STAR simulations

GSTAR framework

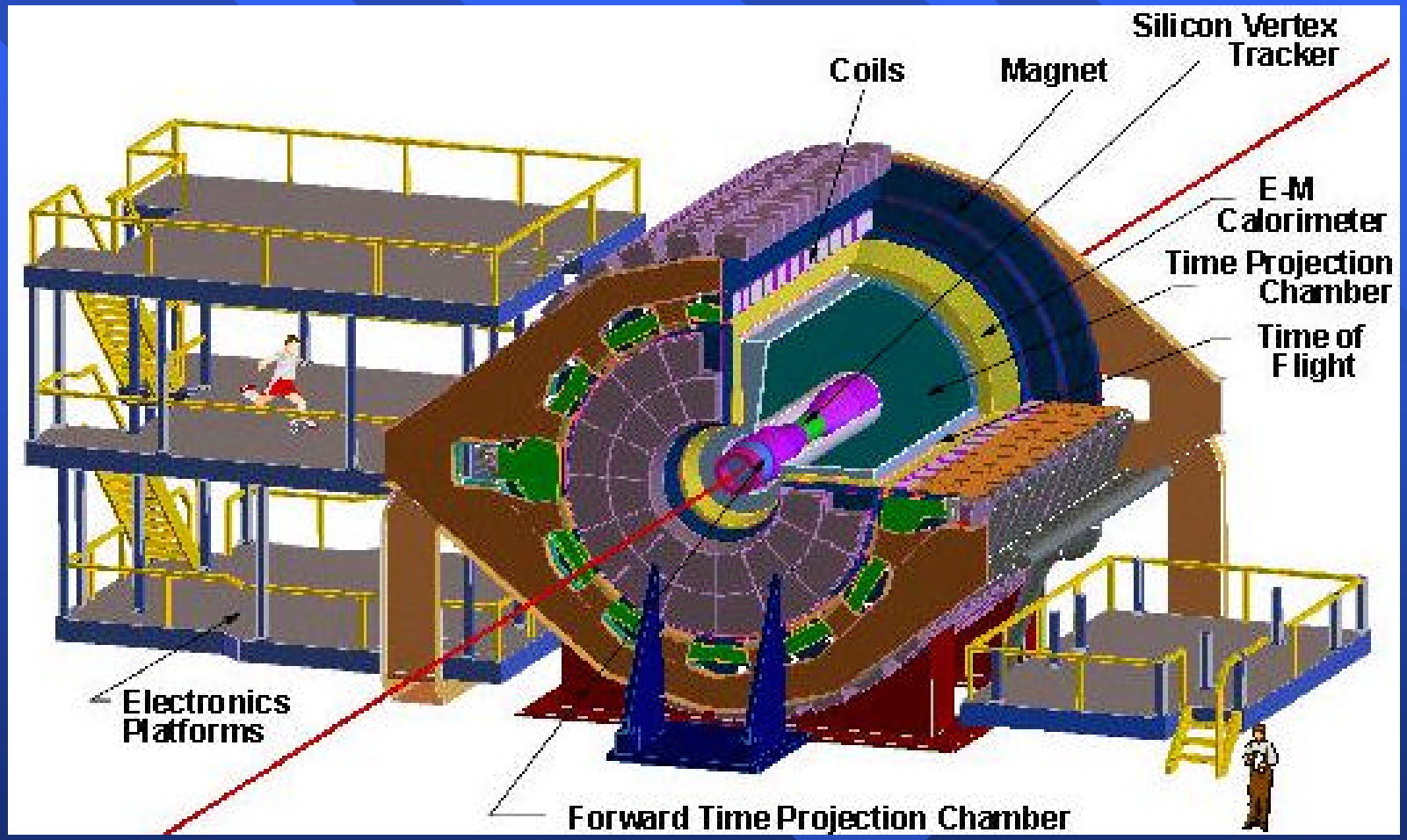
OO geometry/event model

NOVA components

Pavel Nevski

BROOKHAVEN NATIONAL LABORATORY

STAR detector at RHIC



GSTAR

- STAR simulation framework since 96
- has a hierarchical design to clearly separate user code from implementation details
- has improved memory management
  - » elastic ZEBRA (using malloc)
  - » no limits on number of tracks, vertices, hits etc (apart from physical memory limits)
- has built-in interfaces to implementation
  - » Geant3/PAW, MySQL, ROOT

Hierarchical design

- Open System Interconnection (OSI) model as example: functionality in term of layers
  - basic (physical) layer - platform dependant code, system libraries,graphics etc
  - low (logical) layer - ZEBRA, DZDOC,HIGS
  - upper (transport) - G3, Paw+Kuip, DB, ROOT
  - system (session) - AGI, ROOT accessors
  - user (application) - modules in F, AGI, C++

STAR geometry

- Formalized description in specification language, including hits and DB access
- Many developers, very detail geometry (almost 2,000 different volumes)
- Altogether less then 8000 lines including field parameterization , easy to read
- No step routine is needed in most of the detectors, no “if statement” problem

GSTAR performance

- Fast enough - 30 min/10,000 particles, with a general 1 MeV cuts
- Calorimeter cuts tuned with test beam date down to 50 KeV
- Interfaced to all event generators
- Robust and well debugged production tool

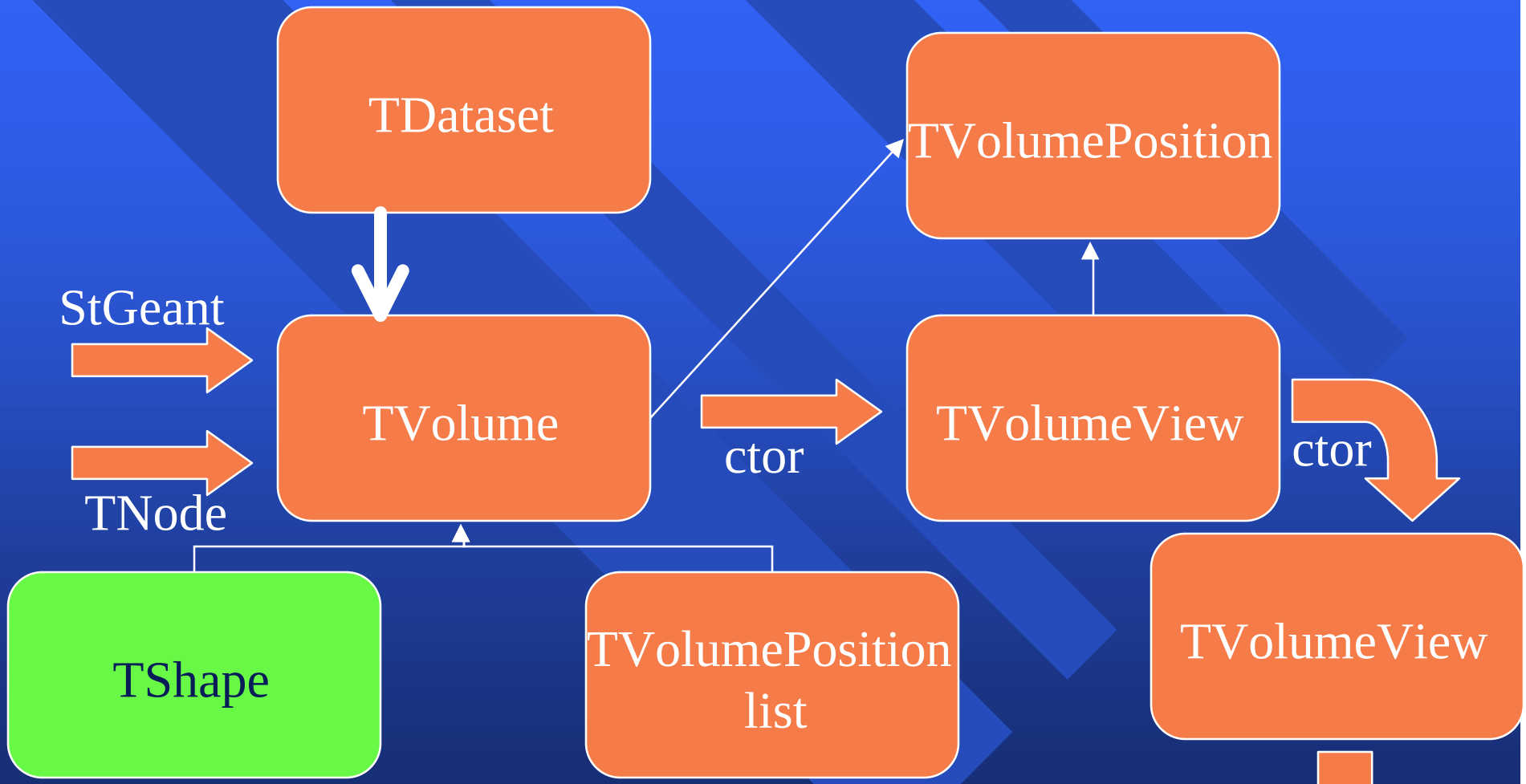
Requirements for rOOt interface

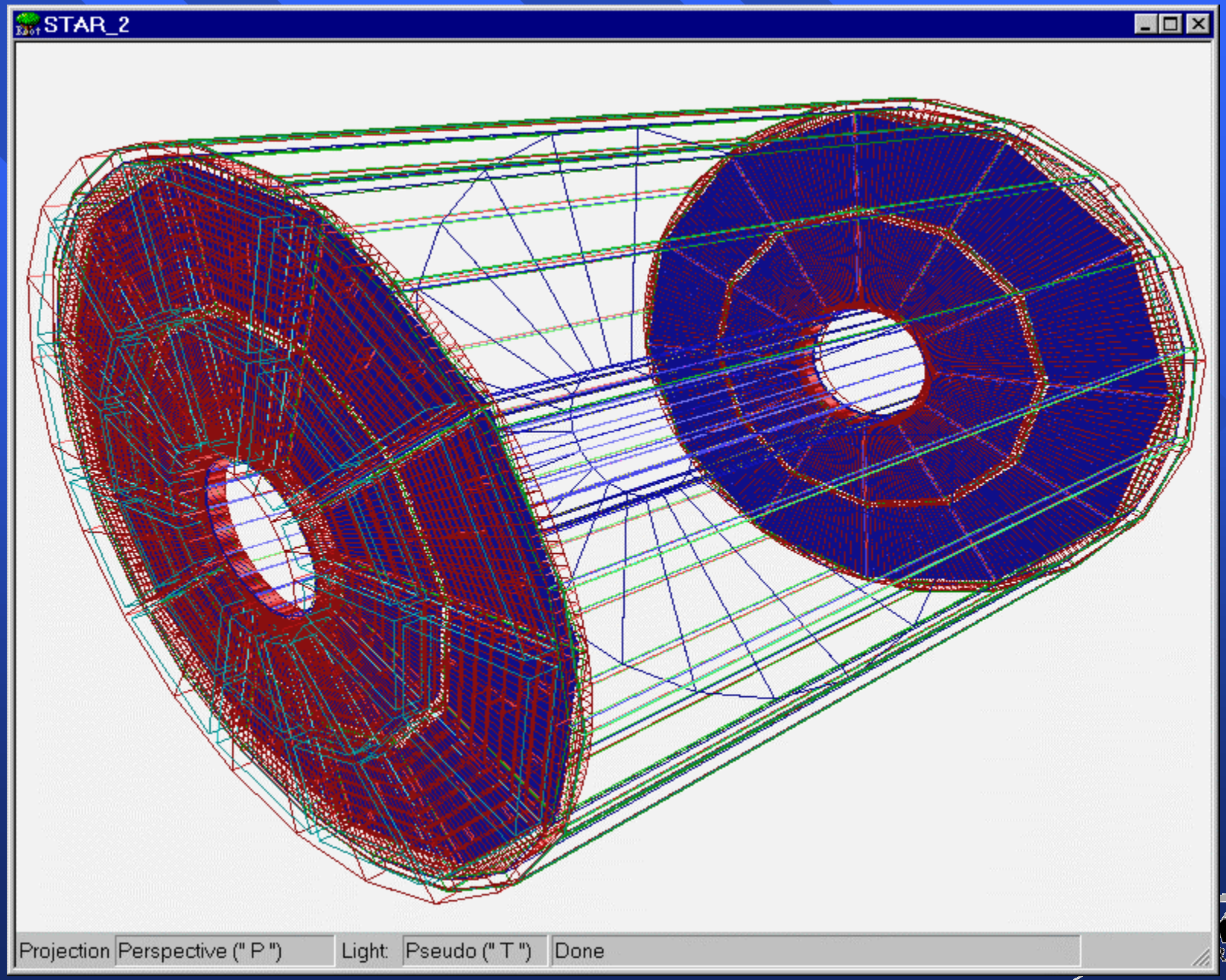
- Flexible, expandable access to geometry objects from reconstruction program
- Modern visualization and navigation
- Access to hits from a C++ code as if they were normal C++ objects
- fun, and even more fun

New elements

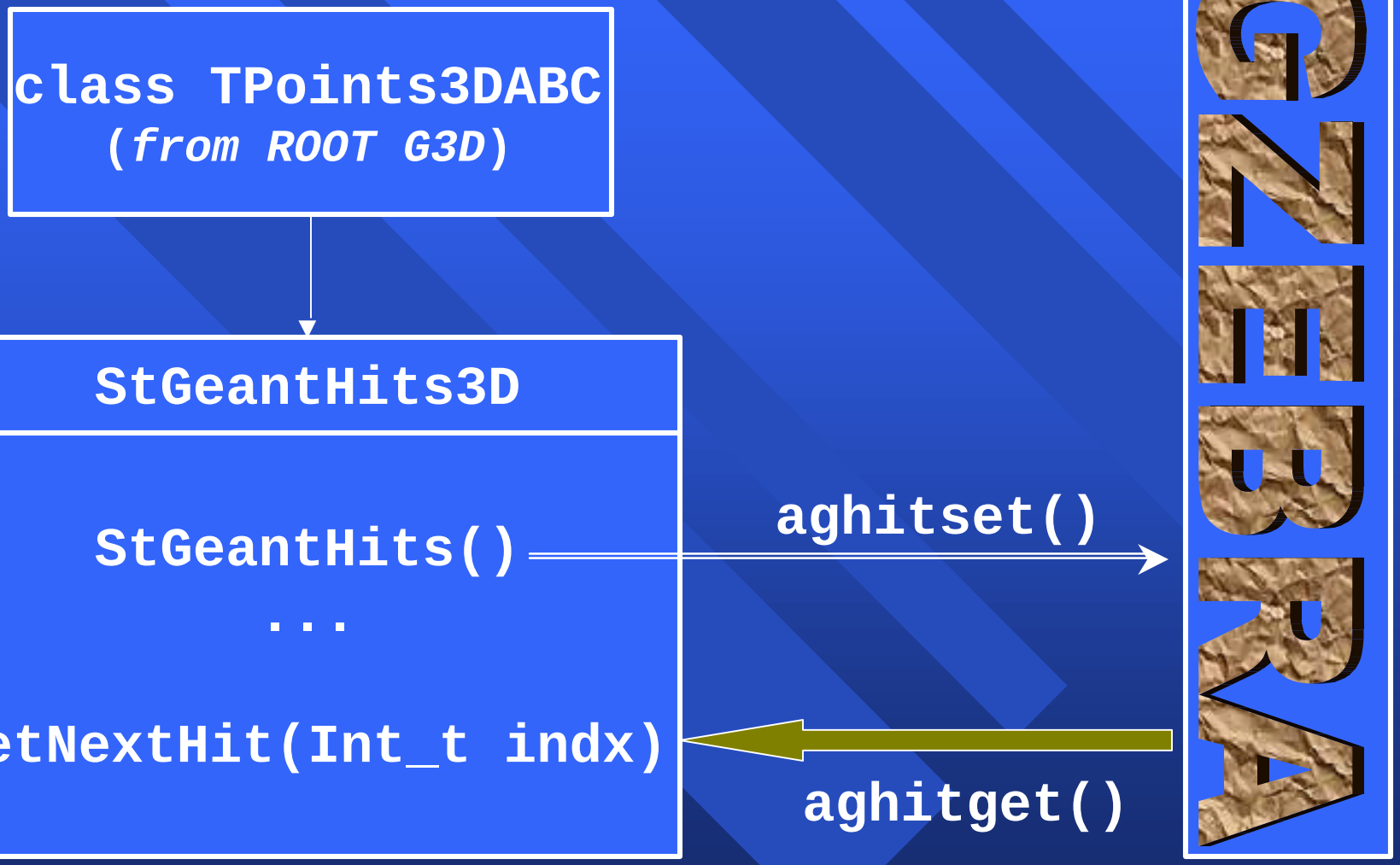
- Initially missing elements
  - Geometry navigator - trivial
  - Geometry decoder - not so trivial, but feasible
  - Volumes and positions separately - TVolume
  - Volumes as position container - TDataset
  - Hit navigator - trivial
  - Hit presenter - StGeantHits

G3 geometry model

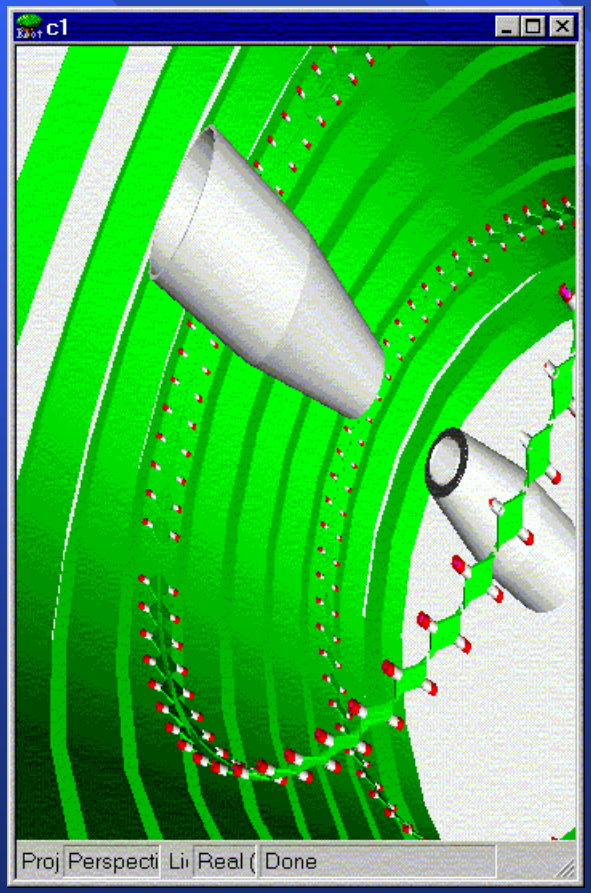


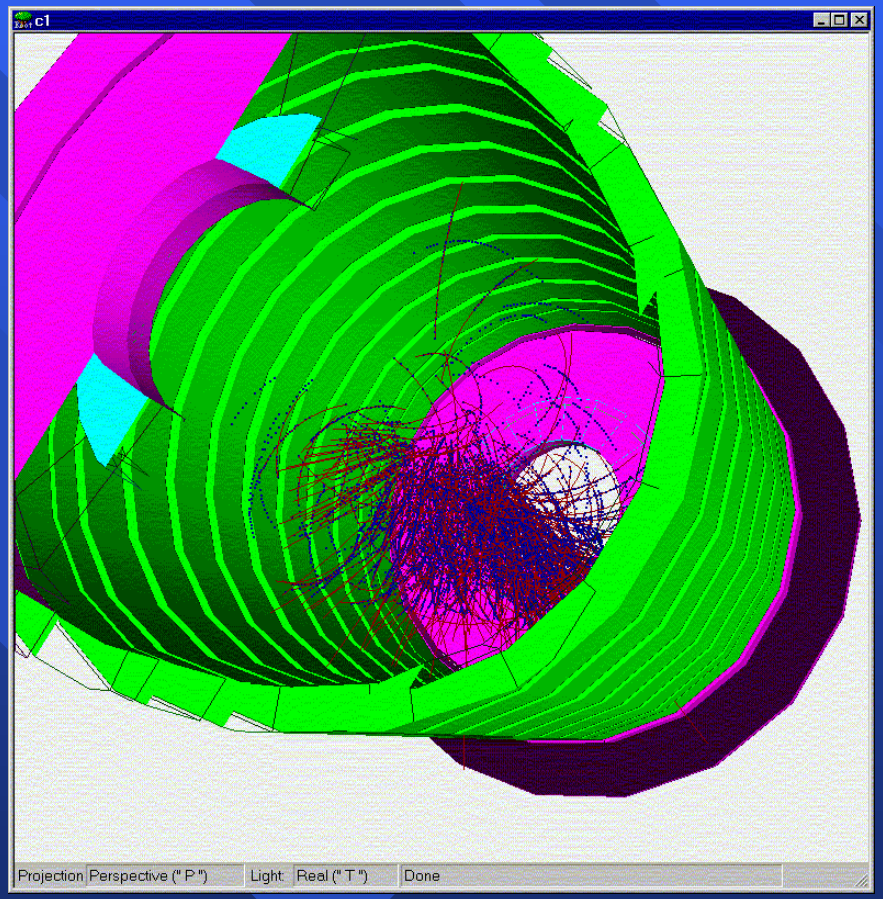

STAR_2
Projection
Perspective (" P ")
Light:
Pseudo (" T ")
Done

Geant Hit Access Class



OpenGL viewer



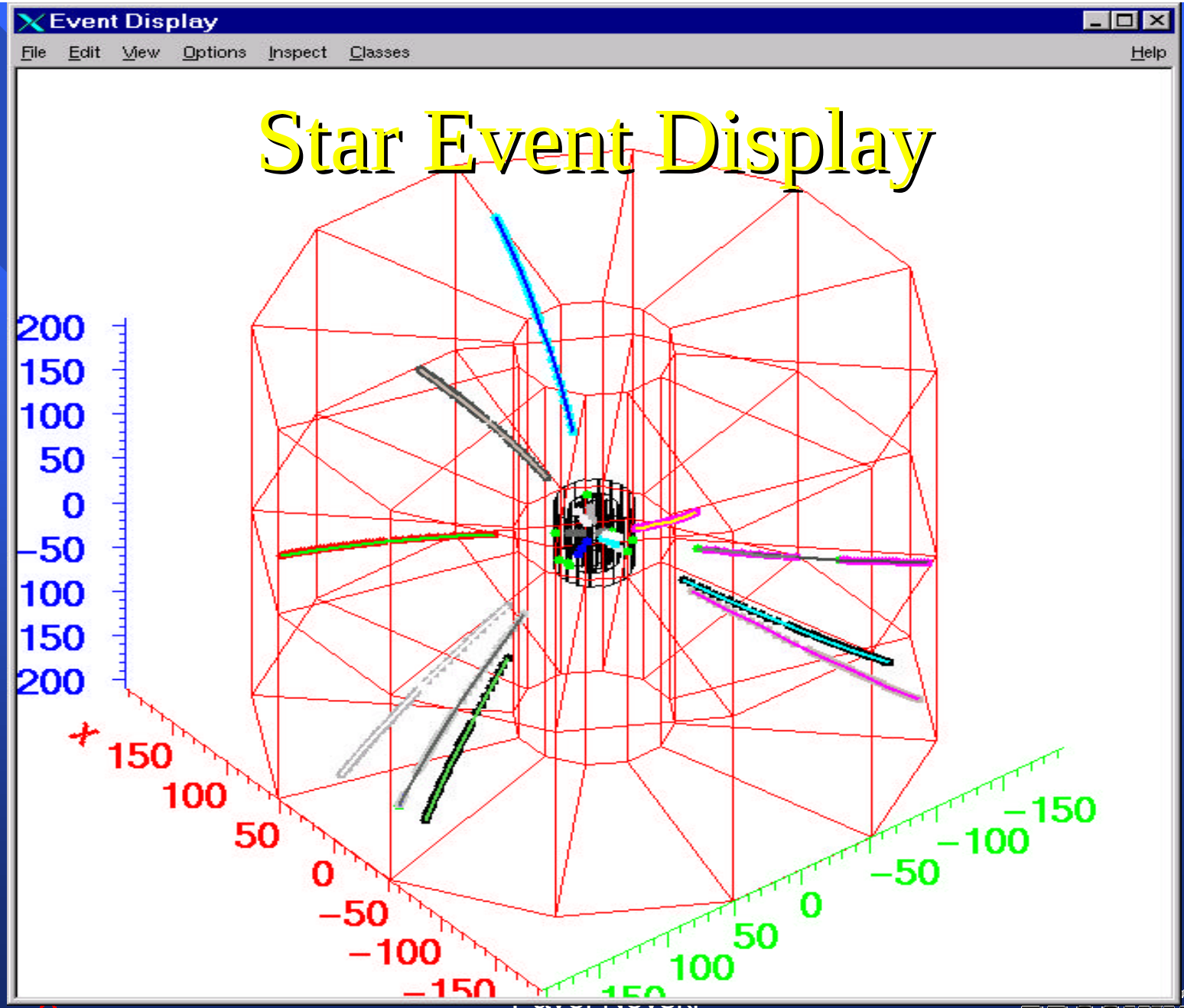


Pavel Nevski

STAR

BROOKHAVEN NATIONAL LABORATORY

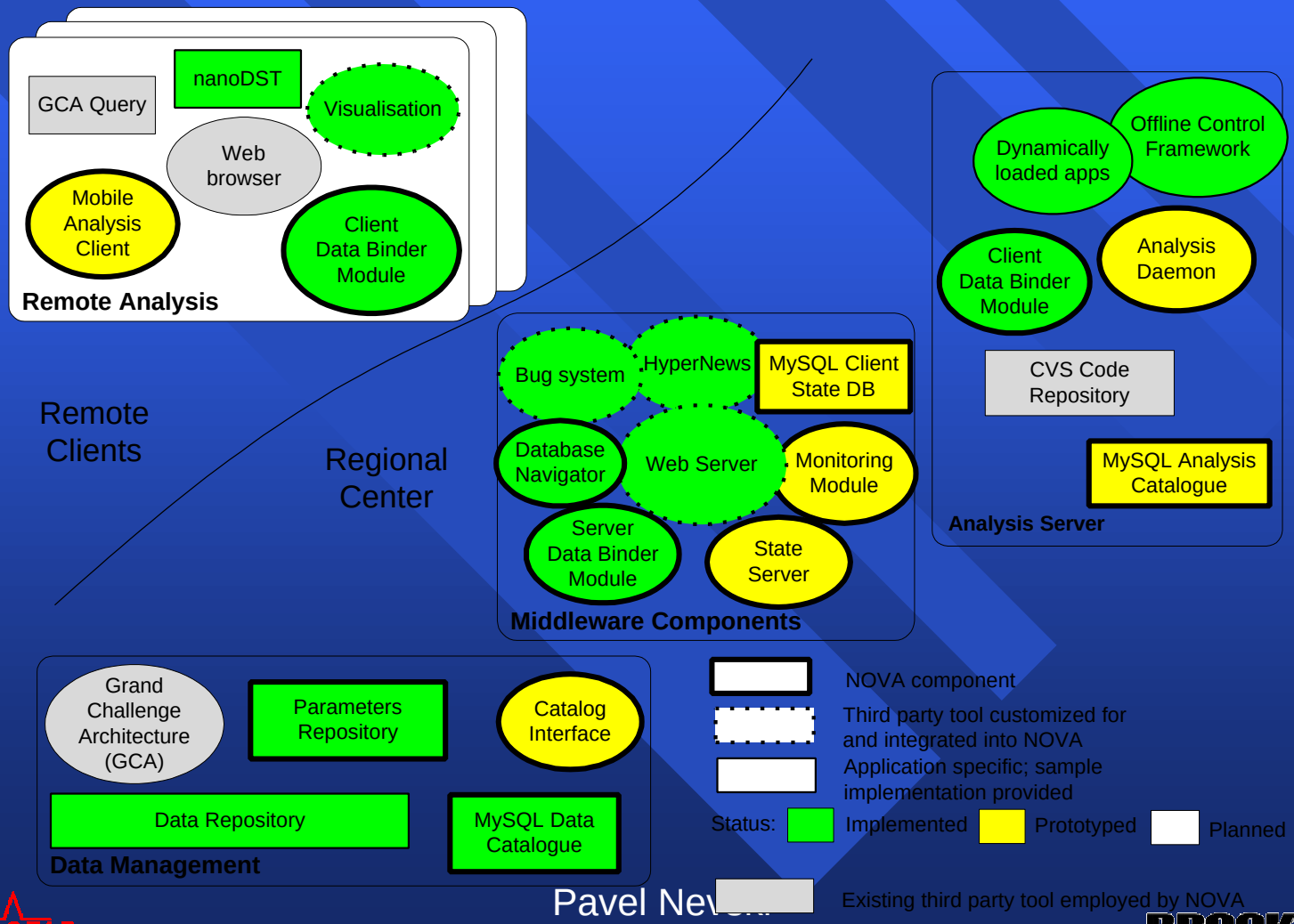
Star Event Display



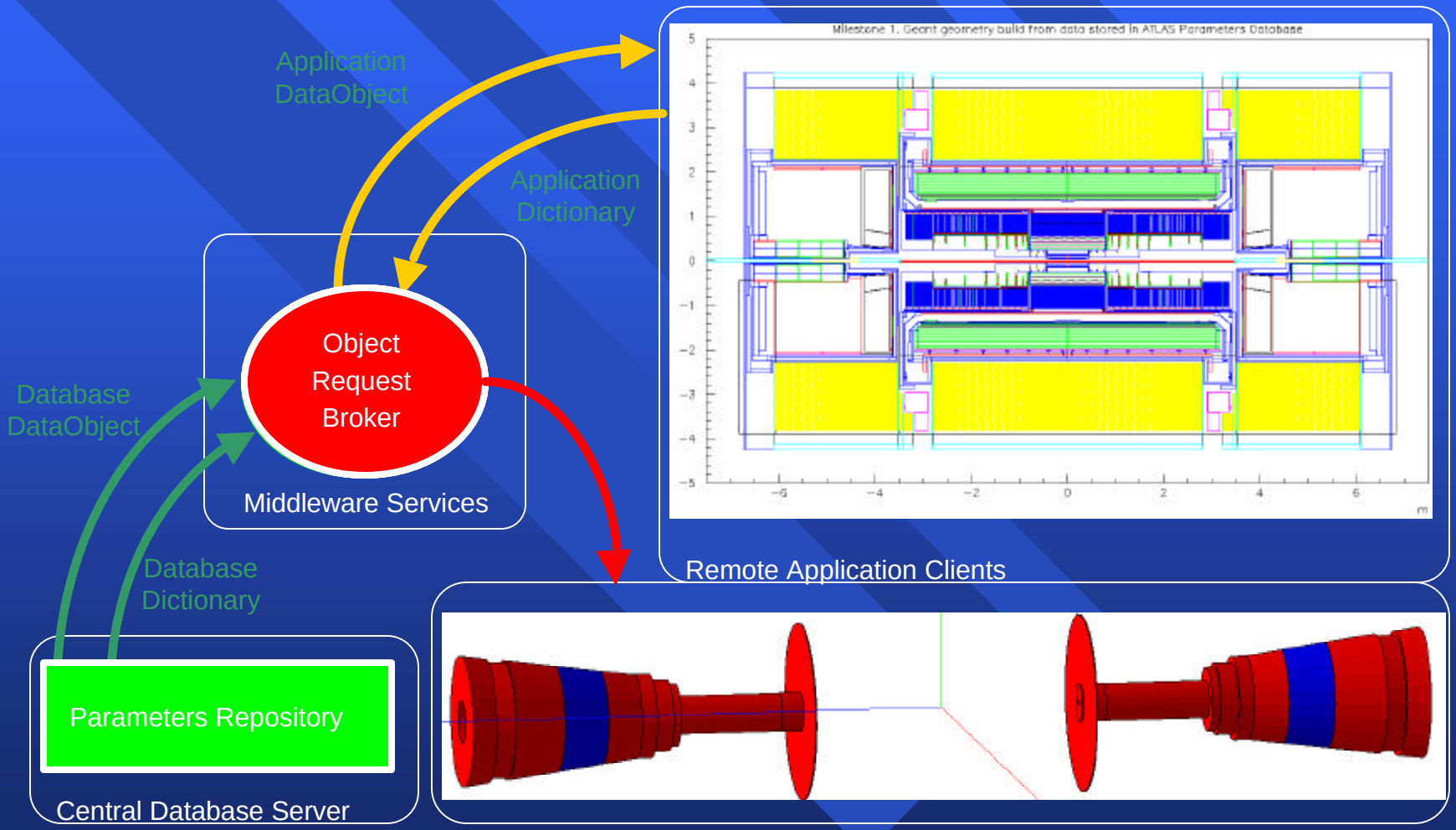
STAR

BROOKHAVEN NATIONAL LABORATORY

NOVA Architecture



Dynamic Object Broker



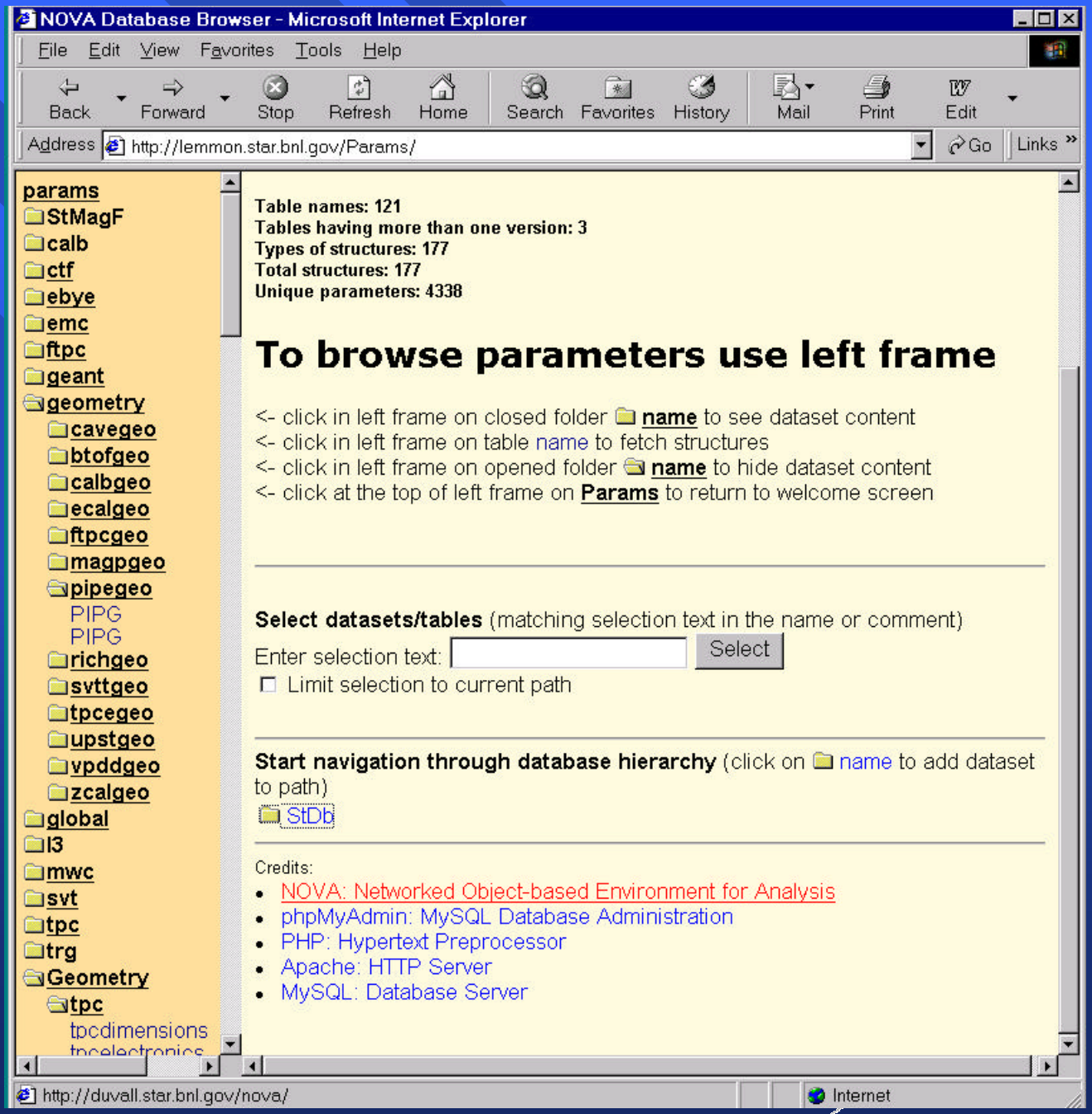
Pavel Nevski

STAR

BROOKHAVEN NATIONAL LABORATORY

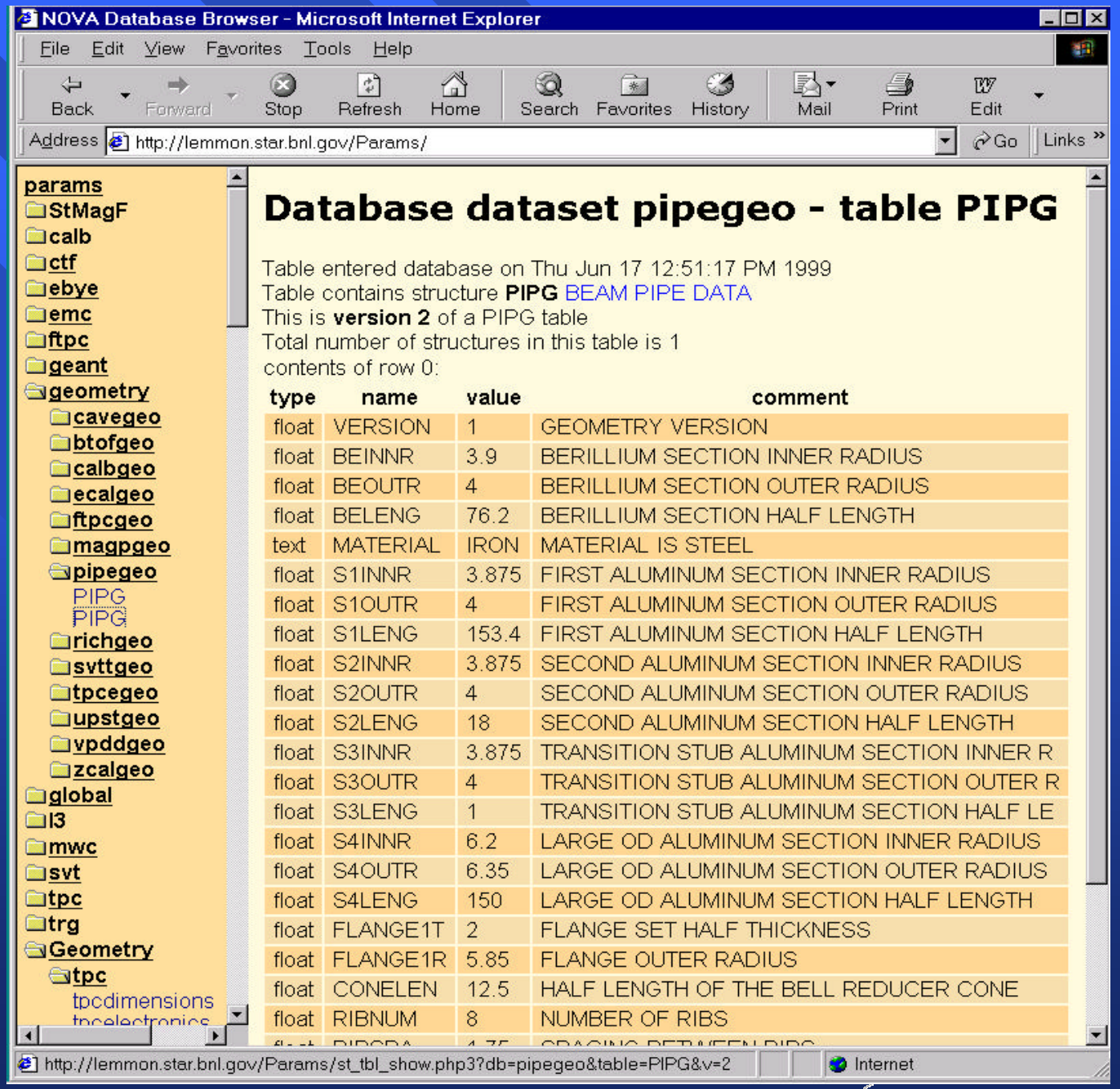
STAR geometry

- Modules: 14
- Structures: 34
- Instances: 45
- Parameter values: 841



Database Browser

- Versioned geometries



Simulation Production

- Multilayer model (NOVA component)
  - universal Simulators (smart cockroaches)
  - common Dispatcher - token coordinator
  - output/QA Filter
  - HPSS Sinker
- Open system - extendable functionality

Production bandwith

- simulation - 50 ev/day/processor - scales well
- dispatcher - scalability tested up to 10K ts/day
- filtering - local disk I/O - no limit at 1 TB/day
- sinking - stable up to 1 GB/day

Conclusions

- GSTAR is a stable production tool
- G3 geometry is based on MySQL DB
- G3 geometry is available as ROOT classes
- Looking for a G4 interface