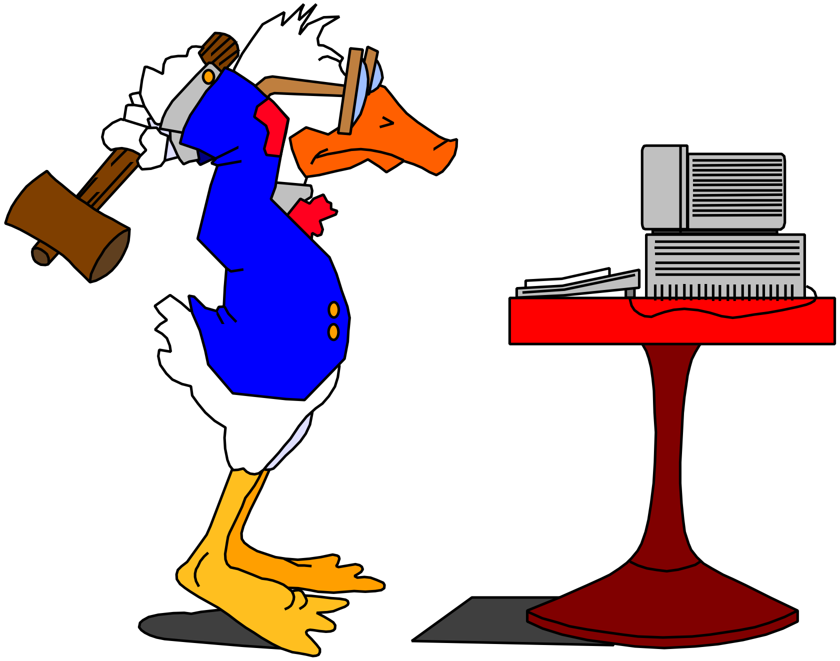
STAR COMPUTING

STAR Data Analysis



Joint Alice-Star Meeting

BNL, April 8-9 2000

STAR Data Analysis

- Framework same for reconstruction & analysis
- Analysis specific infrastructure is still evolving
- The how-to is clear - just has to get done
- Physics working groups PWGs actively develop code which are:
  - PWG specific analysis
  - common: simplify specific tasks, plotting, data handling
- Missing pieces are no show-stoppers (MDC3)
- Still: we benefit from the fact that year1 is simpler in many aspects

Components

- root4star, STARs ROOT framework
  - Maker scheme, I/O Maker
- Persistent OO Datamodel: StEvent
- Interface to simulated/embedded data: StMcEvent
- Grand Challenge Apparatus, data mining
  - Tag Database, data mining
  - File Catalog
- Histogramming: ROOT
- Visualization: ROOT based, independent DSV
- Most components were written/implemented during the last 2 years ⇒ 100% C++

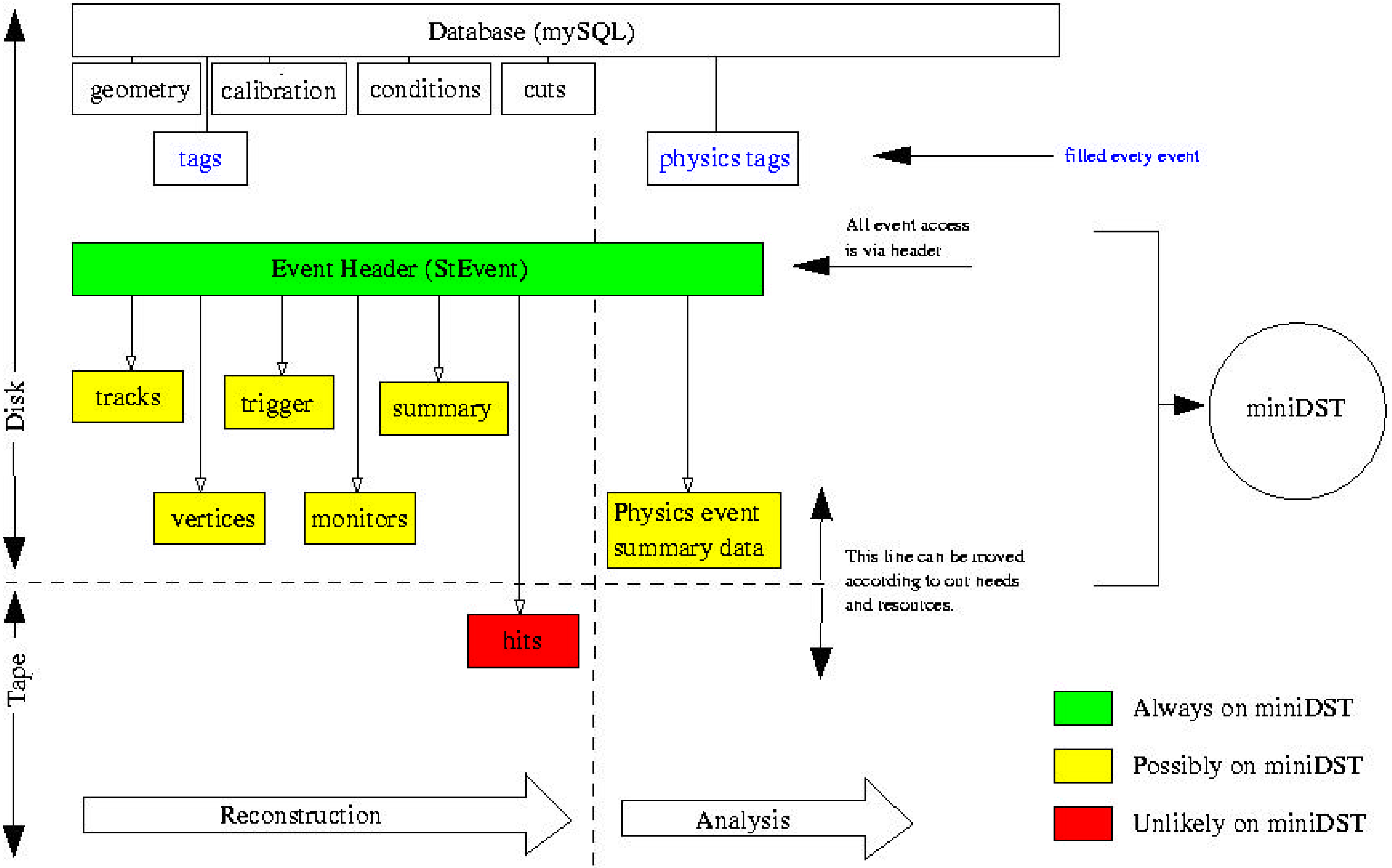
STAR COMPUTING

Framework - How-to

- Macro: doEvents.C
  - arguments: nevents, data files/data directories
  - loads shared libs
  - sets up chain : I/O maker + (StEventMaker) + AnalysisMaker
  - loops over events
- Class StAnalysisMaker
  - contains the analysis code
    - StMaker::Init(): book histos, ntuples
    - StMaker::Make(): selection, cuts, combinatorics etc.
  - compiled as shared lib loaded by doEvents.C
- Both exist as working examples - modified by user/PWG
- Simple scheme: allows new users a quick start but gives enough flexibility for complex tasks (example)

xDST, x = mini, μ, nano, ...

- Most urgent issue: implemented soon, almost ready
- STAR rule: miniDST == STAR-wide defined
  - usable by everyone
  - Format/Classes defined within StEvent
  - usually a subset of StEvent
  - allows correlation with TagDB
- relaxed: xDST where x ≥ μ can be user/pwg specific
  - often simple ntuples/trees
  - or self made
  - or again a smaller subset of StEvent



Database (mySQL)

geometry

calibration

conditions

cuts

tags

physics tags

filled every event

All event access is via header

Event Header (StEvent)

tracks

trigger

summary

vertices

monitors

hits

Physics event summary data

This line can be moved according to our needs and resources.

miniDST

Disk

Tape

Reconstruction

Analysis

🟩 Always on miniDST

🟨 Possibly on miniDST

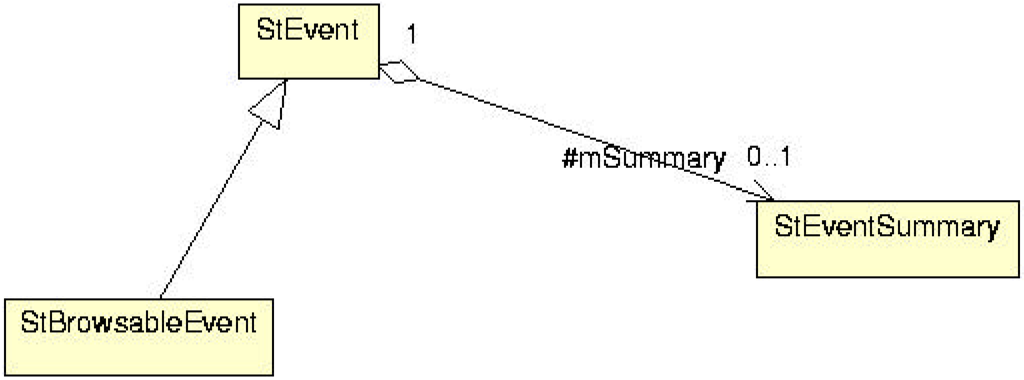
🟥 Unlikely on miniDST

StEvent

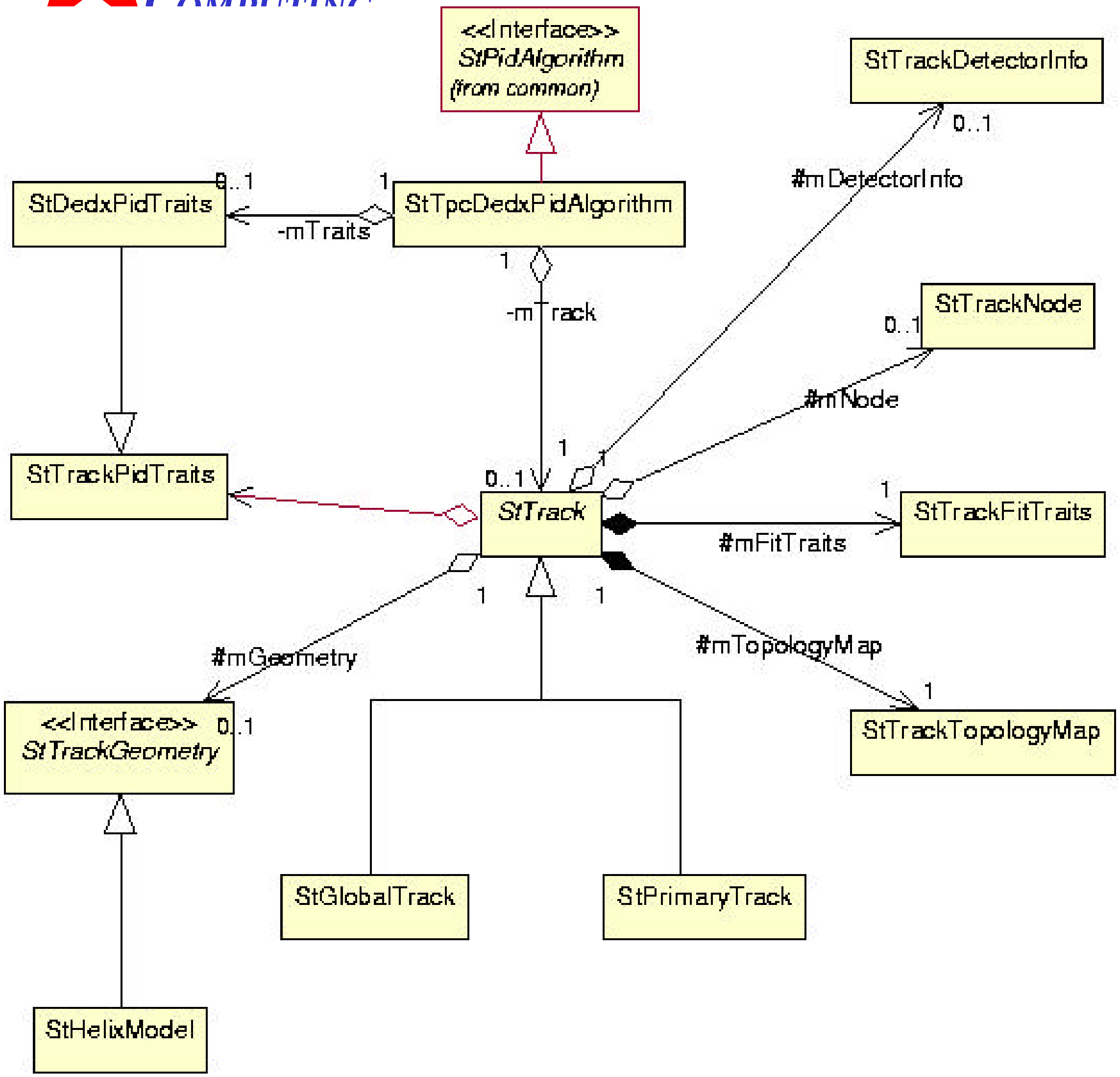
- Currently: OO Datamodel for DST analysis
- Very soon: StEvent directly written == DST
  - currently: DST == tables
  - tables read in and used to fill/setup StEvent (overhead)
- Soon: add PSD classes (provided by PWG)
- Slowly but steadily: extend into reconstruction
  - Slow because of FORTRAN legacy code
    - essential components (TPC reco)
    - global reconstruction **must** come next
  - New code, e.g. EMC fills directly to StEvent
- Goal: Become overall persistent datamodel for reco + analysis

StEvent

- Top class: StEvent == event header
  - from here: navigate to tracks, hits, trigger, etc.
  - one single entry point
  - components can be used independently
- StEvent inherits from St_DataSet
- All other classes inherit from StObject (TObject)



StEvent



- Heavy use of abstract interfaces
  - track model
- Use of functors
  - PID algorithm
  - filters etc.
  - sharing of functors between users (PID)
- Minimize dependencies
- Use of StarClassLibrary
  - small foundation classes
- Steadily growing
  - new EMC, L3
  - to come RICH, TOF

StEvent

- All containers based on : StArray
  - STL interface
  - Soon to work with STL algorithms
  - Internally mapped to TObjectArray
  - Distinguish structural/non-structural containers
    - structural: owns objects
    - non structural: only references objects
- Pointers can be stored as such

```
class StPrimaryTrack {
// …
private:
      StVertex *mVertex;  //$LINK
// …
      ClassDef(StPrimaryTrack,1);
};
```

C++/STL/ Platforms

- StEvent uses STL where CINT doesn’t see it
- STL is increasingly used in user code and reconstruction code (user like it)
- SUN/Solaris
  - until recently CC4.2 + ObjectSpace
    - non-ANSI, lot of things missing, lousy all together
  - new CC5
    - ANSI/ISO & StandardC++Library
    - lots of confusion for ROOT port uses old `iostream`
    - still these damned warnings: `X is hiding Y`
- Linux
  - nice and easy
  - STAR rule on Linux : warnings == error

MDC3

STAR-only exercise directed at physics analysis, exercising physics analysis software, analysis infrastructure and CAS tools and facilities.

- 2 weeks, short but well attended (~50-60), 4 (5) out of 6 PWGs
- Physics-wise
  - 1k - 50k event analyzed (~150k produced)
  - first time use of full set of tools (efficiencies!)
  - lots of activities, meetings, discussions
- RCF-wise
  - lousy 1st week, servers moved, RAID crashed, afs gone, network bad
  - 2nd week more stable, data file corruption, LSF configuration

Outlook

- Some components still implemented and exercised on bigger scale
  - Grand Challenge, TagDB
  - Embedding
  - StEvent (ROOT) I/O
- Use MDC3 finding to increase pressure on RCF (**has** to improve)
- *Almost* ready for year-1, still lots of work to do
- Less changes in year-2 (compared to reconstruction)
- Real test: 1st run May/June - August 2000

Ask again @ Quark Matter, Jan 2001