LHCb Computing Status Report

Meeting with LHCC Referees

March 24th, 1999

John Harvey

CERN/ EP-ALC

Outline

- Status of the LHCb simulation program (SICB)
- News on computing facilities used by LHCb
- GAUDI
  - Important milestones since since Oct 1998
  - Architecture review
  - First release of framework, progress on algorithms
  - Implementation issues
  - Programme of work in 1999; plans for future releases
- Training - LHCb OO programming course
- LHCb Software Weeks
- Summary

Status of SICB

- Version 117 beginning of March ‘98
  - ➢ Infrastructure to plug in any event generator (HEPEVT)
  - ➢ LHC proton beams : angles and smearing (trigger studies)
  - ➢ Luminosity handling (multiple interactions per beam crossing)
  - ➢ Updated geometry for Muon Detector and shielding
  - ➢ Magnetic Field Map for new conical magnet design (CERN)
    - ➥ Field less uniform - study impact on μ trigger and tracking
- Port to Windows NT completed
  - ➢ production environment setup on PCSF, still being optimised
  - ➢ ~100 k events simulated since beginning of March
  - ➢ can produce ~50k events per day

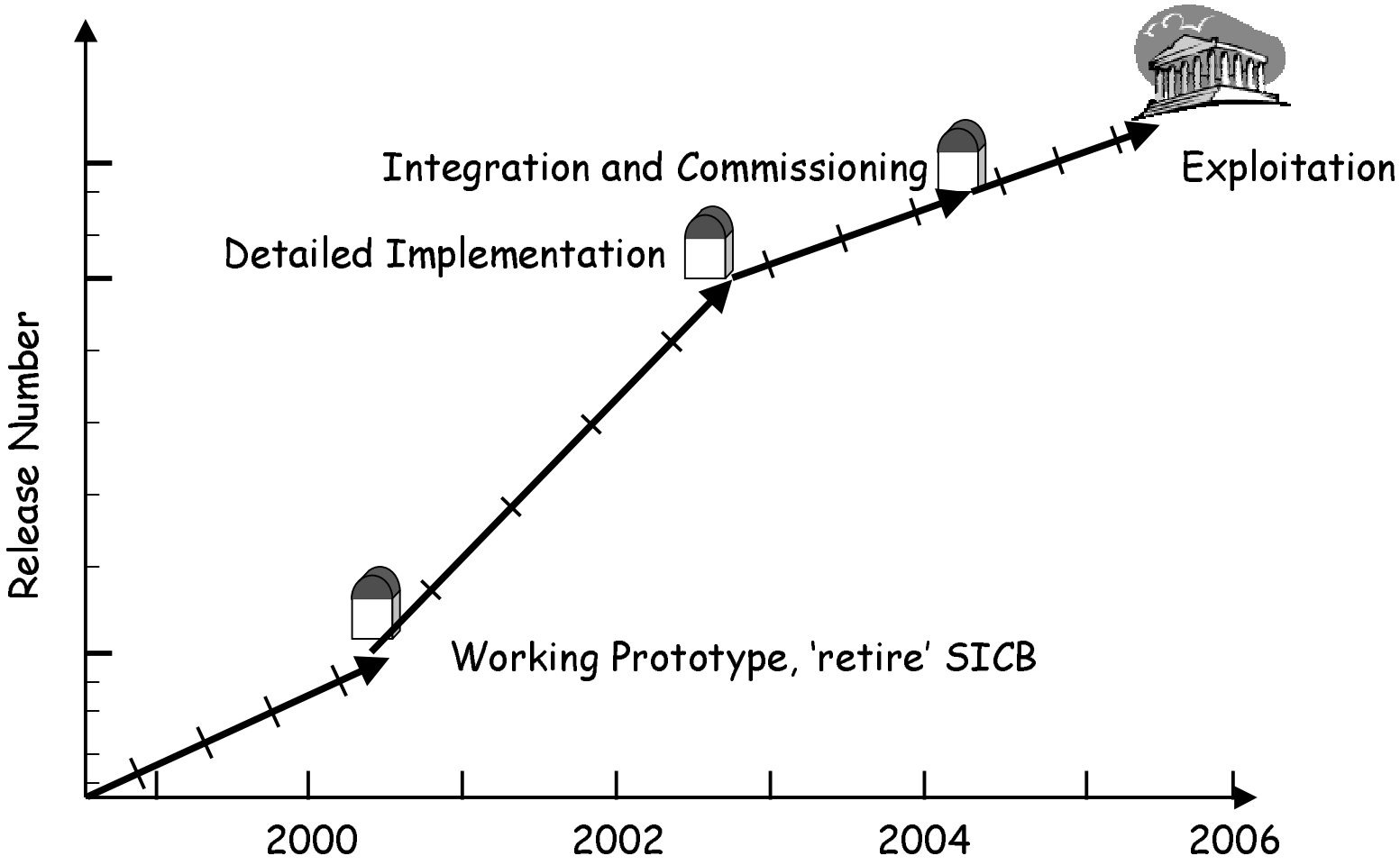
SICB - plans for future releases

- New b-decay generator will be integrated (QQ package - CLEO)
- New vertex detector layout (27 stations, 4 ϕ sectors) - for late 1999
- RICHes
  - ➢ Improve fast parameterization (no background, no pattern recognition)
  - ➢ Full digitization and pattern recognition - study extended tracking and CPU needs
- Calorimeters
  - ➢ Projective geometry for Preshower, ECAL and HCAL
  - ➢ More accurate GEANT Simulation - lower thresholds and full sampling
  - ➢ New trigger code : 2x2 algorithm and 3x3 algorithm in the 4/12 scheme
- Muon Trigger background - showering in μ shielding and neutron capture
- New beam pipe design

Computing Facilities

- Monte Carlo production
  - PCSF (CERN) - NT
  - IN2P3/ Lyon - UNIX
  - RAL - UNIX and NT
- Collaboration facilities
  - Liverpool - 300 node PC/Linux farm under development
  - Rio - PC/Linux farm
  - Moscow - PC/Linux farm
- Analyses use public batch facilities at CERN (RSPLUS)
  - must envisage private capacity (SHIFT) in 2000

LHCb Offline Software Road Map



Strategy for development of new software

- We are convinced of the importance of the architecture
  - ➢ architect (experienced designer) and design team (domain specialists)
- Identify components, define their interfaces, relationships among them
- Build framework from implementations of these components
  - ➢ “framework is an artefact that guarantees the architecture is respected”
  - ➢ to be used in **all** the LHCb event data processing applications including : high level trigger, simulation, reconstruction, analysis.
  - ➢ Build high quality components and maximise reuse
- Incremental approach to development
  - ➢ new release every two months
  - ➢ gradually add functionality
  - ➢ use what is produced and get rapid feedback

Important Milestones since Oct ‘98

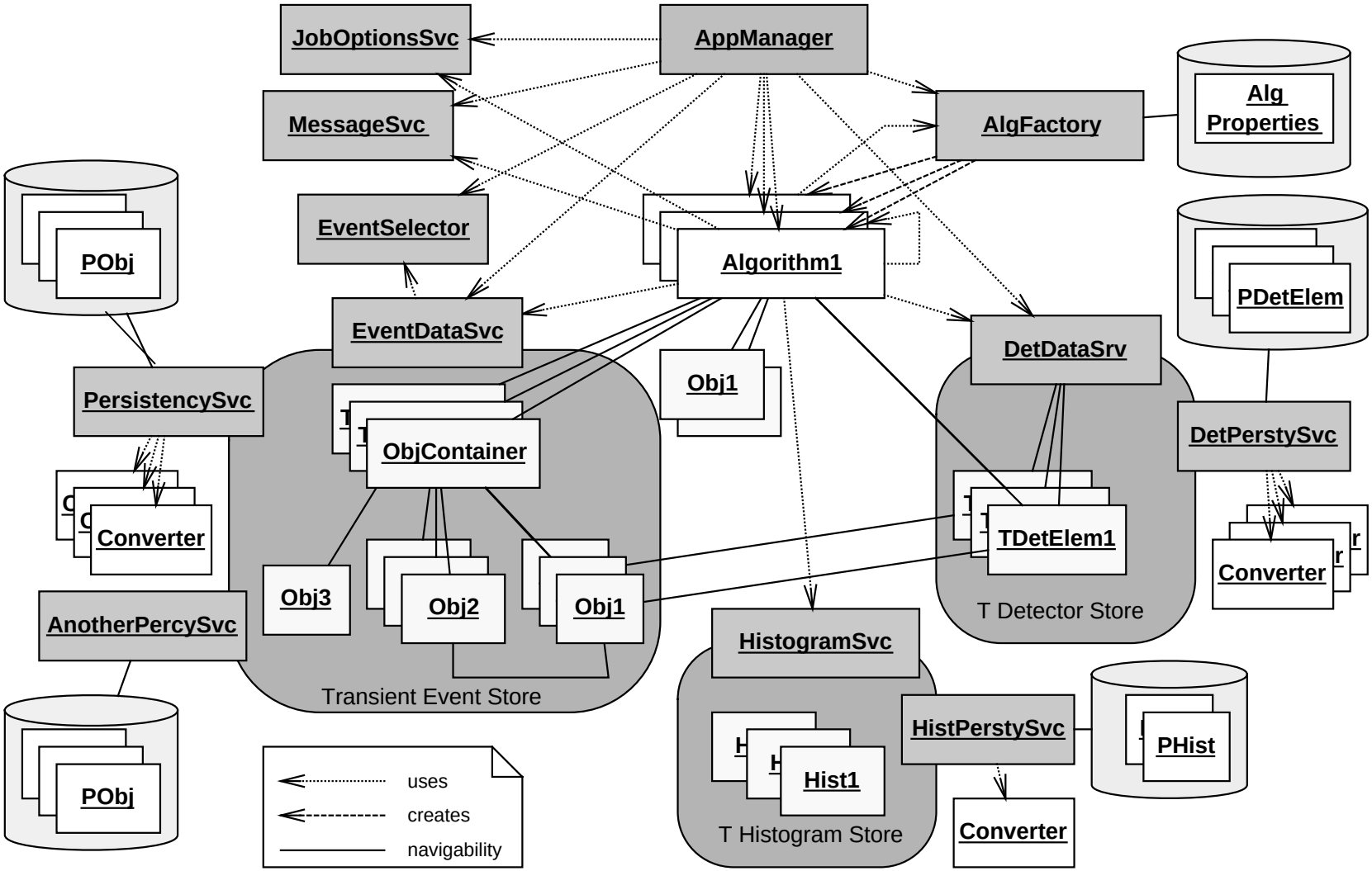
- Sept ‘98 - architect appointed and software design team
- Nov 25 - Review of LHCb software architecture (GAUDI)
  - ➢ agreement on components to be implemented in version 1
- Dec 7-11 - First LHCb course in OO Analysis & Design
- Jan 18-22 - Second LHCb course in OO Analysis & Design
- Feb 5 - Release of first version of GAUDI framework
- Feb 8-12 - First LHCb Software Week
  - ➢ Work programme agreed for version 2 of GAUDI
  - ➢ Five new members of the GAUDI team to tackle next phase

Architecture Design

GAUDI

General **A**rchitecture for **U**nified **D**ata **I**nterfaces

LHCb Software Architecture - GAUDI



Major Design Criteria

- Clear separation between “data” and “algorithms”
- Three basic types of data:
  - **event data**
  - **detector data** (structure, geometry, calibration, alignment,..)
  - **statistical data** (histograms, …)
- Clear separation between “persistent” and “transient” data
  - Isolation of user’s code
  - Different/incompatible optimization criteria
  - Transient as a bridge between various representations
- Data Store centered architectural style
  - Algorithms as data producers and consumers
- *User code* encapsulated in few specific places:
  - “Algorithms”: Physics code
  - “Converters”: Converting data objects into other representations

Classification of classes

Application Managers	One per application. The "chef d'orchestra".
Services	Offering specific services with well-defined interfaces. Different concrete implementations depending of specific functionality.
Algorithms	Physics code. Nested algorithms. Simple and well defined interface.
Converters	In charge of converting specific event or detector data into other representations.
Selectors	Components to process a selection criteria for events, parts of events or detector data.
Event/Detector data	The data types that the algorithms and converters are using. No complex behavoir.
Utility classes	All sort of utility classes (math & others) to help on the implementation of the algorithms.

Architecture Review

- Review took place on Nov 25th with external reviewers
- Were goals met?
  - Force preparation for the review - Documentation!
    - This was done - all documents available via web
  - Validation of the requirements
    - many use cases evaluated
  - Evaluate early before it becomes a “blueprint” for software
  - Determine where finer grain depictions needed
    - document global knowledge, object relationships are a problem, monitoring state of application must be envisaged, ….
  - Disseminate ideas on what constitutes a good architecture
    - very positive feedback from ATLAS, STAR,…
  - Determine whether can proceed to development
    - YES - deliver something to end users - be prepared to redesign parts

GAUDI Framework Status

- Version 1.0 was released on Feb 5th
- Level of functionality provided:
  - Application Manager is complete
  - Event data service allows existing SICB events to be read
  - Transient event data service allows events to be viewed within a C++ framework and an OO LHCb event model
  - Histogram data service : create, store, retrieve histograms
  - Histogram persistency service : only HBOOK data files so far
  - Implementation of basic services : Job Options, Message,…
- Composed of:
  - Libraries (WNT 4.0, IBM AIX 4.1.5 & 4.3, HP-UX 10.20, Linux RedHat 5.1)
  - Example code
  - Documentation: User Guide, Reference Manual
  - URL: http://lhcb.cern.ch/computing/Components/html/GaudiMain.html

Algorithms

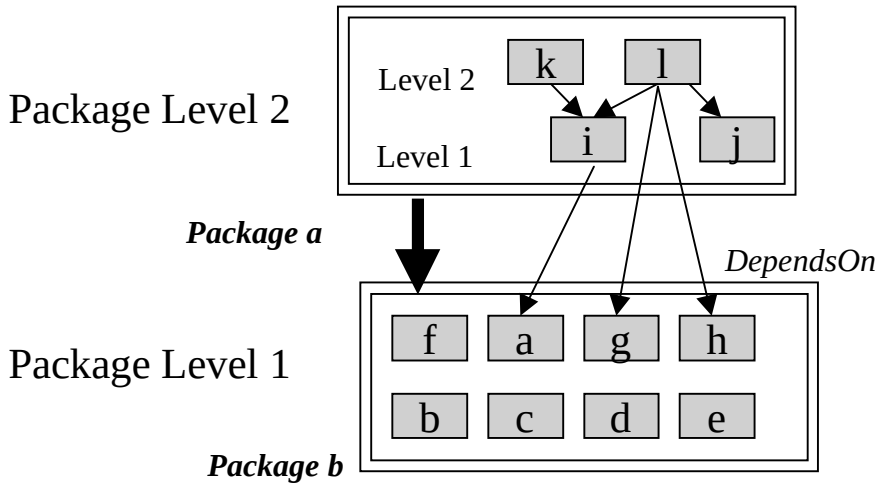
- RICH detectors
  - Goal - re-implement the existing pattern recognition algorithms in OO
  - Complex problem - good test
  - Model radiators, mirrors, detectors ; tracks, pixels, photons
  - OO design made, implementation to be completed soon
  - Compare with FORTRAN algorithm : understandability, cpu usage….
  - Next steps…integrate with GAUDI
- Muon detector
  - take relatively simple piece : digitisation
  - make complete analysis, design and code
  - design made, implement and test soon
  - repeat procedure for reconstruction and trigger
- First ideas presented on Tracking, Calorimetry and Analysis

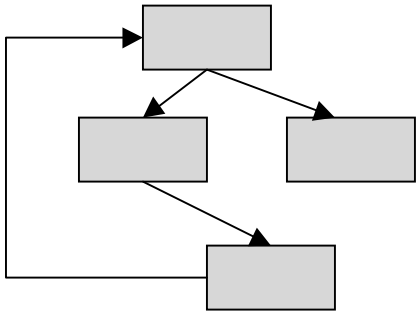
Implementation Issues

- Packages
- Runtime libraries
- Visual Developer Studio on NT
- Code repository - CVS
- Access to code repository from NT - WinCVS
- C++ coding conventions (LHC common project)
  - ➢ specification document to be finalised soon
  - ➢ code check utility to verify rules (36 rules coded so far)
- Software Release Tool
  - ➢ currently use CMT (Orsay)
  - ➢ following progress of SPIDER/SRT project
- Documentation tool

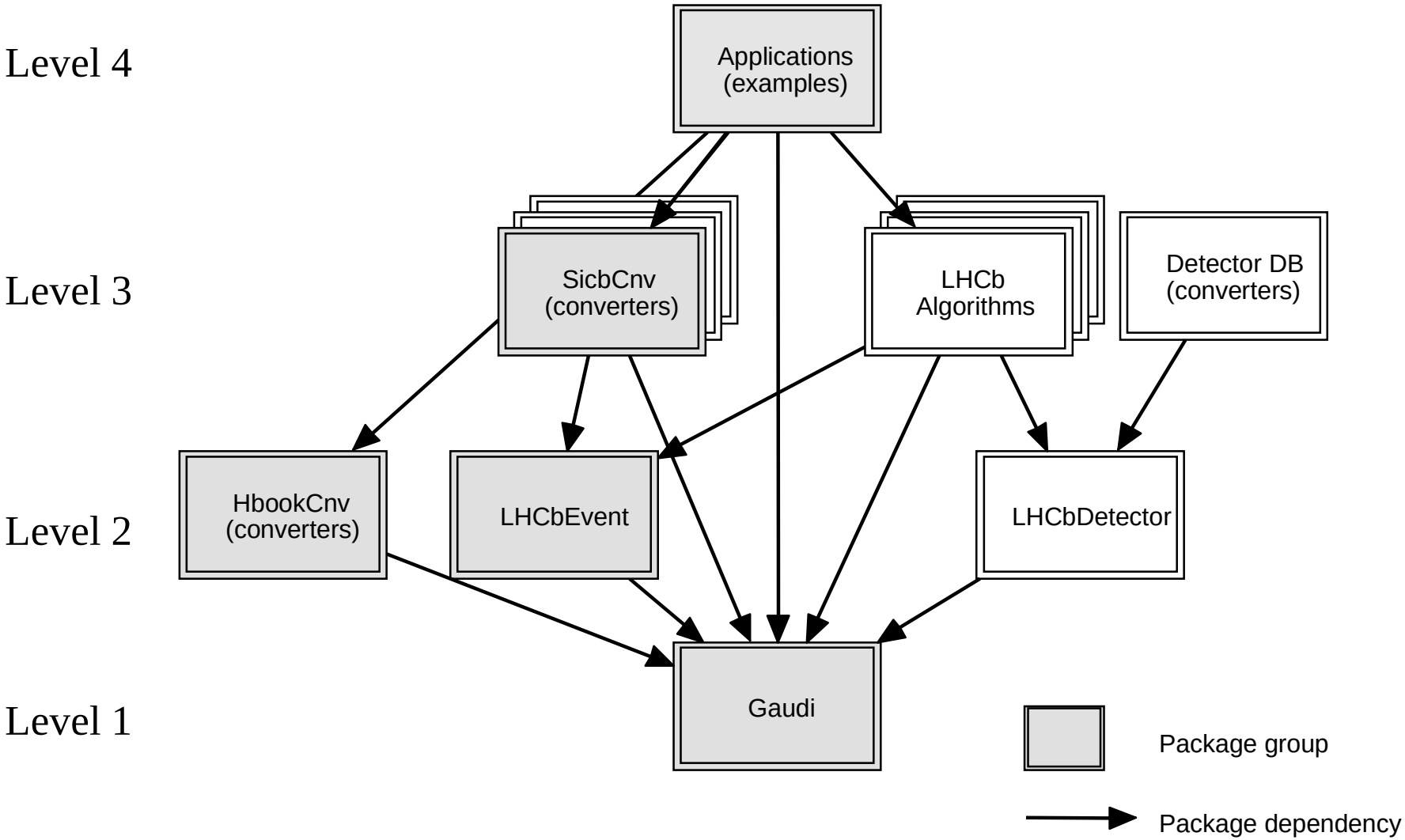
Physical design - Packages

- For large software systems is important to decompose into hierarchies of smaller more manageable entities.
- The physical decomposition has big consequences on compilation time, link dependencies, configuration management, executable size, etc.
- Need a macro unit of **physical design** referred to as a **package**
- Follow rules - avoid cyclic dependencies





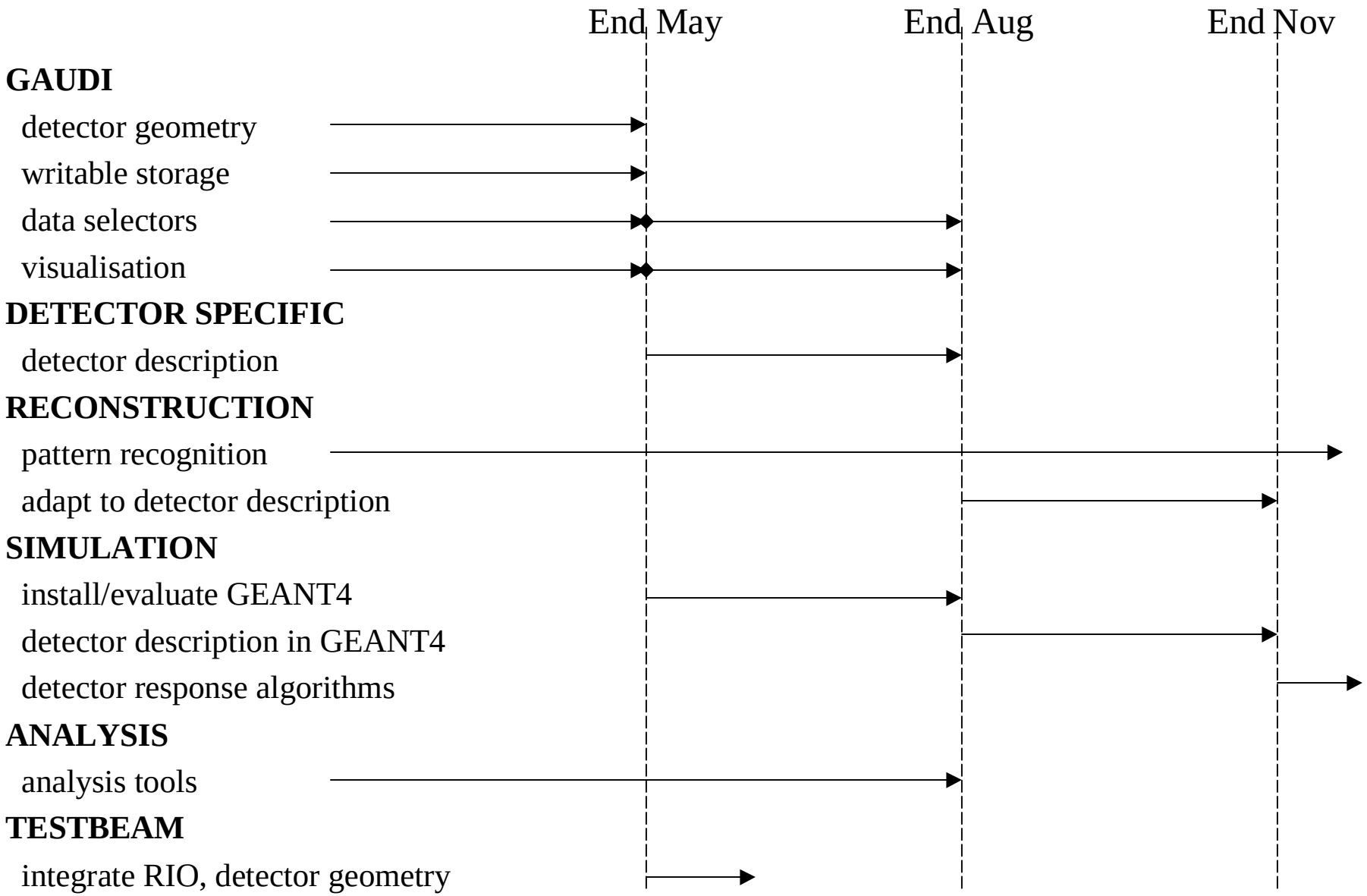
Package Structure



Next version of GAUDI

- Consolidate what we've done
  - Improvements to algorithm interface, histogram interface, message level handling, . ..
  - Need to validate critical design decisions (e.g. the separation between transient and persistent data), for example by measuring impact on performance
- Start adding new components
  - Libraries
    - study what exists (NAGC, clhep, STL,…)
    - make recommendations and guidelines
  - Detector description and geometry - a generic model plus subdetector specifics
  - Writable storage to be able to store results
    - solutions are: ROOT I/O (now), Objectivity(later)
  - Visualization and interactivity
    - The candidate solutions are: ROOT, WIRED/JAS (Java), Open Scientist (OpenGL, OpenInventor,…)
    - We will integrate these 3 solutions with the Gaudi Framework and evaluate

Software Work Programme in 1999



LHCb OO Programming Course

- Five day course held at CERN Dec 7-11, Jan 18-22
- Covers OO Analysis and Design, and hands-on programming
- Establish use of common methods and notation
- 16 people per course, total of ~40 now trained
- Now added to CERN OO training curriculum

Agenda for First Software week

Day		Topic	Attendance Total/CERN based
Monday	p.m.	Tools	25 / 20
Tuesday	a.m. p.m.	GAUDI GAUDI Tutorial	27 / 22
Wednesday	a.m. p.m.	Data Analysis Tools GAUDI Tutorial	22 / 16
Thursday	a.m. p.m.	Algorithms Plan work programme	21 / 18
Friday	a.m. p.m.	SICB Projects	27 / 12

Software weeks in 1999 planned for June 2-4, Nov 24-26

Summary

- First version of new GAUDI framework available
- Development of pattern recognition algorithms using GAUDI
  - ➢ waiting feedback, new ideas, adapt as required
- New components being added which will allow GAUDI to be used as a real reconstruction and analysis tool
- Start projects for each application program
  - ➢ starting with reconstruction
  - ➢ project leader to organise regular working sessions as required

Practical Experience (Niko Neufeld)

- Importance of books, training, trial and error
- Tools - powerful but complex (Rose)
- Libraries - NAGC, clhep, STL
- Steep learning curve - spend lot of time in analysis & design
- Reuse existing solutions…design patterns

DAQ Status - Outline

- Requirements and Architecture, TP numbers, review, workshop, update
- Readout Network
  - problem statement
  - assembling large networks from small switching components
  - recovery of scalability - traffic shaping, intermediate buffers
  - strategy and plans - type of control, configuration size, calculation, simulation
  - prototypes
- Readout Unit - describe prototype design
- SFC - use of intelligent network interfaces
- Myrinet studies
  - results from prototype
  - results from simulation
- Studies of Gbit ethernet and SCI planned or underway
- TFC - status of technical note - missing manpower still
- JCOP - concerns about SCADA project