ATLAS Prototype Framework
May 2000 Milestone

Craig E. Tull

NERSC/LBNL

ATLAS Software WorkShop @ LBNL

May 9-13, 2000

Outline

- Motivation
- Timeline
- Feature Set
- Tutorial

Realities of ATLAS Computing

- Large Data Volume
- Large, Globally Distributed Collaboration
- Long Lived (>15 years) Project
- Large (>2M LOC), Complex Analyses
- Distributed, Heterogeneous Systems
- Reliance on Commercial Software & Standards
- Evolving Computer Industry & Technology
- Object Oriented Programming
- Legacy Software
- Legacy Software Programmers
- Limited Computing Manpower
- Most Computing Manpower are not Professionals

What is a Framework?

- The Unified Modeling Language User Guide by Booch, Jacobson, Rumbaugh
  - —Framework: An architectural pattern that provides an extensible template for applications within a domain.
- Architecture TaskForce (ATF)
  - —Framework: A skeleton of an application into which developers plug in their code, using mechanisms defined by the framework.
  - —Toolset: A collection of functionality, implemented as subroutines and functions, or classes.

Proposed Major Milestones

- May 2000 Prototype
- Jun 2000 Alpha Release Design Review
- Sep 2000 Alpha Release
  - —Incorporate USDP feedback
- Dec 2000 Beta Release
  - —GEANT 3 Simulation
- Mar 2001 Freeze Full Function Release
- Jul 2001 Full Function Release Design Review
- Oct 20

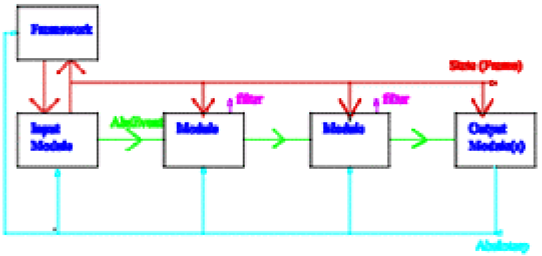
- 01 Full Function Release (Collocated)
- Apr 2002 Freeze Production V1 functionality

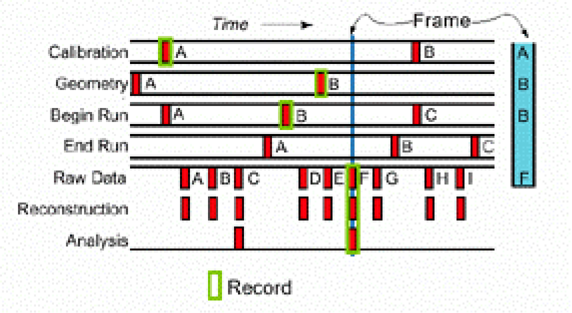
May 2000 Prototype

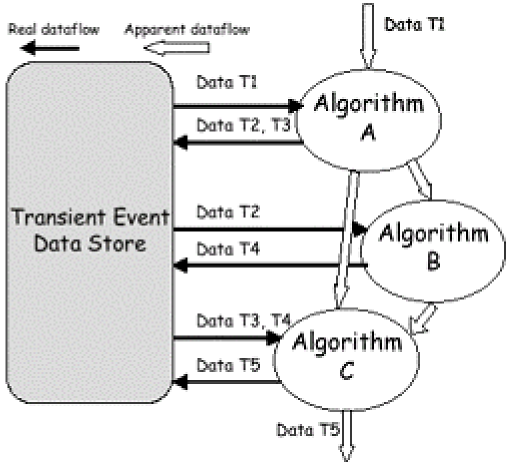
- Prototype Pre-Alpha Framework & Infrastructure
  - —Demonstration of basic functionality
  - —Not easily usable for more than simple tasks
  - —Major user interfaces will approximate final
    - Incremental changes (mostly extensions) will be unavoidable - Tools to aid migration
- May 9, 2000 - Framework Tutorial at ATLAS Software Week.
  - —Early introduction of and feedback from "typical users" (physicists & developers).
  - —Users will be able to understand both current implementation and future plans.

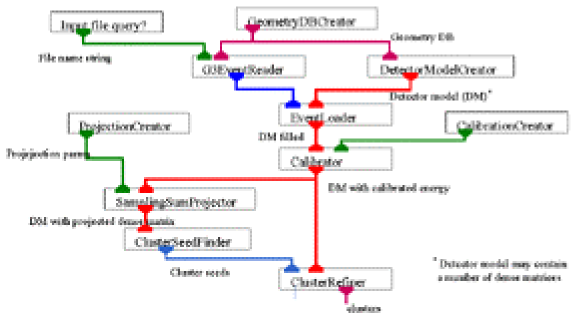
Framework Design Classifications

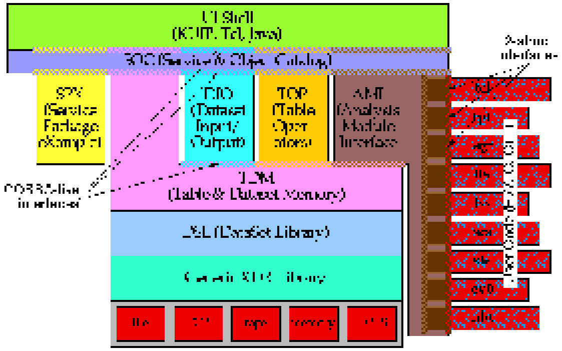
- **Finite State Machine - AC++**
- **Action on Demand - CARF**
- **Stream/Record/Frame - CLEO**
- **Simulated Data Flow - Gaudi**
- Mobile Agents - JAS
- **Object Network - ONCM**
- C++ Interpreter - ROOT
- **Software Bus - StAF**



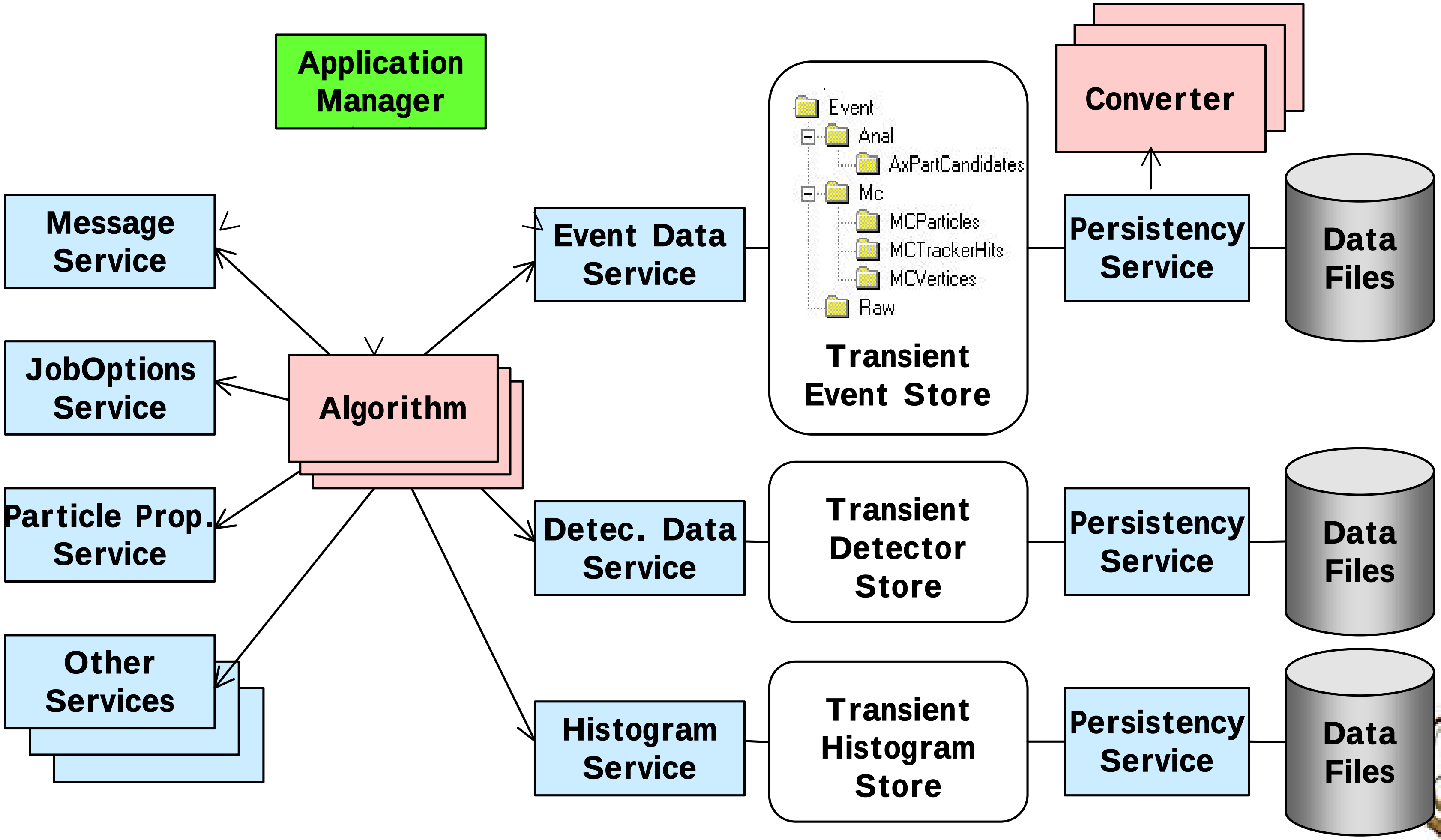







UI Shell
(KUIP, Tcl, Java)
SOC (Service & Object Catalog)
SPX (Service Package eXample)
DIO (Dataset Input/Output)
TOP (Table Operators)
AMI (Analysis Module Interface)
TDM (Table & Dataset Memory)
DSL (DataSet Library)
CORBA-like interfaces
file
tape
memory

GAUDI object diagram



May 2000 Prototype

- Assumption: Use GAUDI as starting point, add ATLAS event model, graphics, and other features.
- Feature set must include:
  - —Read Physics TDR data
  - —Execute sequence of multiple user modules
    - Write and Read from *Transient Data Store*
- Feature set should include:
  - —Generate Event Display
  - —Dynamic loading of user modules
  - —Sequences with branches/filters
- Feature set may include:
  - —Rudimentary interactive user interface
  - —Limited data persistence - HBOOK only

May 2000 Prototype

- Basic transient event store
  - —Evaluated BABAR/GAUDI/CDF versions
    - Decision to adopt GAUDI version
  - —Incorporate existing transient event model (RD Schaffer et al.)
- Allows existing ATLAS (PASO) reconstruction modules with only minor changes
  - — <10 lines of code change
- Sequencing of multiple algorithms/modules
  - —BABAR/CDF model of multiple paths each of multiple modules capable of filtering
    - Hypothesis-based processing
      - —Each path corresponds to a physics signal

Gaudi Components : May2k

- Most Gaudi components are being used as-is for May 2000 prototype.
- Typically, GAUDI users write Converters and Algorithms
- For May 2000 prototype, users do not need to write individual converters for data input. Rather access to Physics TDR data will be through RD Event Model.
- Many Algorithm components will be adapted PASO modules.

- Further development based upon GAUDI-like interfaces may have ATLAS implementations.

May2k Limitations

- Linux-only implementation
  - —RedHat 6.1, ATLAS Release 0.0.41, Gaudi R3
- GAUDI binaries provided as external packages.
  - —SRT "pseudo-package"
  - —SRT-based method for building executables and user modules.
- Documentation will be spare
  - —May Tutorial web pages + GAUDI
- Some interfaces may have limited functionality.
  - —Minimize any non-trivial changes to user code
    - Global edits, typedefs, and/or #includes are trivial

May2k Timeline (Mar 9)

- Feb 15 - Start
- Mar 23- Definition of ATLAS Algorithm/Analysis Module Interface
- Apr 07 - Integration of Physics TDR Data Model
  - —R.D.'s Event Model
    - Implicit Detector Description -No Explicit Service
- Apr 21 - Run Multiple Modules in sequence
  - —Use of Transient Data Store for communication and interchange
- May 01 - Integration of Event Display
  - —Simple IAlgorithm-Like Interface
- May 09 - Tutorial at Software Week

May2k Proto Timeline

- Feb 15 - Start
- Mar 9-11 - Architecture Workshop
- Mar 24 (Mar 23) - ATLAS Algorithm I-face
- Mar 31 (Apr 07) - Physics TDR Data Model
- Mar 31 (Apr 21) - Run Multiple Modules
- Apr 3 - SRT compile Framework
- Apr 6 - LArC OORecon @ LBL/BNL (SR,HM,TW)
- Apr 14 - Framework & Modules @ CERN
- Apr 14 (Mar 31) - Gaudi V4 Released
- Apr 17-19 - Architecture Workshop
- Apr 27 (May 01) - Integration of Event Display
- May 09 - Tutorial at Software Week

Feature set must include:

- Read Physics TDR data
  - —DONE
  - —Works on RedHat 5/6 with Release 0.0.40/41 at LBL, BNL, CERN
- Execute sequence of multiple user modules Write and Read from *Transient Data Store*
  - —DONE
  - —Works on RedHat 5/6 with Release 0.0.40/41 at LBL, BNL, CERN
  - —LArC OORecon example

Feature set should include:

- Generate Event Display
  - —DONE
  - —Framework @ CERN Apr 14
  - —Julius started Apr 17, finished Apr 27
- Dynamic loading of user modules
  - —Shared linking yields most benefits
  - —Gaudi Release 4 has
- Sequences with branches/filters
  - —DONE
  - —Integration into code base Apr 21

Feature set may include:

- **Rudimentary interactive user interface**
  - —SWIG generated Tcl interface
  - —Trivial execution of standard event loop from Tcl
  - —Still work in progress
- **Limited data persistence - HBOOK only**
  - —DONE
  - —Gaudi HistogramSvc & HistogramPersSvc

Tutorial Agenda

Presentations/Exercises by Paolo and Charles

- 09:00 —Introduction & Overview (Craig/David)
- 09:30 —HelloWorld
- 10:30 —Break
- 10:45 —Transient Data Store & mult. Algs
- 12:30 —Lunch
- 14:00 —Sub Algs / Composite
- 15:00 —Histograms and Ntuples
- 15:45 —Break
- 16:00 —Event Display (Julius)
- 17:00 —Porting from Paso (Srini)
- 18:00 —Adjourn

Methodology

- Short presentations
- Emphasis on exercises
  - —start with an empty file (or a simple skeleton)
  - —work towards a fully featured algorithm
  - —new concepts and components introduced at each (sub)step along the way
- Distribute and document examples for self-study as well

Example 1: Part 0

Setup Build Environments

—> source /auto/atlas/tools/WorkshopSetup.csh

- Work area: `~/Workshop`

  —> cd ~/Workshop

- source files are in

  —src41/ControlExamples/GaudiTutorial/src

- header files in

  —src41/ControlExamples/GaudiTutorial/GaudiTutorial

- build from `~/Workshop/build41`

  —> gmake install

- execute from `~/Workshop/run`

  —> ./tutorial_examples < atlas.datback > output

- solutions to examples in

  —~/Workshop/solutions/ex1/part1 ...

Example 1: Part 0
Initial Files

- For each example, copy initial files from

  —~/Workshop/solutions/ex(n)

- copy source files (*.cxx) to your source file area.

  — > cd ~/Workshop/src41/ControlExamples/GaudiTutorial/src

  — > cp ~/Workshop/solutions/ex1/*.cxx .

- copy header files (*.h) to your header file area.

  —> cd ../GaudiTutorial

  — > cp ~/Workshop/solutions/ex1/*.h .

- copy GNUmakefile.in

  — > cd ..

  — > cp ~/Workshop/solutions/ex1/GNUmakefile .

- copy jobOptions.txt to run area.

  —> cd ~/Workshop/run

  —> cp ~/Workshop/solutions/ex1/jobOptions.txt .

May2k Early Adopters

- Real users provide the only believable:
  - —Q & A Feedback
    - Regression testing never finds all bugs
  - —Design Feedback
    - Unusable elegant design useless
  - —Performance Measurements
    - Real code doing real work is only way to measure real performance
- Needed a few "Friendly Users" in first phase willing to put up with rough edges & incomplete interfaces while framework code develops.
  - —Careful to avoid premature support diverting effort from development

Other Adopters

- Liquid Argon Reconstruction
  - —Srinir Rajagopalan, Hong Ma, Torre Wenaus
- Inner Detector Pattern Reconstruction
  - —Laurent Vacavant
- Event Generator(s)
  - —Ian Hinchliffe
- TileCal Test Beam
  - —David Malon
- SCT-Kalman
  - —Werner Wiedermann
- ATLFAST
  - —Peter Clarke, Hywel Phillips

Future Releases

- September 2000
  - —Merge two development strands
  - —Wrapped FORTRAN
  - —Event Model
  - —Limited database integration
  - —Run-time configuration
  - —Limited Physics Analysis output – Histograms/NTuples
- December 2000
  - —Geant3 Simulation integration
  - —Targeted towards Trigger TDR (Summer 2001)

Future Releases (2)

- October 2001
  - —Full Database integration
  - —Geant4 Simulation integration
  - —Physics Analysis Tool integration
  - —Visualization
  - —Statistics & Monitoring tools
  - —Bookkeeping
- Intermediate releases at 3-4 month intervals

Conclusion

- May 2000 Prototype
- Jun 2000 Alpha Release Design Review
- Sep 2000 Alpha Release
  - —Incorporate USDP feedback
- Dec 2000 Beta Release
  - —GEANT 3 Simulation
- Mar 2001 Freeze Full Function Release
- Jul 2001 Full Function Release Design Review
- Oct 2001 Full Function Release (Collocated)
- Apr 2002 Freeze Production V1 functionality
  - —Distributed Capability
- Expect minor releases at ~3-4 month intervals