

TOF Offline

Address/Geometry Object

- Motivation
- TOF Address Object Class
- TOF Geometry Object Class
- Status&Plan

Akio Kiyomichi

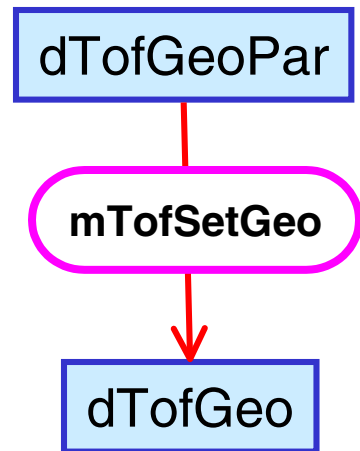
Univ. of Tsukuba

Apr.28, 2000

Old ... Address table & Geometry table

- STAF based parameter tables
 - dTofGeoPar.idl (Panel position)

```
struct dTofGeoPar {
  float rslat;      /* Slat R-position from panel front surface */
  float slat_width; /* Slat width */
  float scintz[3];  /* Slat local-Z position */
  float rpos[11];   /* TOF Panel R-position */
  float phi[11];    /* TOF Panel phi-position */
  float zpos[11];   /* TOF Panel Z-position */
};
```



- dTofGeo.idl (Slat position)

- Created by mTofSetGeo

```
struct dTofGeo {
  short slatid; /* Slat ID */
  short sector; /* Sector (0-1) */
  short side;   /* South=0,North=1*/
  short panel; /* Panel (0-3) */
  short slat;   /* Slat (0-95) */
  float pos[3]; /* TOF slat position */
  float r;      /* TOF slat r-position */
  float phi;    /* TOF slat phi-position */
};
```

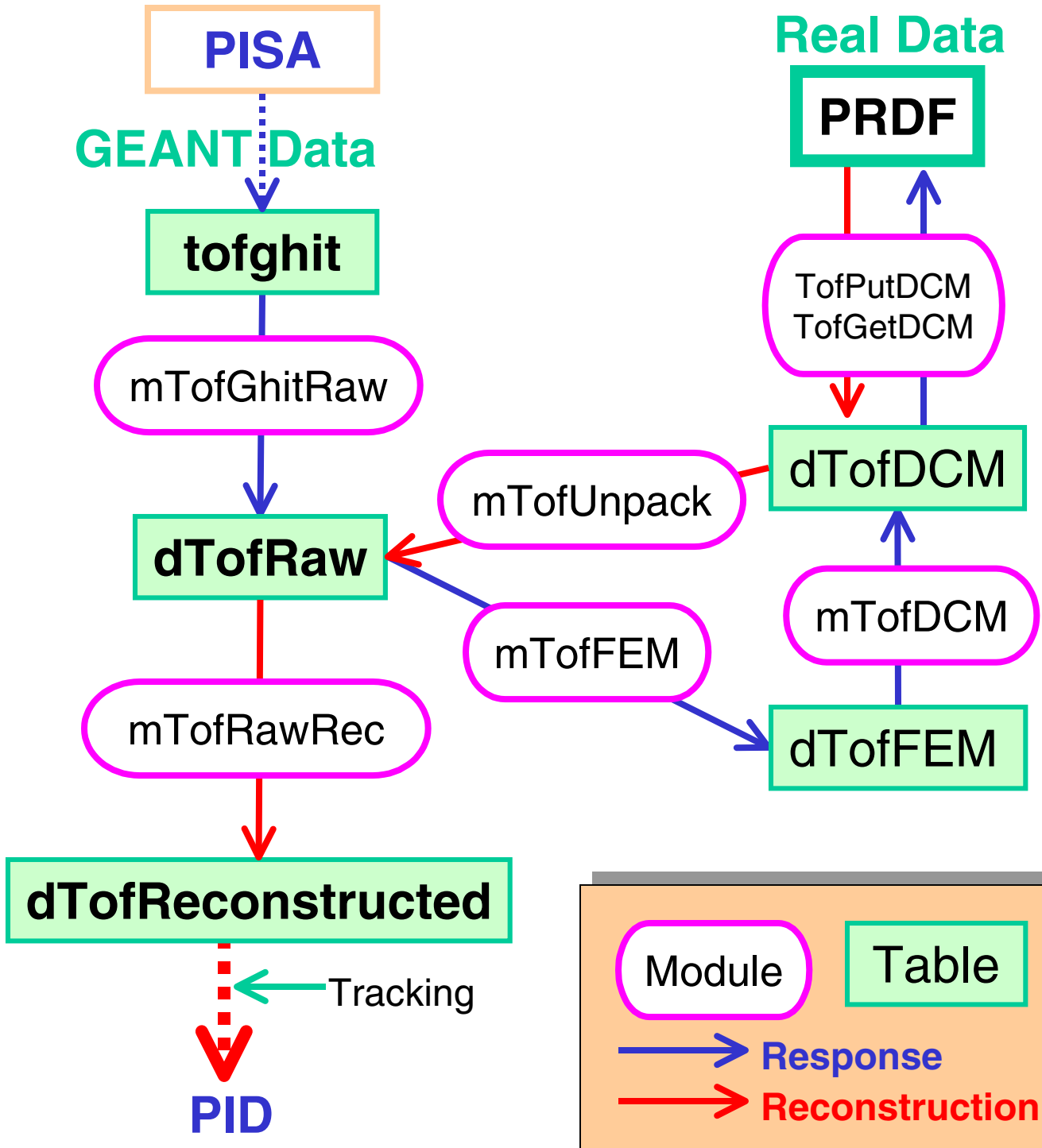
- dTofFEMmap.idl (FEM address map)

- Created by mTofSetFEMmap

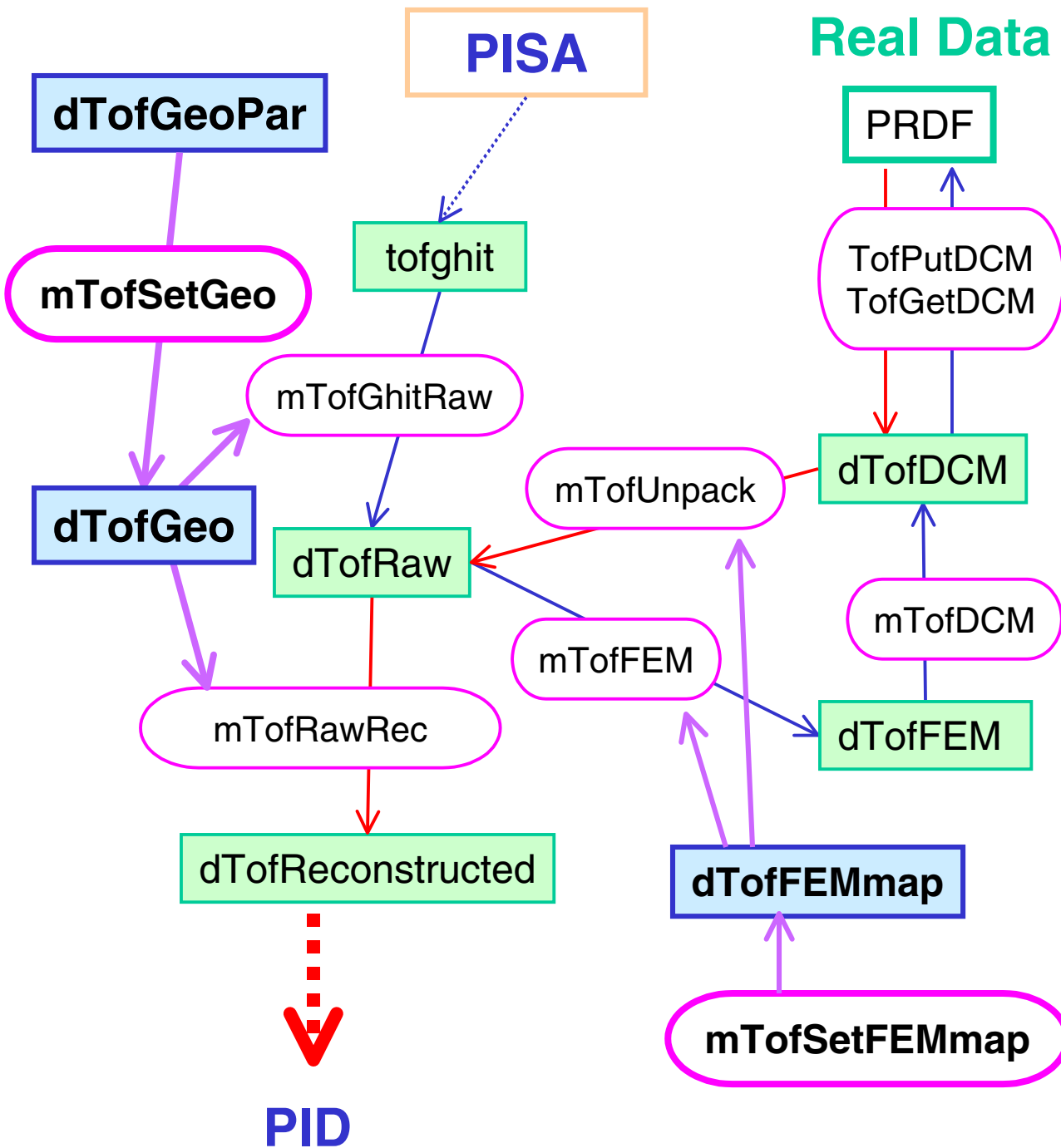
```
struct dTofFEMmap {
  short slatid; /* slat sequential number */
  short crate; /* FEM crate ID (0-7) */
  short slot;   /* FEM slot ID (0-15) */
  short ch[2]; /* FEM channel ch[Bot=0/Top=1] */
};
```

Slatid	crate	slot	Ch[0]	Ch[1]
0	2	3	15	14
1	2	3	13	12
2	2	3	11	10

Offline Data Flow (w/Simulation)



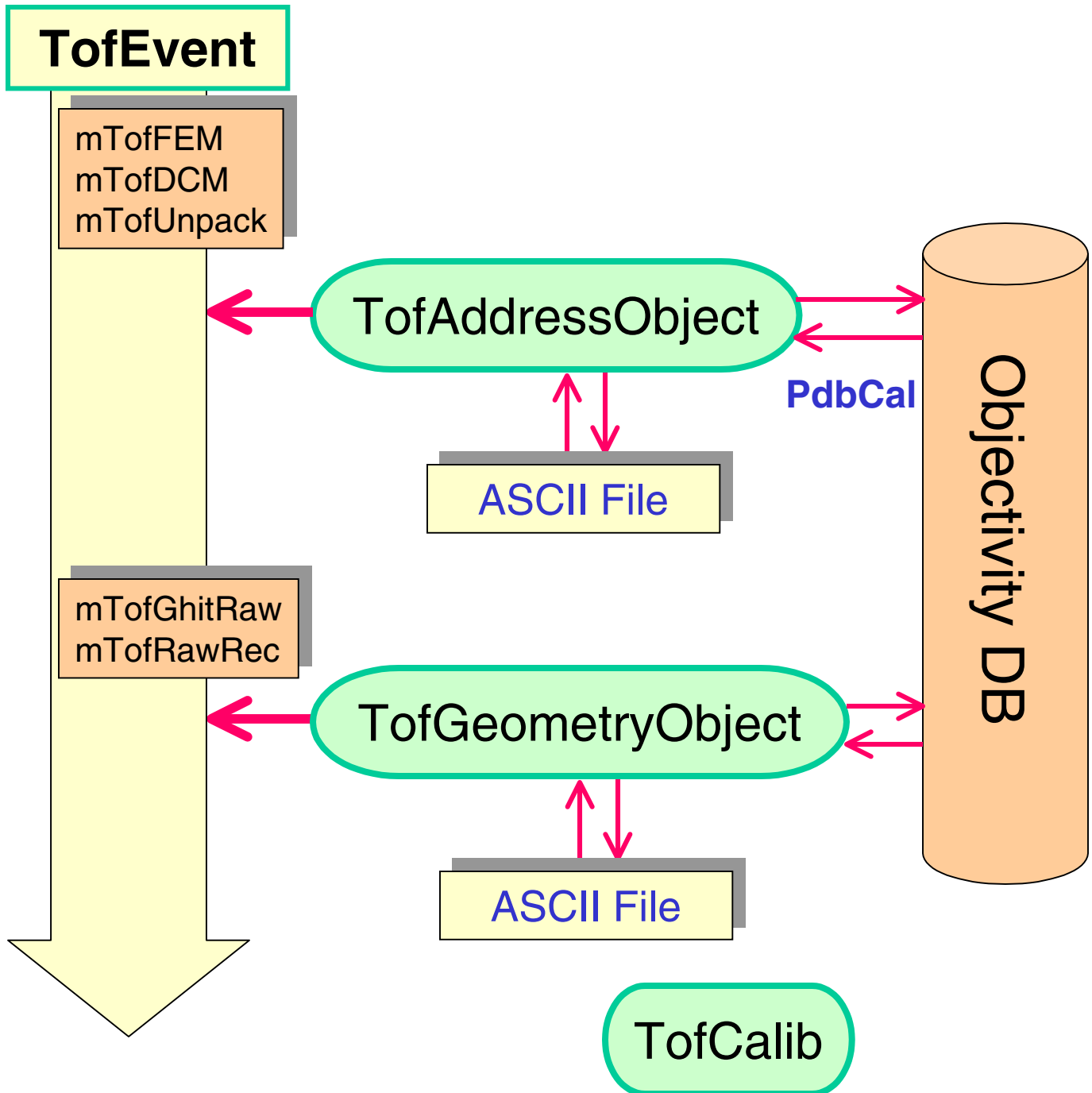
Offline Data Flow (Include Parameter Tables)



Address/Geometry Object

- Why we need Address/Geometry Object?
 - PHENIX standard language is C++ and database tool is Objectivity DB
 - Current parameter database are STAF based table. They cannot access with Objectivity DB directory.
 - Objectivity DB is handled by C++ classes.
- Let's make Geometry/Address Object Classes
 - This Object should include
 - I/O interface with Objy.DB
 - I/O interface with ASCII file
 - for easy update.
 - supplying each parameters in Response/Reconstruction Chain

Data Flow w/Database

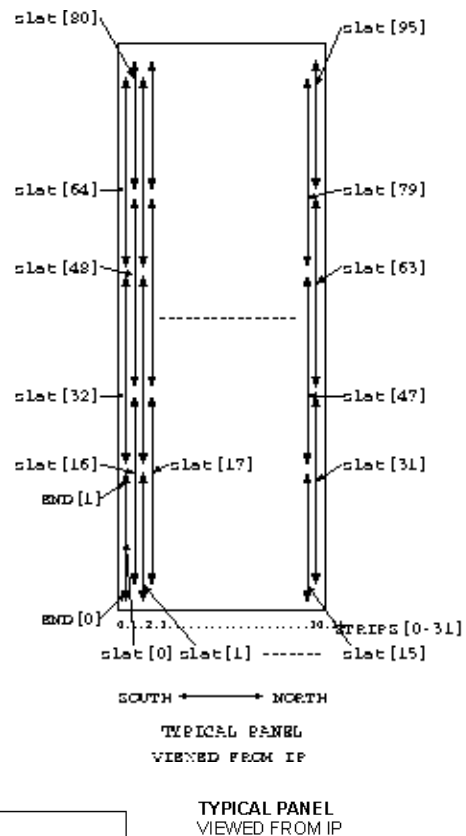
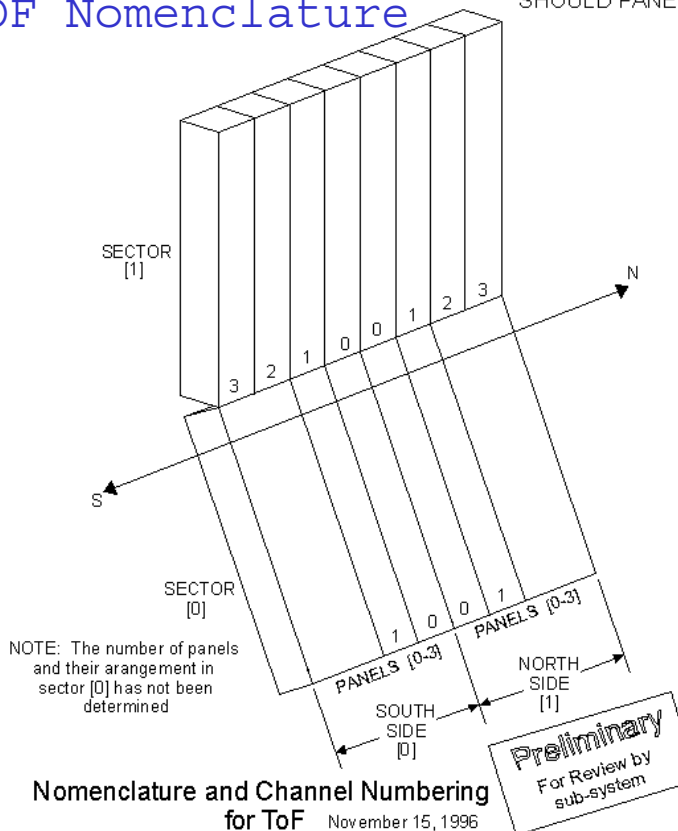


Address DB

- Geometrical (Soft) Address map
 - slatid <--> arm, sector, side, panel,slat
- FEM (Hard) Address map
 - slatid <--> crate, slot, channel
- Cable length DB
- PMT gain DB (Calibration Object?)
 - test result using β -source

TOF Nomenclature

SHOULD PANELS & STRIPS BE NUMBERED N to S or from IP OUT



Format of channel numberstatement:
 Arm [0], Side [0-1], Sector [0-1], Panel [0-3], Strip [0-31], Slat [0-2], End [0-1]

TOF Address Class

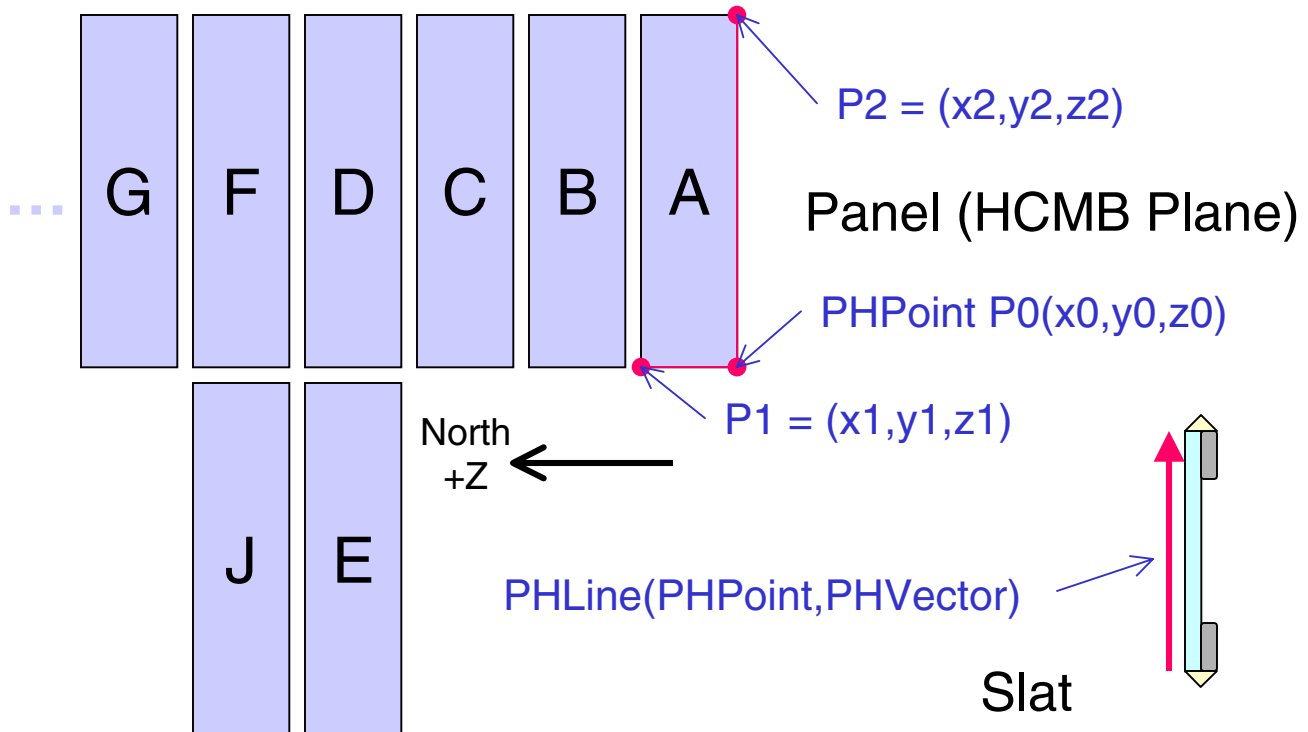
(TofAddressObject.cc)

- Organize and calculate
 - geometrical address is calculated this class.
 - FEM address is obtained from database[Objy. or ASCII]
- Method
 - `int getSlatID(int arm, int sector, int side, int panel, int slat)`
 - `int getSlatID(char *panel_char, int slat)`
 - `int getSlatID(int crate, int slot, int channel) // from FEM`
 - `int getCrate(int slatid) // FEM crate`
 - `int getSlot(int slatid) // FEM slot (board)`
 - `int getChannel(int l, int slatid) // FEM channel`
- Parameters of DB
 - FEM address map
 - `slatid, crate, slot[2], channel[2]`
- DB interface
 - PdbIndex and PHPointerList classes
 - `PdbIndex(int Min, int Max, int volume, char* Name)`

PHPointerList

```
PdbIndex slatid(0, 959, 0, "SLATID");
PdbIndex crate(0, 7, 0, "CRATE");
...
```


Geometry DB



- TOF Geometry
 - Panel : defined by PHPanel (3 PHPoint)
 - Slat : defined by PHLine (point, vector, Length)
 - calculate from TOF Panel position
- use PHENIX geometry class
 - PHPoint : 3-dim point (x,y,z)
 - PHCylPoint: 3-dim Cylindrical point (r,phi,z)
 - PHVector: 3-dim vector
 - PHLine: line define by 2 points or 1 point + 1 vector
 - PHPanel: limited plane with 4 edges
 - » defined by 3 points

TOF Geometry Class

(TofGeometryObject.cc)

- Organize and calculate
 - panel position and slat position offset parameters are obtained from database[Objy or ASCII]
 - slats position are calculated from panel position.
- Method
 - PHPanel getPanelGeo(int panel_seq)
 - PHLine getSlatGeo(int slatid)
 - PHPoint getSlatXYZ (int slatid)
 - PHCylPoint getSlatRPhiZ (int slatid)
 - PHVector getSlatVector(int slatid)
 - float getSlatLength (int slatid)
- Parameters of DB
 - Panel geometry : 3 points * 10 panels
 - Slat offset : (point + vector)*960 slats
- DB interface
 - PdbCoordinate Class in PdbCal
 - PdbCoordinate(x, y, z, dx, dy, dz)

Status&Plan

- TofGeometryObject, TofAddressObject have been released.
- Currently TOF GEANT geometry and address are stored in the PHENIX_TRY_CAS Objectivity database.
- We can test and check them.

```

rcas2021 39% root -b

root [0] .x tofObjyFetch.C
root [1] TofGeometry->print(800);

##### TofGeometryObject #####
  SLATID = 800

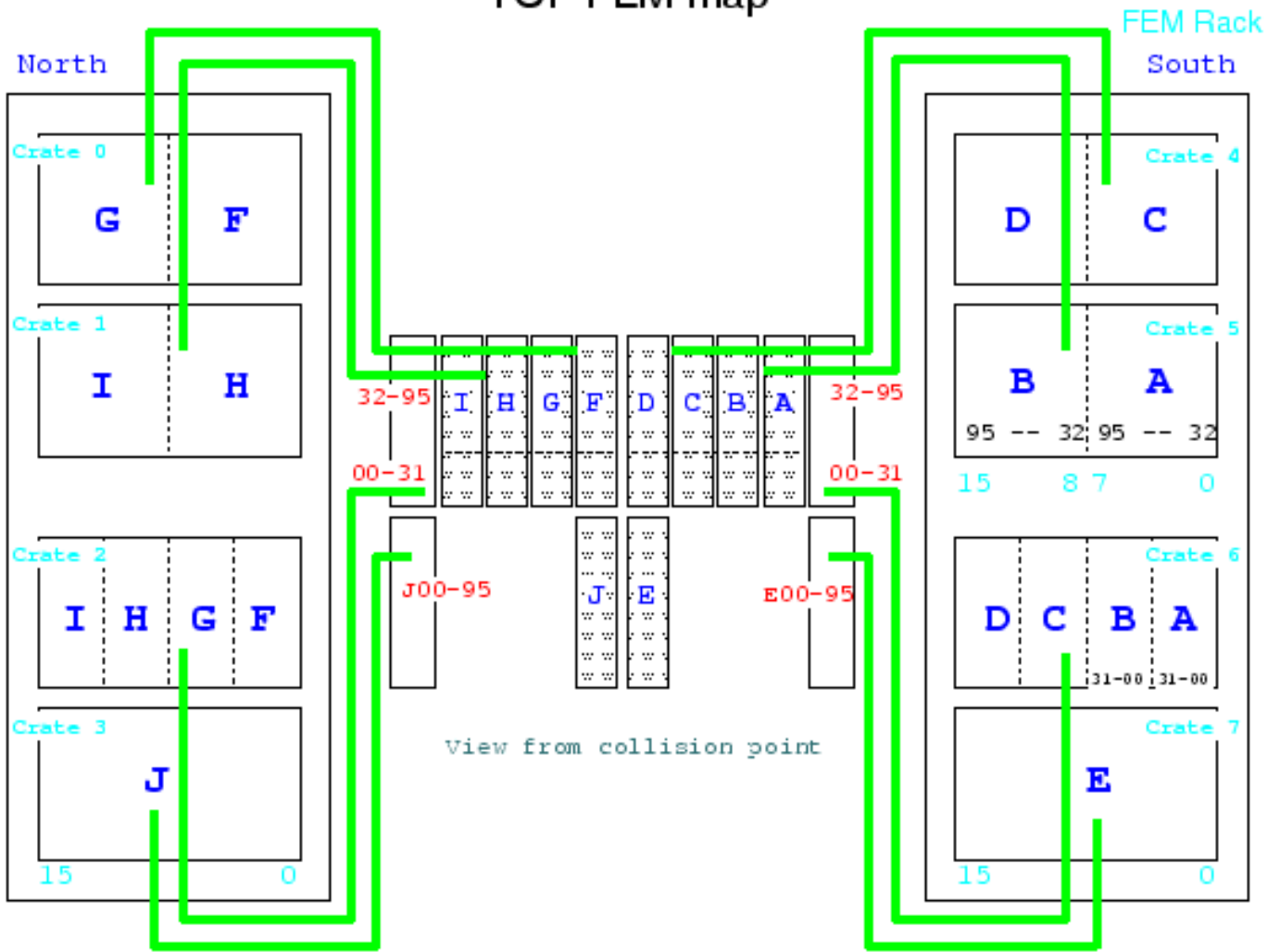
dTofGeo
  ARM      = 1
  SECTOR   = 0
  SIDE     = 0
  PANEL    = 0
  SLAT     = 32
  panel_seq = 8
  panel_char = E

PanelCenter(public) = ( -464.71, -192.49, -24.4246 )
PanelCenter(private)= ( -464.71, -192.49, -24.4246 )
SlatXYZ(public)     = ( -463.308, -204.472, -46.5594 )
SlatXYZ(private)    = ( -463.308, -204.472, -46.5594 )
SlatRPhiZ(public)   = ( 506.422, -2.72597, -46.5594 )
SlatVector(public)  = ( -0.382715, 0.923867, 0 )
SlatVector(private) = ( -0.382715, 0.923867, 0 )
SlatLength = 63.77 [cm]  SlatWidth  = 1.52654 [cm]
root [2]

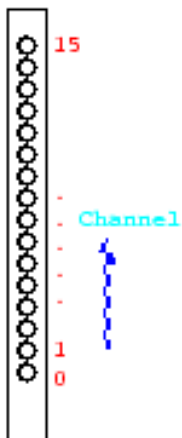
```

- Next issue
 - Write TofEvent Class using Geometry, Address and Calibration classes.

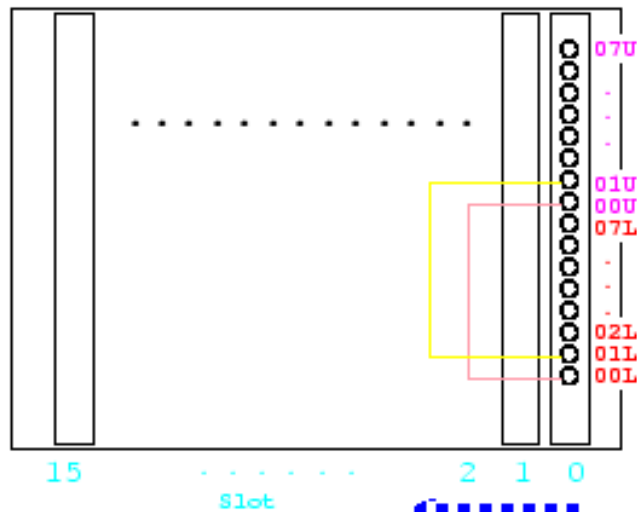
TOF FEM map



FEM Board



FEM Crate



Last Update
Mar.28, 2000
Akio Kiyomichi

