmまたはedmを、HLAにはSADを採用した。基本的に先行したEPICS制御システムであるKEKBの資産・経験を利用している。ただし、HLAにはPythonとROOTも使用されている。GUIアプリケーションは集中管理されたサーバ計算機で起動され、約20式のThin Client端末が表示機 (X window terminal) として使用されている^[4]。任意のアプリはどの端末でも表示可能で、また同じアプリが複数動作・表示されることも多々ある。

2.2 AlarmシステムとArchiveシステム

加速器が停止した時、停止情報はMPSシステムに頼っている¹。ただしMPS画面は3加速器それぞれ別なので、オペレータは少なくとも3つの画面をにらみながら対応することになる。現状では、実験ユーザなどコントロール室に居ない人が加速器停止の詳細情報を知ることは困難である。停止原因などの情報はオペレータが電子ログに手作業で書き込むが、実験ユーザはその記述を参照していると思われる。また、警告レベルの情報は、多数分散した個別の機器GUI画面を見て判断するしかない状況にある。

加速器機器の状態を記録するArchiveシステムも、JAEA (LINAC, RCS) と KEK (MR) で別個に存在して相互乗り入れは出来ない。JAEAはRDB-baseで独自開発したもの、KEKはEPICS標準のchannel archiverを元に多少改修したものである^[6]。

3. 加速器統合画面の構成の検討

3.1 基本方針

過去数年のJ-PARC加速器運転経験に基づき、オペレータ視点からどういう画面・システムが望ましいか検討しよう。

- (1) 個別の機器の詳細を知る必要は無い。各機器 component が加速器運転モードに対応した「正常状態」と確認できればよい。機器数が多い場合は、集約して数を減らす²。
- (2) 運転の定型操作に対応した操作メニューを用意す

る。設定値の直接代入は不適切である。操作手順が煩雑な場合は、処理Scriptを開発してClickで起動するようにする。

- (3) 運転に必要な十分な情報を整理した形で表示する。全加速器の全画面に於いて共通のlook and feel、共通の表現ルール (緑はOK、赤はNG、青は休止、など) を用いる。
- (4) 加速器のupgradeに応じた改修が容易であること。機器不調など一時的に特殊な状況で運転する場合³に対応できること。
- (5) どんな画面があるか、理解しやすいような階層化・整理を行う (詳細は次節)。

3.2 階層化された加速器統合画面の構成案

オペレータが加速器状態を把握し操作する「加速器統合画面」について、基本方針を勘案しながら具体的に考えてみる。

- (1) 画面は階層化して整理する。最上位 (TOP階層) はごく少数の全体画面、次 (ACC/CO階層) に各加速器 (LINAC, RCS, MR) および制御 (CO) の代表となる画面を置く。その下の層 (DEV階層) には、各加速器の機器 component それぞれの代表画面を置く。
- (2) 【TOP階層】 運転全体画面とAlarm全体画面の2つを作成する。前者は運転状態全般を示す現存の「Beam Destinations and MR」画面のイメージ、後者はAlarmがどこで発生しているかを示す画面 (現在は存在しない) である。この階層の画面は表示のみで、操作機能は無い。
- (3) 【ACC/CO階層】 各加速器 (LINAC, RCS, MR) の代表画面は、電源情報と共にビームや運転情報も同じ画面に表示する。また、下位層の機器 component を代表する状態が簡潔に表示する。制御 (CO) の画面には、Timingや計算機情報など、加速器横断的な制御情報を表示する。
- (4) 【DEV階層】 各加速器画面の下位に、個別の機器 component 画面を作成する。例えば、Magnet / RF / Monitor / VAC など。また、制御のDEV階層は、Timing、PPS、計算機 (server, ioc)、ネットワーク、などの画面を整備する。
- (5) TOP階層のAlarm全体画面の下位層として、各加速器 (LINAC, RCS, MR) および制御 (CO) のAlarm画面を置く。これらはオペレータが知るべき警告や異常イベント (時系列表示) を表示する。

以上の思想に基づいて統合画面の全体構成を俯瞰したものを図2に、またTOP階層2画面のイメージを図3に示す。

¹ PPS (人的安全システム) による停止の場合を除く。

² 例えばMR-BLMは220台あるが、350BT, INS-A, ARC-A, などと集約する、など。

³ 例えばノイズ誤動作するBLMの一時的マスク、など。

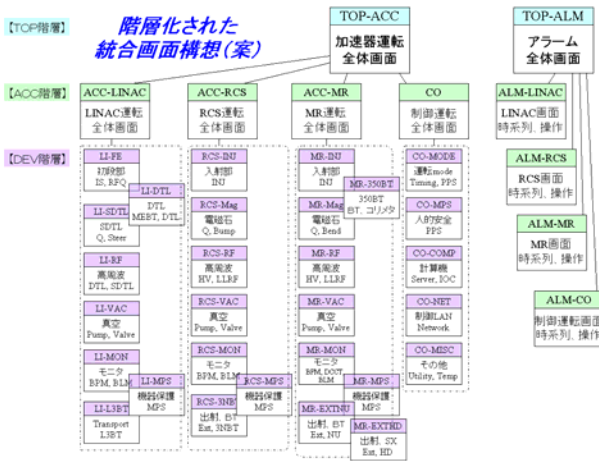


図 2：階層化された加速器統合画面の全体構想

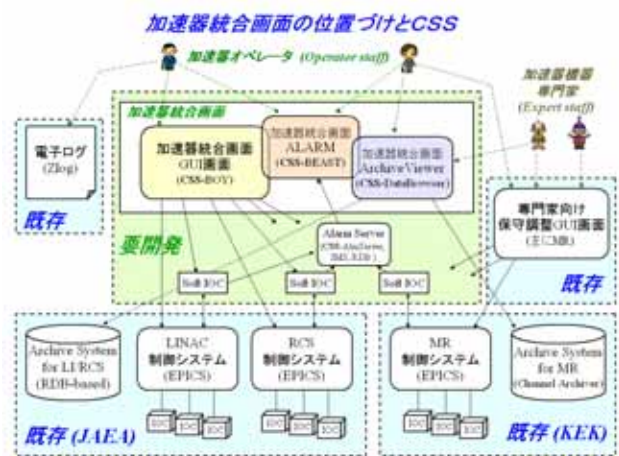


図 4：加速器統合画面の位置づけとCSS

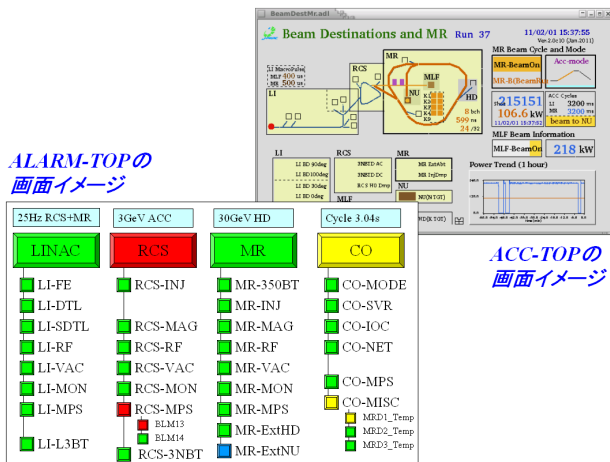


図 3：TOP階層の画面イメージ

4. 加速器統合画面システムの位置づけ・整備計算機環境

4.1 加速器統合画面システムと既存制御システム

J-PARC加速器制御システムは、加速器毎に別グループが開発した。加速器統合画面システムは、既存の制御システムに手を入れるのではなく、その上位の新システムとして構築するのが現実的と考えられる。この場合、既存の制御システムとの間の整合のためのソフト階層が必要であるが、制御システムはいずれもEPICSベースシステムなので、EPICS Soft IOC として実装できる。

加速器統合画面システムと既存制御システムの全体像を図4に示す。オペレータ支援という観点から、図4には既存のArchive Systemと電子ログシステム(通称Zlog)との関連も示す。なお、(オペレータ向けでは無い) 機器専門家向けの保守調整画面は、図4の右手に位置づけられる。既存の多数の画面がそれにあたると目される。

4.2 CSS (Control-System Studio)の評価

加速器統合GUI画面開発のplatformとしては、EPICS での標準的なツールであるEDM (GUIエディタ) やALH (Alarmシステム) などが考えられる。しかし近年EPICS communityでは、CSS (Control-System Studio) の活発な活動がある[7]。

CSSは単独のツールではなく、複数のツールをJava Eclipse Platformの中で整備することで、ツール間の連携を容易にすることを狙っている。本件では、CSSのツールのうちSNSで採用されている BOY (GUIエディタ)、BEAST (Alarmシステム)、Data-Browse (trend-graph or archive viewer)、などに注目している。図4で示すように、加速器統合画面で新規開発が必要な部分はCSSツールとよく整合している。目論見どおりに進めば、1) BEASTで異常/警告検出、2) BOYで現在情報の確認、3) Data-Browserで過去archiveの調査、4) 電子ログへの記録、という一連の作業が、J-PARC加速器オペレータのPC端末での定型化された操作で可能になるであろう。一方CSSはJavaベースで従来のツールより「重い」ので、特に表示能力や処理速度等が我々の目的に十分かどうか注意しなければならない。

平成23年7月、CSS開発者の一人がKEKに短期滞在中であり、KEKおよびJ-PARCの制御グループ共同でCSSの試験的導入と評価を進めている[8]。例として、CSS Data-BrowserをJ-PARC MR Archiveシステムに接続し読み出した履歴グラフを示す(図5)。

CSS Data-BrowserによるMR加速器履歴データの表示

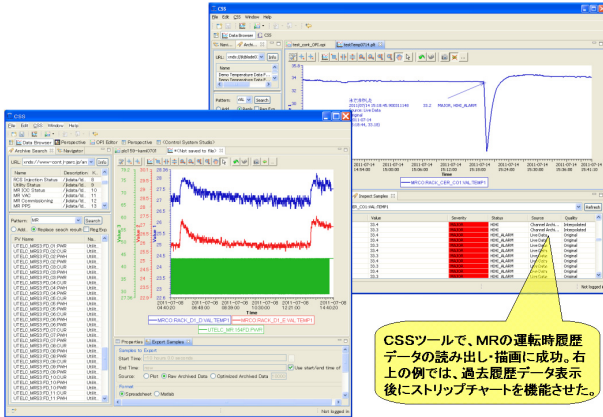


図 5： CSS Data-Browserの例

5. まとめと謝辞

J-PARC加速器のオペレータ用運転画面について、現状の問題点を指摘し、オペレータ視点の加速器統合画面を開発するための全体像を検討した。また、CSSの評価を進めている。今後詳細な画面設計を進め、向こう1年程度での開発・整備をめざしたい。

原稿を書くにあたってご意見いただいた皆様に感謝いたします。

参考文献

- [1] 長谷川和夫 他、J-PARC加速器の現状、第8回加速器学会 (this meeting)
- [2] N.Kamikubota, "J-PARC status", a presentation at EPICS Collaboration meeting in Shanghai, March 2008
- [3] <http://www.aps.anl.gov/epics/index.php>
- [4] 吉田奨 他、J-PARC制御へのThin Client端末の導入、第4回加速器学会、和光、平成19年8月、p.381
上窪田紀彦 他、J-PARC MR運転用計算機の利用環境整備、第7回加速器学会、姫路、平成22年8月、p.690
- [5] 川瀬雅人 他、3GeVシンクロトロン加速器統合制御システム開発プラン、第8回加速器学会 (this meeting)
- [6] 飯塚上夫 他、J-PARC MR加速器アーカイブデータ表示システムの構築、第6回加速器学会、東海村、平成21年8月、p530
飯塚上夫 他、J-PARC MR archive dataの高度利用、第8回加速器学会 (this meeting)
- [7] <http://cs-studio.sourceforge.net/>
or <http://www-linac.kek.jp/cont/css/index.html>
- [8] Kay Kasemir, SNS ; Kazuro Furukawa, KEK ; Takashi Obina, KEK ; private communication
- [9] <http://xaldev.sourceforge.net/>