

# AIDA - Abstract Interfaces for Data Analysis



Andreas Pfeiffer, CERN/IT

**`andreas.pfeiffer@cern.ch`**



# Outline

---

- Motivation
- Architectural issues
- AIDA Workshop Apr-30/May-01
- AIDA and Geant-4
- Future Plans



# The AIDA project

- AIDA project (Abstract Interfaces for Data Analysis) was initiated at the HepVis'99 workshop
  - since then several (mini-) workshops (main ones Paris 2000 and Boston 2001) have been organized
- Presently active mainly developers from existing packages
  - JAS (Tony Johnson)
  - Lizard (Andreas Pfeiffer)
  - OpenScientist (Guy Barrand)
  - Wired (Mark Dönszelmann)



# Introduction

- Design Interfaces for Data Analysis (in HEP)
  - *"The goals of the AIDA project are to **define abstract interfaces for common physics analysis tools**, such as histograms. The adoption of these interfaces should make it easier for developers and users to select to use different tools without having to learn new interfaces or change their code. In addition it should be possible to exchange data (objects) between AIDA compliant applications."*
- **Open** for contributions of any kind
  - suggestions, code, implementations, ...



# Motivation

## ■ Abstract Interfaces

- Only **pure virtual** methods, inheritance only from other Abstract Interfaces
- Defines a kind of a “protocol” for a component

## ■ Components **use** other components **only** through their Abstract Interface

- Minimize coupling between components
- Provide flexibility to interchange implementations of these interfaces
- Allow for faster turn-around time
- Allows and try to re-use existing packages

# Architectural issue: Components (I)



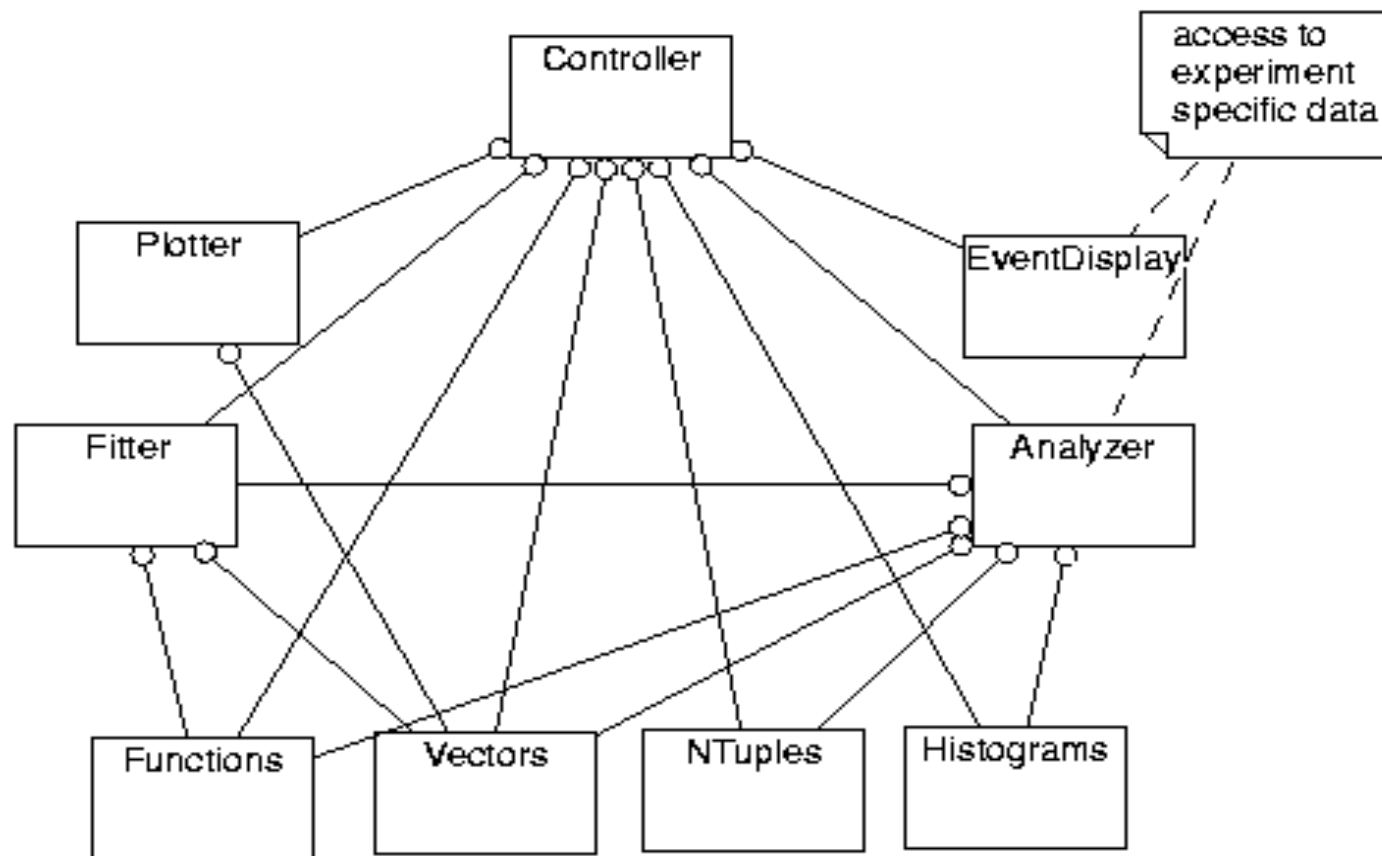
- Identify components by **functionality**
  - not by “historic use”
- Emphasize separation of **different aspects** for each **component**
  - example: Histogram
    - **statistical entity** (density distribution of a physics quantity)
    - **view** of a “collection of data points” (which *can* be a density distribution but also a detector efficiency curve)
    - **command** to manipulate/store/plot/fit/...
- Identify and **use patterns - avoid anti-patterns**
  - learn from other people’s experiences/failures

# Architectural issue: Components (II)



- “User’s view” is different from “implementor’s view”
  - separate Abstract Interfaces for both aspects
  - “command-layer” vs. “implementation-layer(s)”
- **UserInterface** as a separate **component**
  - by definition couples to most of the other components
    - *Facade* pattern
  - promotes **weak coupling** between the other components
  - interfaces to scripting and/or GUI

# Initial Categories and dependencies







# History

- started in Sept. 1999 (HepVis'99, Orsay)
- release 1.0 summer 2000
  - concentrated on “developers view”
    - Histogram package only
      - IAxis, IHistogram, IHistogram1D, IHistogram2D, IHistogramFactory
- Used by Gaudi/Athena (LHCb, Atlas, Harp)
  - Gaudi people involved in design
- Adopted and used in Geant-4 examples
  - new category created in Geant-4 for analysis using AIDA interfaces

# AIDA Workshop in Fall 2000 (Orsay)



- Start working from “users view”
- Concentrated discussions on
  - Histogram
    - | “developer” vs. “user” view
  - Management
    - | directory like Tree
  - Store
  - Tuple
- About 20 Interface classes were defined
  - written down in Java and checked into FreeHep-CVS

# AIDA Workshop Apr 30/May 1



## ■ Agenda

- Show and Tell
- Infrastructure
- Discuss future development

## ■ Very constructive

- concentrated on open issues from Paris workshop and Infrastructure



# Infrastructure (I)

## ■ names of C++ files

- `<InterfaceName>.h` for Interfaces
  - e.g., `IHistogram.h`

## ■ location of repository

- Anonymous CVS access
- `:pserver:anoncvs@cvs.freehep.org:/cvs/aida` (passwd *aida*)
- module: `aida`
- ( `java/hep/aida` | `cpp/AIDA` | `web` | `doc` | `xml` | `tools` )
- no `<pkg>` subdirs; `cpp/AIDA/IHistogram.h`
- web pages in same repository



# Infrastructure (II)

## ■ “checkout area” includes tools ...

- `<sw/contrib>/AIDA/tools/ant-jarFiles`
- `<sw/contrib>/AIDA/build.xml`
- `<sw/contrib>/AIDA/GNUMakefile`
- `<sw/contrib>/AIDA/ReadMe`

## ■ ... to create “release area”

- `<sw/contrib>/AIDA/<version>/AIDA/IHistogram.h`
  - `#include <AIDA/IHistogram.h>`
  - `version : x.y.p`
- `<sw/contrib>/AIDA/<version>/lib/` (only for .jar files)
- `<sw/contrib>/AIDA/<version>/doc/`
- `<sw/contrib>/AIDA/<version>/xml/`
- `tar/zip` of “`AIDA/x.y.p/`” as `<sw/contrib>/AIDA/<version>/AIDA-x.y.p.tar.gz`



# Infrastructure (III)

- mailing list (archived) exists at CERN (<listName>@cern.ch)
  - project-aida (open) general discussion list
  - project-aida-dev (open) bug reports, developer specific issues
  - aida-announce (posting moderated, subscription open)
- cvs tag names:
  - [vXrYpZ](#) for releases
  - [build-xxx](#) (and "Paris-2000", "Boston-2001", ...)



# Status of Components

Histogram	close to complete, implementation exists
Tree	discussed, being implemented
Cloud	discussed, being implemented
Store	discussed, design close to complete
Tuple	discussed, design not complete
Function	discussion started
Fitter	discussion started
MasterFactory	discussion started
Plotter/Document	discussion not yet started



# Time schedule / release plan

- Release for **discussion** and **feedback**
  - even if not (yet) complete
- V 2.0.0 out mid May 2001 ("Boston release")
  - C++ and Java version of the Interfaces
  - patch-releases added if needed
- Aiming initially for 2-3 month release frequency
- Web page (<http://aida.freehep.org/>)
  - On web page links to implementations
    - from whoever provides one (and informs us)





# Analysis Tools

## ■ Java Analysis Studio

- started on reference implementation for AIDA-2.0.0
- JAIDA package to access Java components from C++
  - C++ proxies implement AIDA using Java Interfaces

## ■ Open Scientist

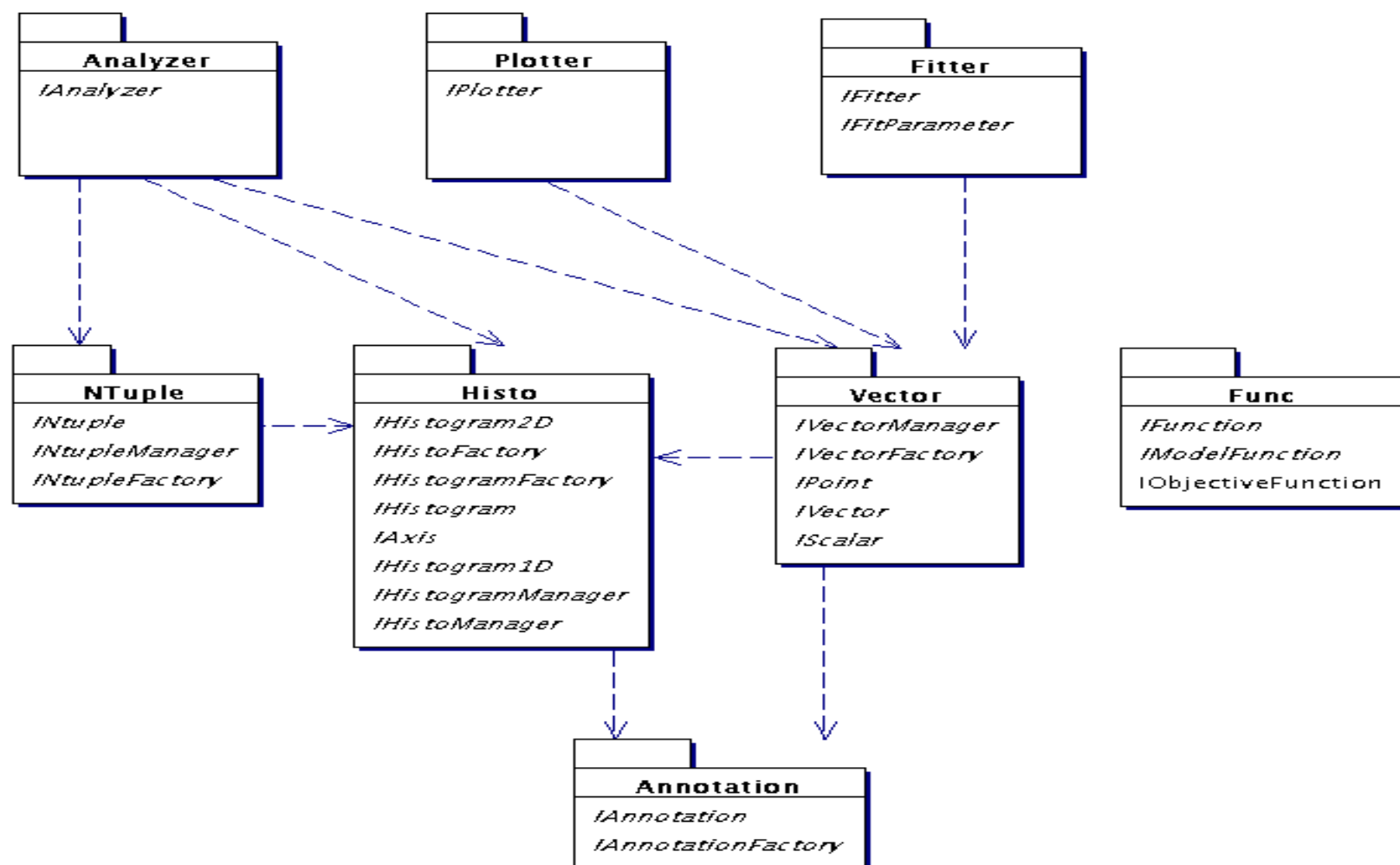
- plan to have full implementation by end 2001

## ■ Lizard / Anaphe

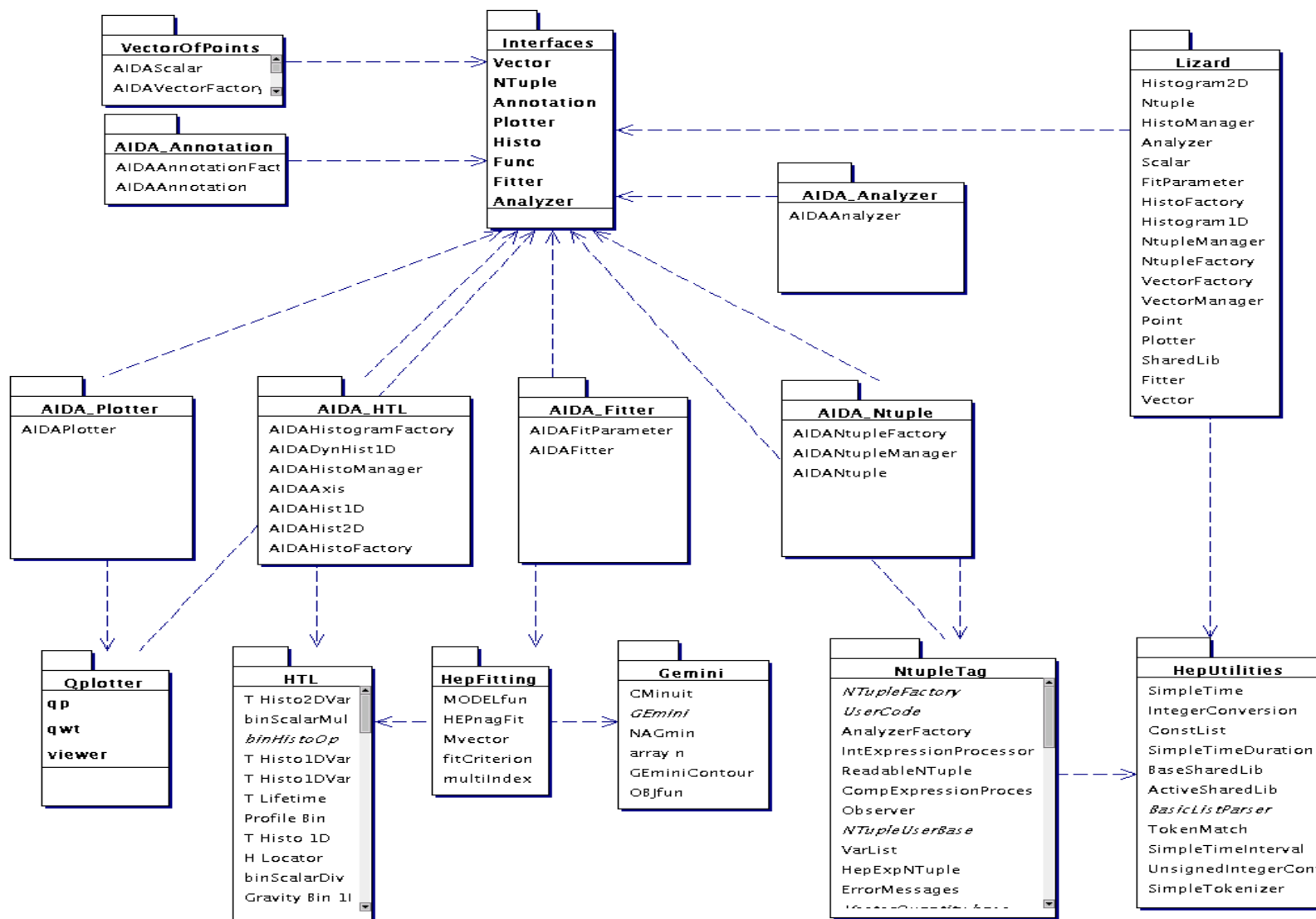
- starting on implementation in Aug. 2001
- partial implementation (AIDA-1.5 ?) earlier for G-4

## ■ Demo this afternoon for various systems

# Lizard Internals: Interfaces



# Anaphe: some components

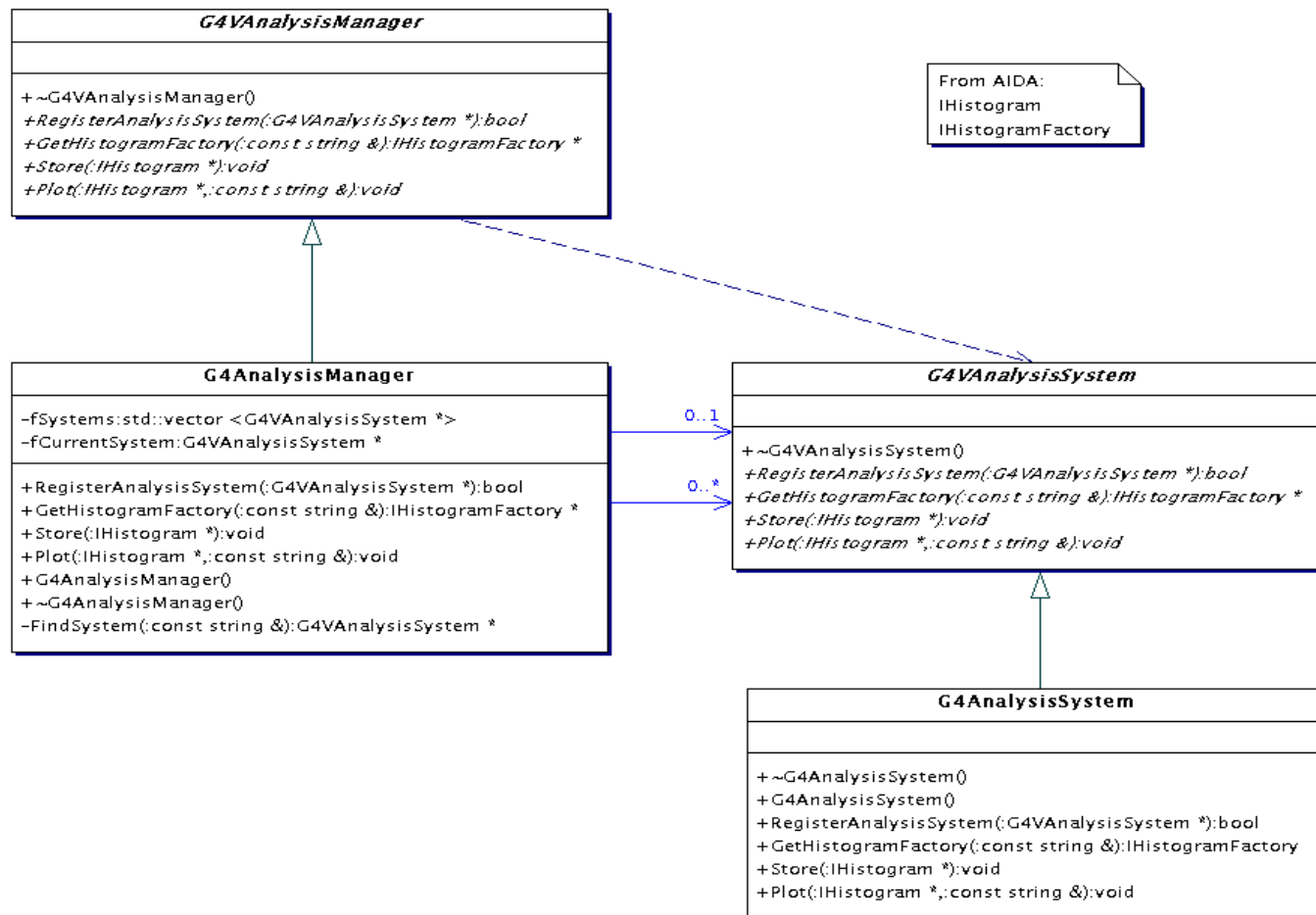




# Geant-4 Analysis

- Presentation by Guy Barrand tomorrow
- Uses AIDA histograms (Version 1.0)
- Need for “specific code” in G4AnalysisSystem
  - specific names for libraries implementing the Ifs
  - could go away with future “MasterFactory” (AIDA)
  - need to communicate with AnalysisTool from G4 appl.
- Concentrate on Ntuples and Factories at this workshop
  - “Underground” project parallel session

# Geant-4 Analysis Category





# Summary

- Design of Abstract Interfaces for Data Analysis
  - Maximize flexibility and re-use
  - Allow for faster turn-around time
  - Allows and try to re-use existing packages
- No need to go for “least common denominator”
  - use “reasonable” superset
  - concentrate on proper design
- Use in Geant-4 analysis category
  - several examples using AIDA with implementations provided by JAS, Lizard/Anaphe, OpenScientist

