SAD in Accelerator Operation and Virtual Accelerator

From a Viewpoint of Controls

Kazuro Furukawa, KEK
<kazuro.furukawa@kek.jp>
at J-PARC Commissioning Group Meeting
On Oct.30.2002.

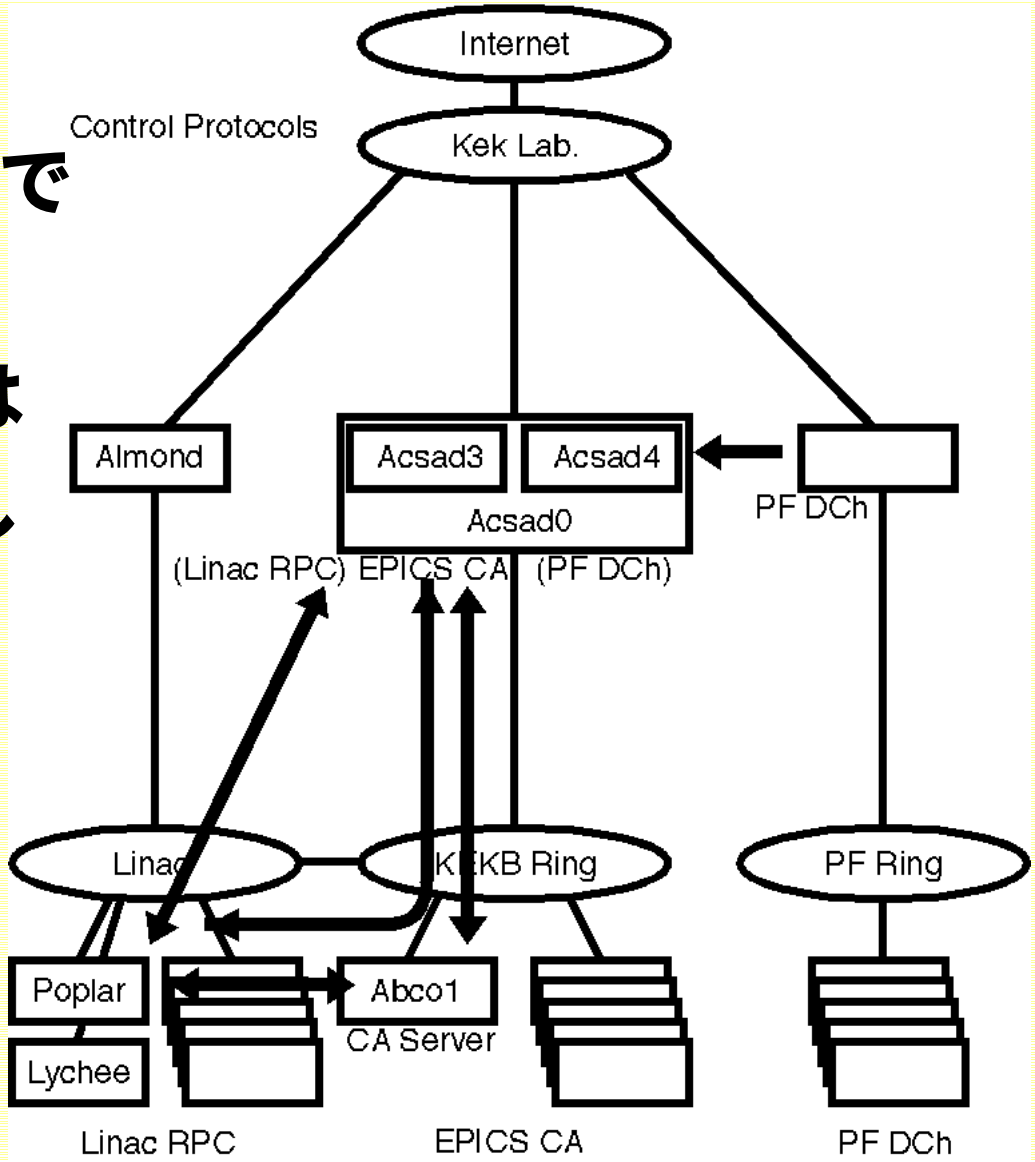
Contents

- Some Background in KEKB
- Script Languages
- Virtual Accelerator

KEKB Ring Linac Controls

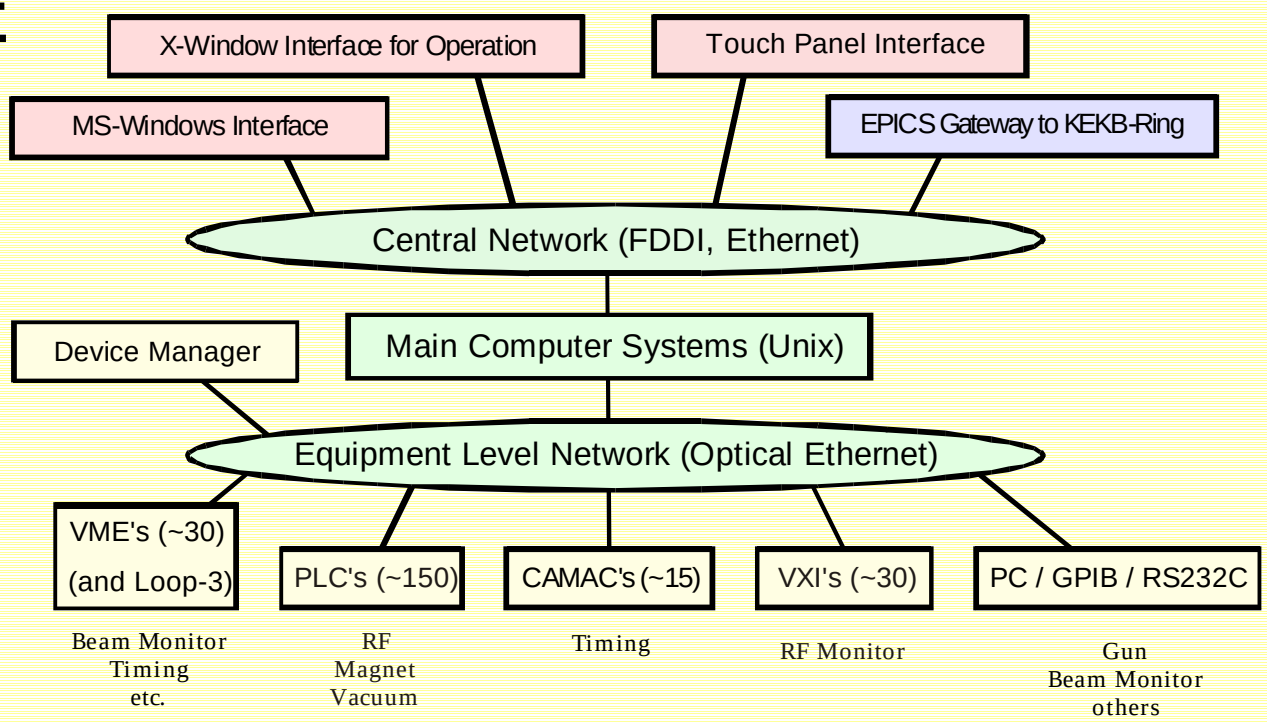
- **異なる制御** System **を同様の** Operator Tool **で統合**
- Ring **と** Linac **の間には複数の仕組みが混在し柔軟に対処**

(**今後** EPICS CA **を増やす方向**)



Linac Controls

- Remote-Procedure-Call (RPC) **を基礎にした** Client - Server System
  - 10 **年前に** Home-Grown
    - **機器** Level **は様々な** Hardware **の組み合わせ**
    - **上位は機器から独立**
    - EPICS Gateway
    - **国際/業界標準**
    - **多階層構成**



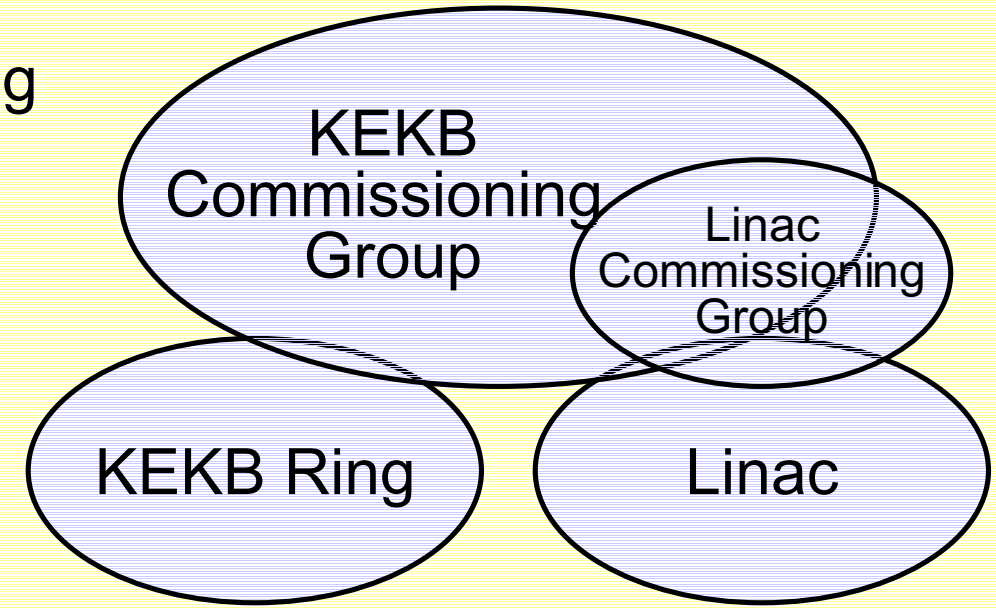
KEKB Ring EPICS Controls

- EPICS の開発と表示等単純 Application は abco1
  - 1 台の HP-UX、PF-AR 用の abco2 が Backup
  - 主に Medm を使用し開発が早い（機器担当者の試験に便利）
  - Logic は IOC Database に載せて実行
    - あまり頻繁に変更しない
- 通常 Operation Software は acsad/alsad
  - 3 台の Tru64/Alpha、3 台の Linux/Intel、及び Macintosh など
  - SAD/Tk を主に使用し、Algorithm も Client 側で開発
    - 運転中も変更を行うことがある
- これらの他に Python/Tk も用いられる
- 加速器側 EPICS Server は IOC/VME 約 100 台
  - その下に CAMAC, ArcNet, GPIB, Serial など

KEKB Commissioning Groups

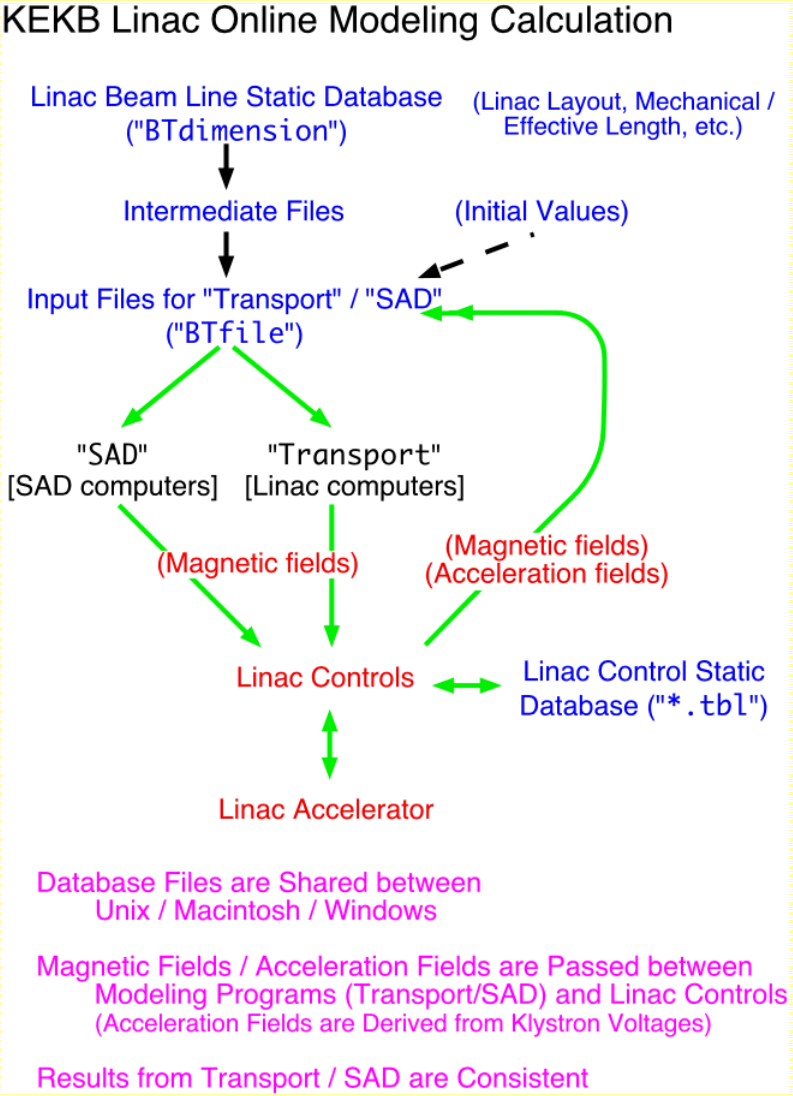
◆Commissioning Group (KCG) の構成

- ✜Linac Commissioning (LCG)
  - ♦Linac から約 6 人
  - ♦Ring から約 10 人
- ✜KEKB Ring Commissioning
  - ♦LCG 全員
  - ♦さらに Ring から約 20 人
- ✜Linac Commissioning の過程で Software が形作られた (1997~)
  - ♦SAD + Tk + ...



Linac Beamline Geometry

- Linac Beamline Geometry Conversion
- In MAD Format
  - Generate Every 5-minutes
  - With Realtime Magnetic and Electric Field Information
- SAD do not Read MAD Data Directly
  - SAD Reads Linac Info. Realtime
- May Need to Link
  - Input Data Conversion Routine (?)



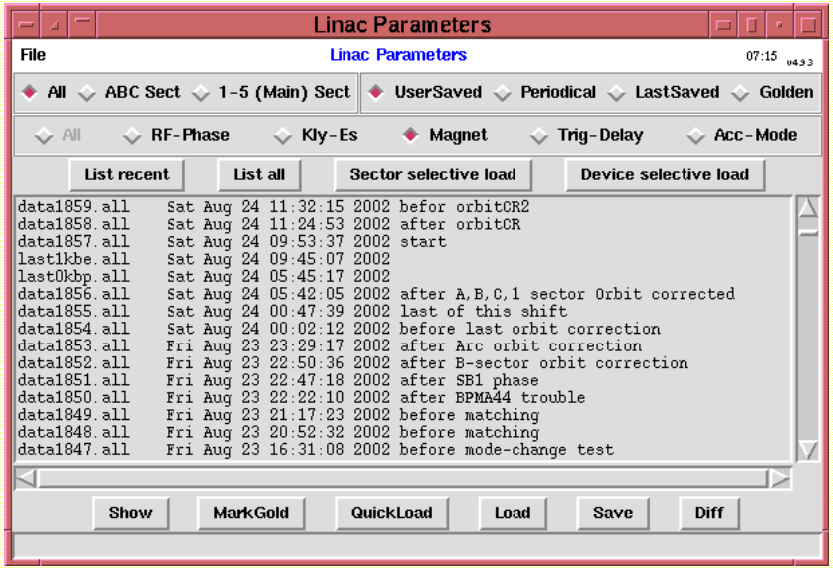
Compiler and Scripting Languages

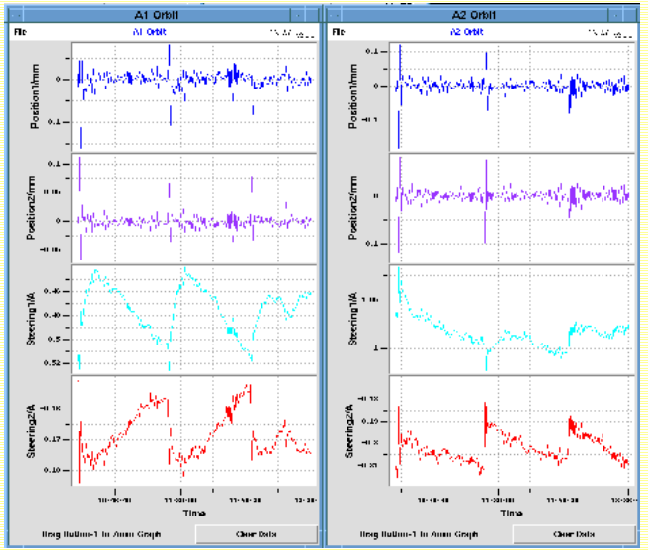
- **基礎となる高速処理**
  - EPICS Database **間の** Link **と** Compiler **言語で開発**
    - Compiler **言語は** EPICS **の場合主に** C **が使われている**
- **運転用** Application
  - Rapid-Cycle Development
    - Idea **をすぐに実装できる環境が必要**
- Script (Interpreter) **言語が重要**
  - KEKB **では** SAD/Tk **と** Python/Tk、Linac **では加えて** Tcl/Tk
    - —Tcl/Tk**は** 10 **年前から、他は** KEKB **から**
    - List **処理、** Data **処理機能（統計、** Fit、FFT、...）
    - Graph Plot、**標準的な** GUI Widget、Database、Archiver
    - **外部** Task **との通信**
    - **その上** SAD **は加速器を知っている**

Tcl/Tk

- Tk Widget の Native Language
- 多少 Minor だが初心者にも使いやすい
- EPICS 業界でいくつかの Lab. が使用
  - 簡単な例
    - Button を表示して shell command 実行を対応付ける

```
Pack [button .b -text Measure -command  exec beamscan ]
```





SADScript

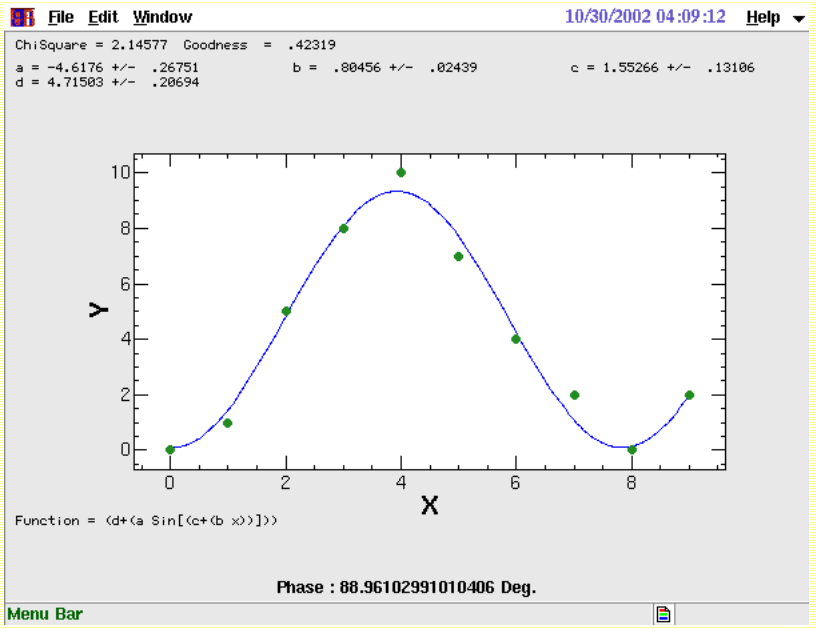
- Mathematica-like Language
  - Symbolic Manipulation **はない（ので速い）**
  - EPICS CA (Synchronous **及び** Asynchronous)

    CaRead/CaWrite[ ], CaMonitor[ ], etc.
  - Oracle Database
  - Tk Widget
  - Canvas Draw and Plot
  - KBFrame on top of Tk
  - Data **処理機能**（Fit、FFT、...）
  - Process **間通信**（Exec, Pipe）

    System[ ], OpenRead/Write[ ], BidirectionalPipe[ ], etc.
  - Greek Letter **を簡単に表示できる**
  - KEKB **では加速器知識が必要無いものにも多数用いられる**

SADScript

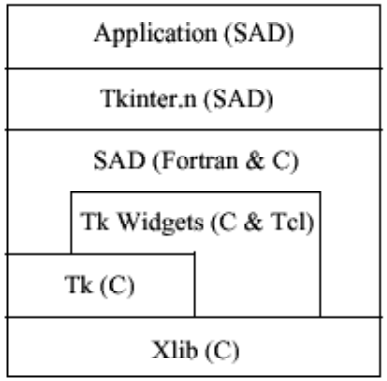
◆Example




```
FFS;
w=KBMainFrame["w1",fm,Title->"t1"];
$DisplayFunction=CanvasDrawer;
W1=Frame[fm];
c1=Canvas[w1,Width->600,Height->400,
 Side->"top"];
Canvas$Widget=c1;
data = {{0,0}, {1,1}, {2,5}, {3,8}, {4,10}, {5,7}, {6,4}, {7,2}, {8,0}, {9,2}}
fit = FitPlot[data,a Sin[x b + c] + d, x, {a,5},{b,1},{c,1},{d,5},
 FrameLabel->{"X","Y"}];
phase = StringJoin["Phase : ", (c/.fit[[1]]) 180/Pi, " Deg."];
f1=KBFComponentFrame[w1,Add->{KBFText[Text->phase]}];
TkWait[];
Exit[];
```

SADscript の情報

- SAD Web Page
  - 生出氏、他
  - 加速器関連
- SAD Tkinter Doc
  - 生出氏
  - SADscript 関連
  - Tk 関連



DRAFT

SAD/Tkinter の使い方

生出勝宣
KEK, Oho, Tsukuba, Ibaraki 305, Japan
oide@acsad1.kek.jp

1997 年 11 月 5 日
(SAD1.0.5.6b に対応)
rev.1

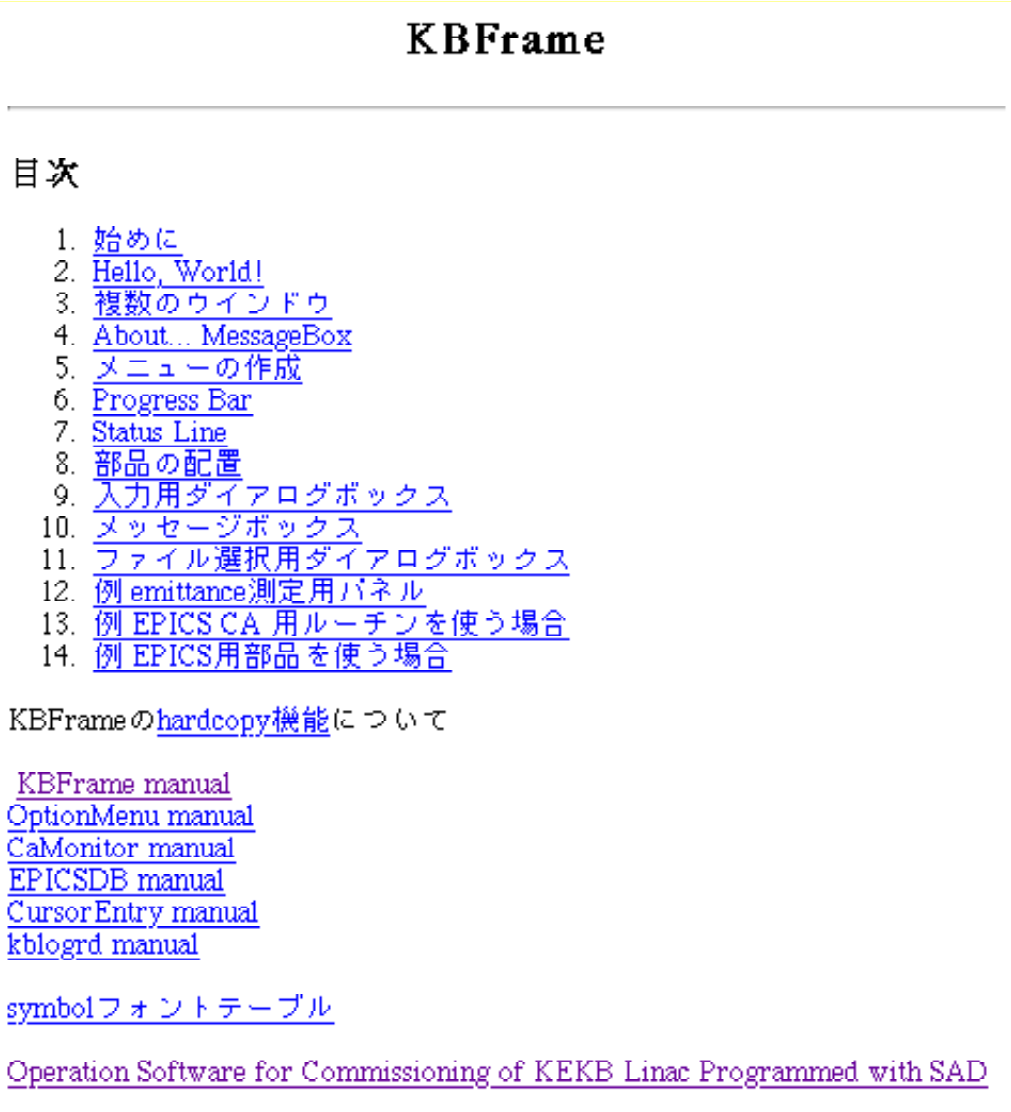
SAD は加速器の設計コードとして 1986 年から KEK で開発され現在に至っています (その概要はホームページ `http://www-acc-theory.kek.jp/SAD/sad.html` を参照)。最近では EPICS チャネル・アクセスや Python/Tkinter、Tcl/Tk インタープリータなどが組み込まれ、SADScript インタープリータ言語とあわせて、単に加速器の設計やシミュレーションに限らず、汎用のシステムとして利用可能なものになりつつあります。SAD/Tkinter は SAD/FFS/SADScript インタープリータから、X ウインドウのアプリケーションを書く道具、Tk tool kit を使うためのライブラリです。

このマニュアルに書かれている内容、SAD のプログラム及びライブラリは、今後予告なく随時改編されます。このマニュアルの最新版は上記のホームページからいつでもダウンロードできます。

なお、このマニュアルは SAD/Tkinter のすべてをカヴァーできていません。その理由は筆者自身がその全機能を経験・把握していないためです。そこで読者の皆様には Brent Welch: *Practical Programming in Tcl and Tk, 1995, for Tcl7.4 and Tk4.0* を併読されることをお願いします。このマニュアルに説明がなくても、Tcl/Tk に備わっている機能は必ず利用可能です。

SADscript の情報

- KBFrame Web Page
  - 赤坂氏、他
  - SAD/Tk の上に
    KEKB Operation 向けの
    統一された
    Look-and-Feel を実現



KBFrame

目次

1. 始めに
2. Hello, World!
3. 複数のウインドウ
4. About... MessageBox
5. メニューの作成
6. Progress Bar
7. Status Line
8. 部品の配置
9. 入力用ダイアログボックス
10. メッセージボックス
11. ファイル選択用ダイアログボックス
12. 例 emittance測定用パネル
13. 例 EPICS CA 用ルーチンを使う場合
14. 例 EPICS用部品を使う場合

KBFrameのhardcopy機能について

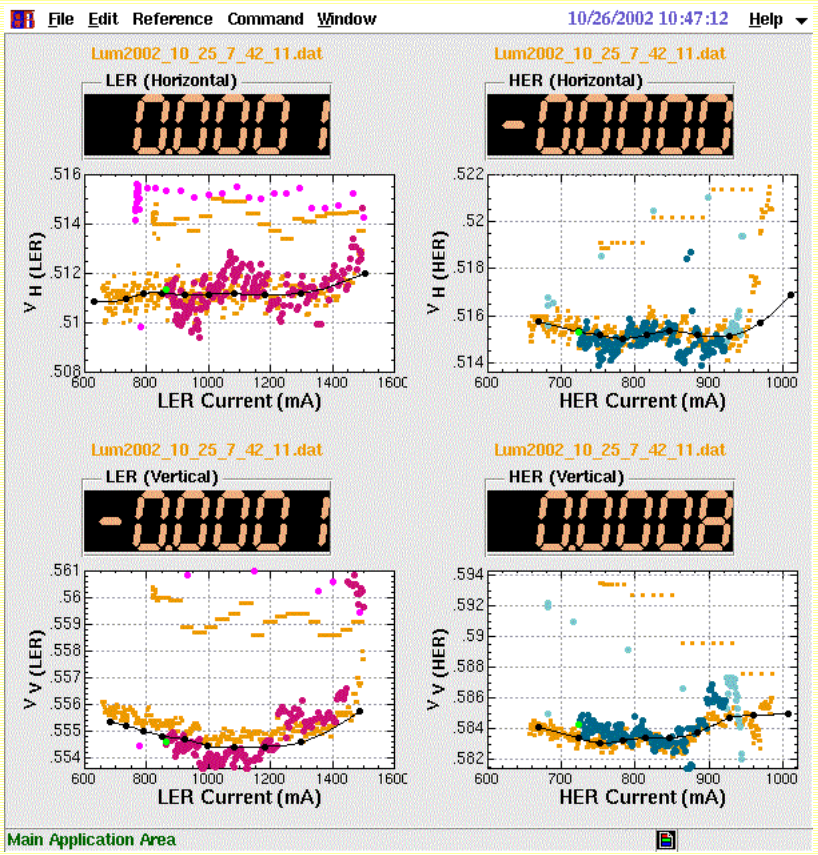
KBFrame manual
OptionMenu manual
CaMonitor manual
EPICSDB manual
CursorEntry manual
kblogrd manual

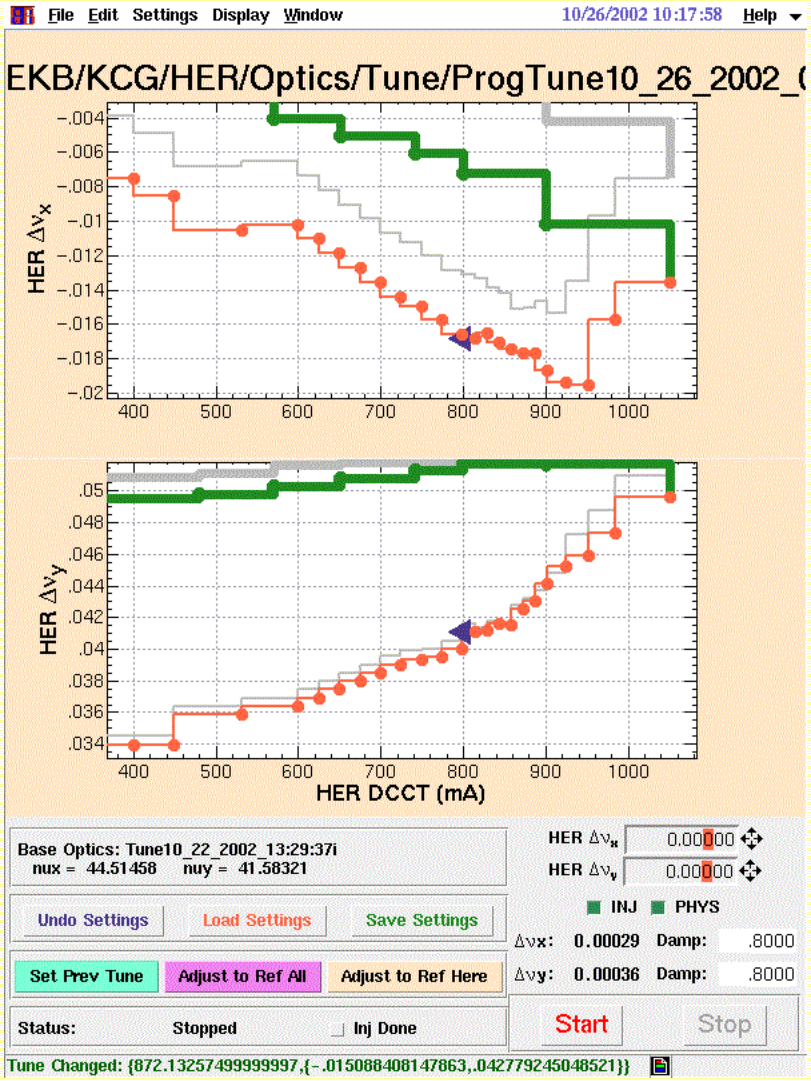
symbolフォントテーブル

Operation Software for Commissioning of KEKB Linac Programmed with SAD

KEKB Operation Panel Examples

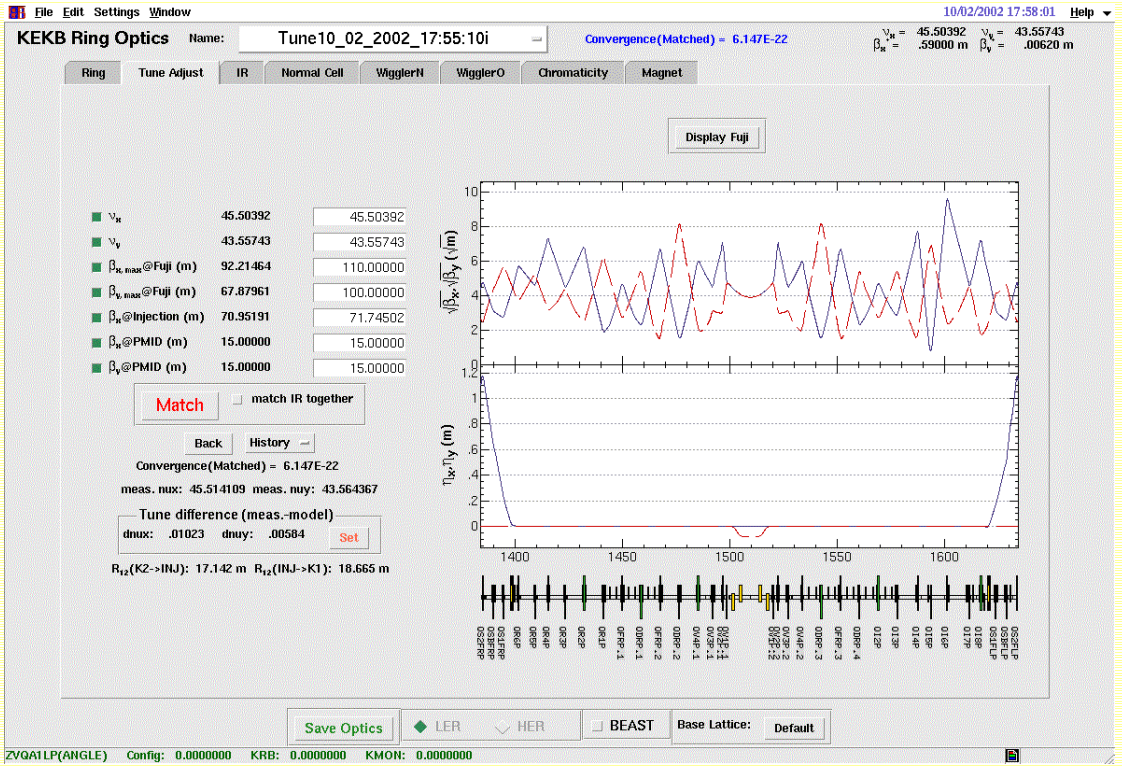
- Tune Measurement and Tune Changer





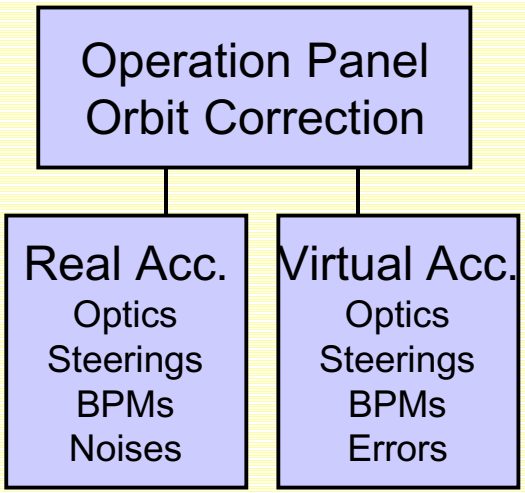
Virtual Accelerator in KEKB

- Tune/Optics Server
  - Keep A Model of Real Accelerator
  - Can Change Tune, Chromaticity, etc, on Request by Other Panels
  - Act as a Virtual Accelerator



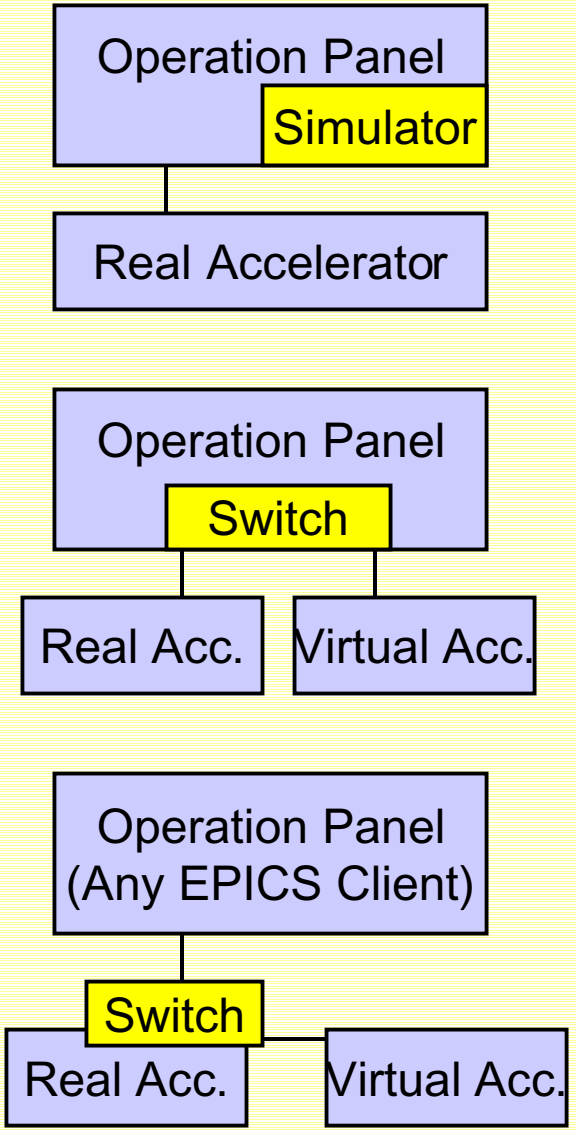
Example Virtual Accelerator

- Virtual Accelerator may Provide
  the Both Fake Steerings and Fake BPMs
  Maybe with Simulated Errors/Noises
- Orbit Correction Application may Work
  On Those Fake Information



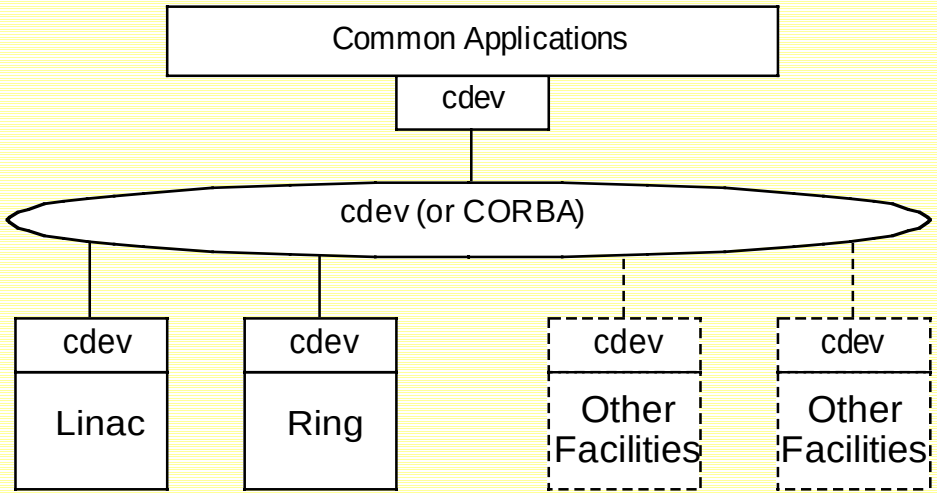
Virtual Accelerator with EPICS

- Fake Accelerator Implementation
  - With EPICS Channel Access
  - In A Single SAD Application
    - Built-in Simulator in Operation Panel
    - Only SAD Applications
  - Separate Simulator (Virtual Accelerator)
    - Needs Some Switching Mechanism
  - Separate Simulator (Virtual Accelerator)
    - In EPICS Semantics (EPICS Simulation Server)
    - Any Operation Panel (not only SAD)
    - SAD Simulation Server should Act as EPICS Channel Access Server



Virtual Accelerator

- Other Implementation Possibilities
  - Upper Level Protocol Like
    - CORBA
      - —Used in Several Lab.
    - Cdev
      - —May be Used in LHC (?)
  - May Cover Systems Not On EPICS
  - Not Covered in This Talk



EPICS Simulation Mode

EPICS Database - Simulation Mode

A set of fields to support simulation are supplied on all hardware input records.

SIMM = YES makes this record a simulation record.

A link to a database value to put the record into simulation mode is specified in the SIML. A non-zero number puts the record into simulation mode.

SVAL is the engineering units value used when the record is in simulation mode.

SIOL is a database location from which to fetch SVAL when the record is in simulation mode.

SIMS is the alarm severity of the record if it is in simulation mode.

- That is, EPICS Records may have Proxy Records

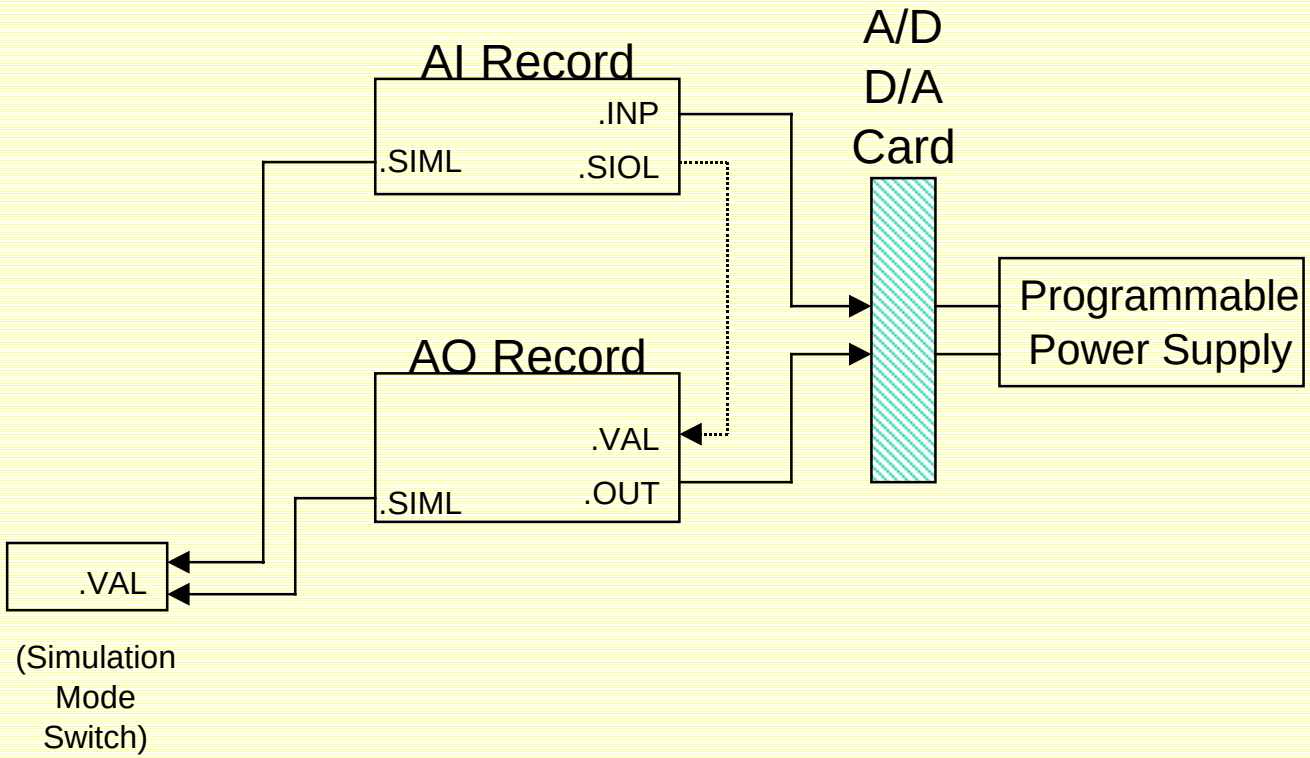
EPICS Simulation Mode

- SIMM - Simulation Mode
  - This field has either the value YES or NO. By setting this field to YES, the record can be switched into simulation mode of operation. While in simulation mode, input will be obtained from SIOL instead of INP.
- SIML - Simulation Mode Location
  - This field can be a constant, a database link, or a channel access link. If SIML is a database or channel access link, then SIMM is read from SIML. If SIML is a constant link then SIMM is initialized with the constant value but can be changed via dbPuts.
- SVAL - Simulation Value
  - This is the record's input value, in engineering units, when the record is switched into simulation mode, i.e. when SIMM is set to YES.
- SIOL - Simulation Value Location
  - This field can be a constant, a database link, or a channel access link. If SIOL is a database or channel access link, then SVAL is read from SIOL. If SIOL is a constant link then SVAL is initialized with the constant value but can be changed via dbPuts.
- SIMS - Simulation Mode Alarm Severity
  - When this record is in simulation mode, it will be put into alarm with this severity and a status of SIMM.

Simulation Mode

EPICS Simulation Mode Simple Example

- Tests Logic without Hardware

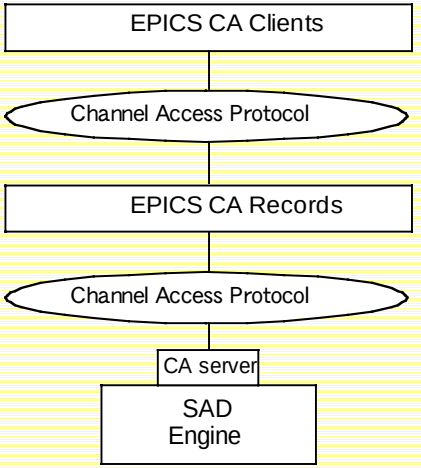


SAD as EPICS Simulator

- Implementing a Virtual Accelerator
- SAD Simulator in Channel Access Server
  - Serves Channel Values Requested by Channels (Records)
    in Simulation Mode (SIOL),
    Acting as a Channel Access Server
  - Slightly more Difficult to Implement (at the First Stage)
- SAD Simulator in Channel Access Client
  - Provides Channel Values Needed by Channels (Records)
    in Simulation Mode (SVAL)
  - Easier to Implement (?)
    - Needs Some Studies

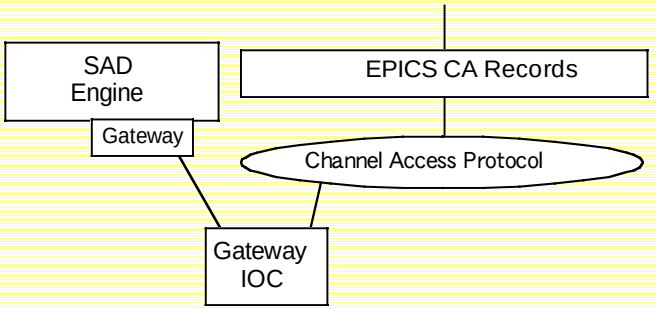
Channel Access Server

- Using Portable Channel Access Server
  - Needs Interface from Server-side Channel Access Library to SAD
    - Written in C++



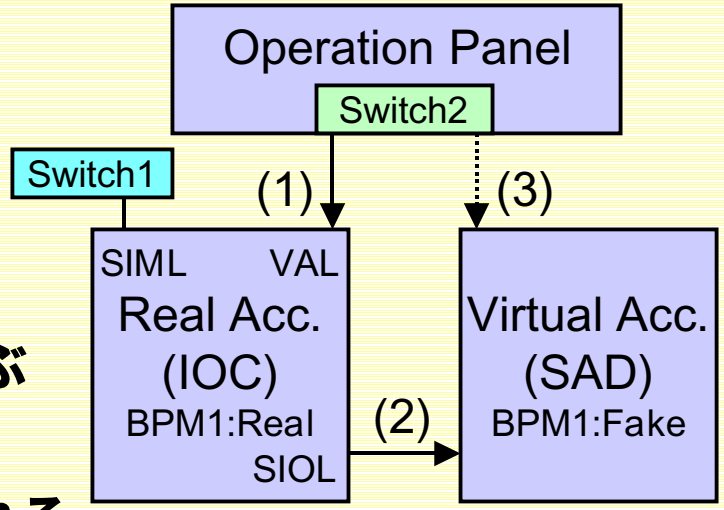
- Using IOC Core of EPICS-3.14 (or Later)
  - Needs Device Support for SAD
    - Maybe Easier

- Using Intermediate Soft Records
  - SAD may act as an EPICS Client
    - Maybe Easier



EPICS/SAD Simulator の使い方

- (1) **通常運転時**
  - **実加速器の** BPM1:Real **に対して**
    普通に EPICS **読み出しを行う**
- (2) **建設時、** Maintenance **時**
  - Switch1 **を** On **して** Simulation-Mode**を選ぶ**
    BPM1:Real **を経由して**
    仮想加速器の BPM1:Fake **が読み出される**



Switch1 (**これも** 1 **つの** EPICS Record) **が** On **であること以外は何も変わらないので、どんな** Operation (Client) Software **にも有効**

- (3) **運転中の** Simulation

Operation Panel **になんらかの方法で組み込まれた** Switch2 **を切り替え、運転に使われている** BPM1:Real **の代わりに** BPM1:Fake **を読み出す**

Operation Panel Software **を書き換えて** Switch2 **を作る必要があるが、仮想加速器自体は** (2) **と同じものが使える**

（この Page **は** Talk **後に追加）**

Near Future

- Implement Simulator Mechanism ?
- Collaboration Between Commissioning Group
- And Controls Group is Important
- ...
  - (Commissioning Experiences at Electron Linac)