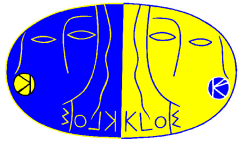
Work finished

M. Antonelli	ISR and ϕ decay simulation
M. Moulson	Bank-reduction code for DST's

Status reports



M. Palutan	Trigger simulation parameters
S. Dell'Agnello	DC geometry review
S. Miscetti	EmC response validation
S. Miscetti	Event selection for background insertion Background insertion for EmC
M. Moulson	Background insertion for DC DC dead/hot simulation
P. de Simone	DC *s*-*t* relations with new sag model
S. Giovannella	GEANFI on IBM
C. Bloise	Other MC tuning
I. Sfiligoi	DB modifications for DST's

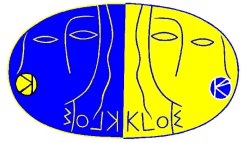
DC background insertion: status

Insertion code adapted from MBCKADD; incorporated into INSERT
Reads DTCE bank from secondary YBOS array

<u>Issues:</u>

1. Initialization of raw *s*-*t* relations for MC in DCONVR
   Default: only smear DHIT positions with 200 μm resolution
   Doesn't work for inserted hits with placeholder DHIT entries
   Default overridden by DCPRAR Run Init routine
2. Correct treatment of background hits with negative times
3. Temporal alignment of MC and background hits
4. Consistency of raw *s*-*t* relations for MC and data

Preparation of DC background hits



Hits with negative times can block physical hits

- TDC's in common stop; only 1st hit in 4 μs window read out
- Background DTCE banks must contain these hits

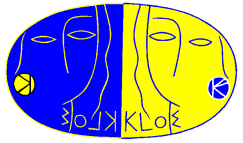
Background DTCE must not be t0-subtracted

- T0_MENO suppresses negative hits

Background DTCE must not be hot-channel suppressed

- For correct DC trigger simulation, all hits must be present
- DC trigger can use list in trigger DB to veto hot channels by run
- Use dedicated module to eliminate hot and dead channels after TSKT

Temporal alignment of background hits



Read non t0-subtracted, non hot-suppressed DTCE on secondary array

Reject background event if DTCE(1) has been t0-subtracted

Read T0GL on secondary array

Perform t0 subtraction:

```
t0 = t0_DC + t0_step1 +
     (DeltaCaviCaloRef –
          DeltaCaviCalo) –
     DtDC
```

Normally done in T0_MENO

t0's for channel defined with respect to DeltaCaviCaloRef

Mean t0 displacement in cosmic-ray events

Keep hits with negative times

Intercalate MC and background hits

Keep hit with earlier time when two hits overlap

Suppress negative hits only at end

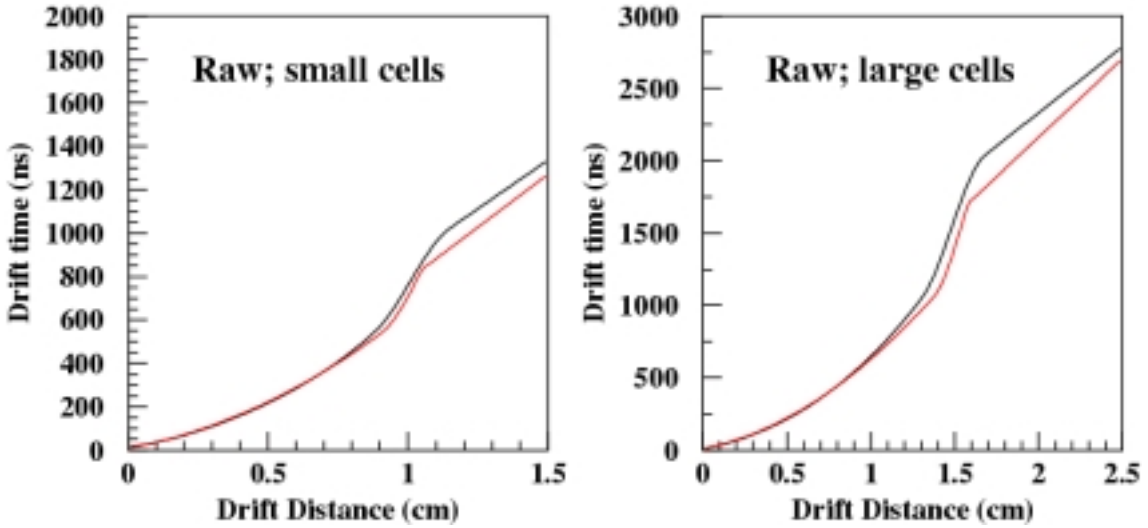
SMEAR_T0 and DCONVR assume sign carries L/R info

Negative times cannot fluctuate positive because of TSKT jitter

Raw s-t relations for MC and data

Raw *s*-*t* relations for MC/data differ by 100-200 μm at 1-2 cm from wire

Background hits not reconstructed with same radii when inserted

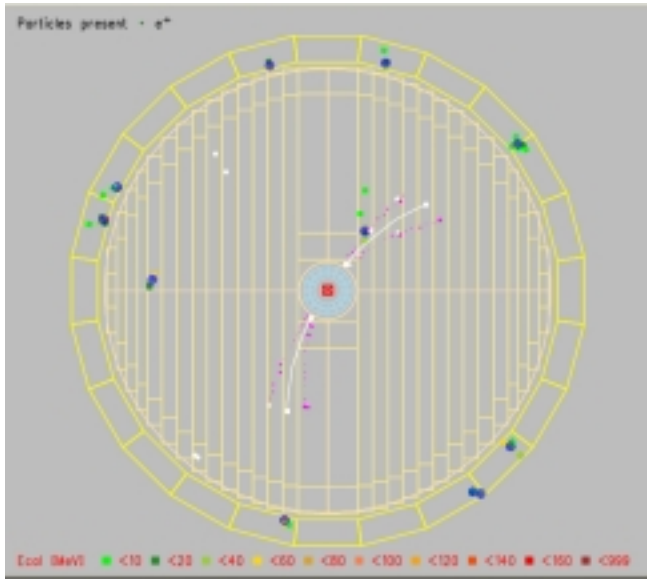


Implications for reconstruction quality for inserted tracks?

Solutions:

1. Use raw *s*-*t* relations for data in MC
   For fine *s*-*t* relations, want to simulate effect of not taking wire sag into account when calculating cell geometry
2. Adjust “data” times to “MC” times in INSERT

Effect of different s-t relations



MC events:

KS, KL neutrals

All hit banks dropped

“Background” events:

Events in **bha** stream

At least 2 tracks with >20 hits each

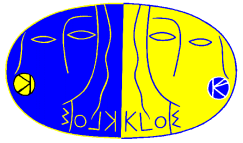
Study reproducibility of track reconstruction when inserting tracks into MC events without hits by visual scan of 100 events (200 tracks):

Intact reconstruction	
Perfect	**164**
Split reproduced	**9**
Different reconstruction	
New split	**16**
Split recovered	**5**
Badly reconstructed	**1**
Lost	**1**

About 90% reproducible, with few % excess of split tracks and small losses

(allowing cancellation of new/recovered splits)

DC insertion: to-do list



For DC insertion/simulation code:

A/C module to suppress hits on hot/dead channels

Adjustment for different *s*-*t* relations in MC/data

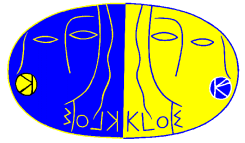
Form of solution (if any) to be determined

Make pristine copy of DTHA bank to allow reprocessing without background?

For INSERT module:

Install KID routines for reading background file

Combine or separate neutral kaon runs?



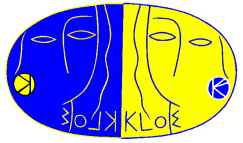
Combined production $K_S \rightarrow$ all, $K_L \rightarrow$ all	Separate production $K_S \rightarrow \pi^0\pi^0$, $K_L \rightarrow$ all $K_S \rightarrow \pi^+\pi^-$, $K_L \rightarrow$ all Also differentiated by K_L decay in DC?	Combined generation $K_S \rightarrow$ all, $K_L \rightarrow$ all Streaming to dst by MC truth
Simpler to produce	Simpler to analyze	Simple to produce and analyze (if no reprocessing)
Fewer files (if file length unsaturated)	Smaller files	Smaller files
Less disk turnover? (if people cooperate)	Less disk turnover? (if event subset dominates interest)	Less disk turnover? (if event subset dominates interest)
	Lighter disk access (if event subset dominates interest)	Lighter disk access (if event subset dominates interest)
No need to prioritize	Possible to prioritize	No need to prioritize
Naturally treats rare channels		Rare channels treated well in generation Problems with zero-length files
Well-suited for background studies (rare K_S decays, non-K_SK_L physics)		Acceptable compromise for background studies (mechanically more running, total volume and content of data set unchanged)

Production scheme: examples

	$K_S K_L$	$\phi \rightarrow$ all	$K_S \rightarrow \pi^+\pi^-$ $K_L \rightarrow \pi^+\pi^-$
σ (nb)	1050	3100	1.4
Max evts/1GB file	25000	30000	40000
Max L per file (nb^{-1})	25	9.7	28000
Files/200 nb^{-1} run	8 full length	20 full length	0.008 (1 file ~320 evts)
If 15 raw files of ~13 nb^{-1} each…	OK 15 files ~55% full	Must split MC files 30 files ~50% full	15 files ~20 evts each Must group raws
Background reuse factor	35	100 → 50 (raw files used twice)	0.033

To split MC files across raw files: background from entire raw file used for each corresponding MC file; reuse factor adjusted accordingly

Requirements on DB2



Background can be treated as a datarec stream

New DB requirements for MC runs/files:

Runs are generated for each raw file in data set

Additional complications from grouping raw files/splitting MC files

New DB2 tables in logger schema for official MC production:

.mco files specified by (**logger.mc_runs** → **logger.mcprod_runs**):

MCCard_Id, *MCRun_Nr*, **Farm_Id, GB_Nr, MCFile_Nr**

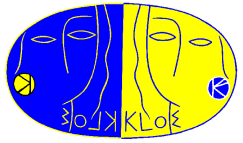
.mcr/.dst files specified by (**logger.mc_logger** → **logger.mcprod_logger**):

MCCard_Id, *MCRun_Nr*, **Farm_Id, GB_Nr, MCFile_Nr**

MCBack_Id, Datarec_Version, Stream_Id, **MCStream_ID(?)**

Old tables will continue to exist (for use for “small” generation jobs)

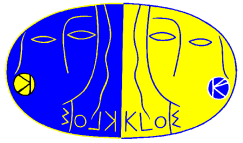
Run-variable quantities for generation



Item	Needed for	Source	Ready?
$\sqrt{s}$	Gen	HepDB	yes?
$\mathbf{p}_\phi$	Gen	HepDB	probably
$\mathbf{x}_\phi$	Gen	HepDB	probably
Offline DC hot/dead	Recon		
Trigger DC dead	Recon	DB2	
Trigger thresholds	Recon		
s-*t* relations	Gen/Recon	HepDB	Needed?

List of runs to generate; verification of DB entries for all such runs?

Scripting tasks



For each **raw** file in data set:

Background selection:

Open and scan **bha** files, write background file

Generation:

Read ($\sqrt{s}$, $\mathbf{p}_\phi$, $\mathbf{x}_\phi$), random seeds from DB

Read luminosity from DB, calculate number of events to generate

Write cards and submit job

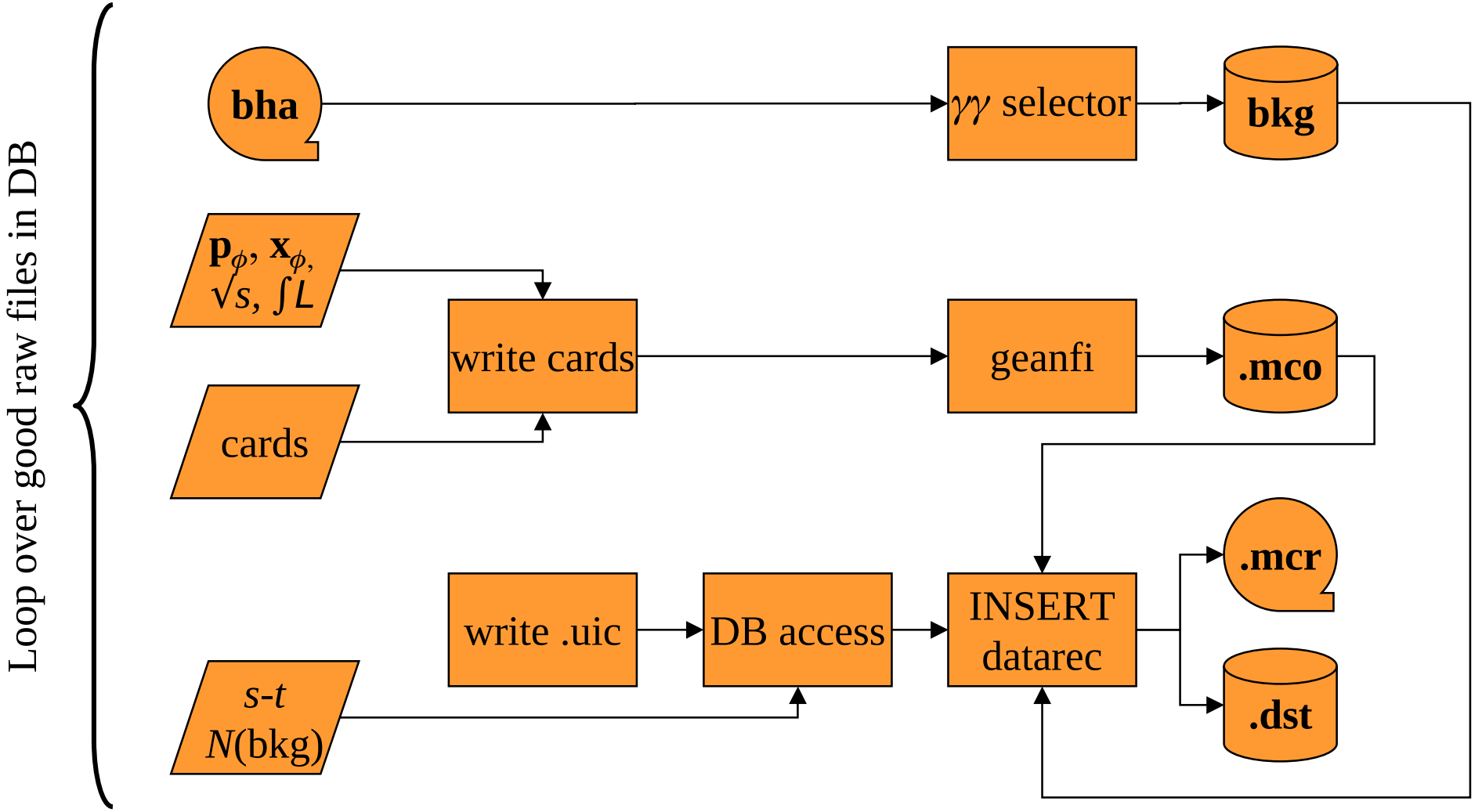
Reconstruction:

Look up background file, calculate event-reuse factor

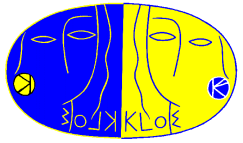
Generate talk-to inputs for any modules requiring them

INSERT requires specification of background file, reuse factor

Complete production flowchart



Unresolved questions/task assignments



Questions:

Restrict MC production to AIX machines?

Use queue manager?

New task assignments:

- DB modifications
- Scripting/DB interface for background selection
- Scripting/DB interface for generation phase
- Scripting/DB interface for reconstruction phase

Is there a more natural way to divide up scripting work?

- Good run list/DB verification