



LCIO

Overview and Status

ECFA Workshop, Montpellier
November 14, 2003
Frank Gaede DESY -IT-



Outline

- Introduction
- Data model
- Software design
- Implementation
- Status/Features
- Customers/Users
- Summary

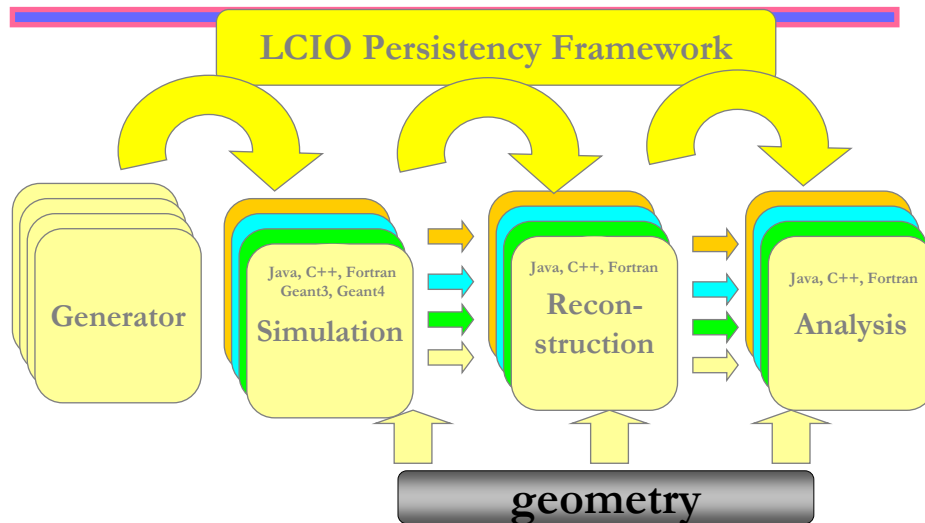


Introduction

- at Prague workshop decided to have **Data format/persistency task force:**
"Define an abstract object persistency layer and a data model for linear collider simulation studies until the Amsterdam workshop."
- -> **LCIO** – Linear Collider Input/Output
 - DESY/SLAC/LLR joined project
 - design of data model and software introduced at Amsterdam workshop
 - now production version

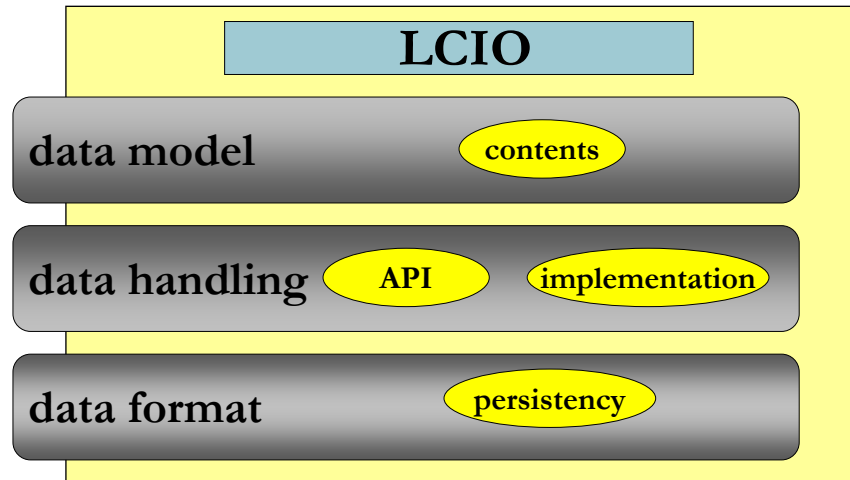


Motivation for LCIO





LCIO persistency framework

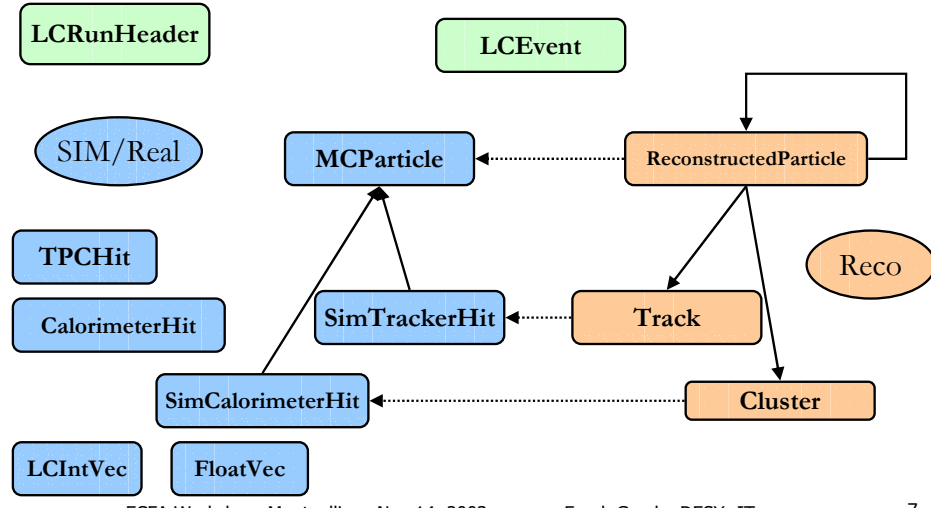


Requirements

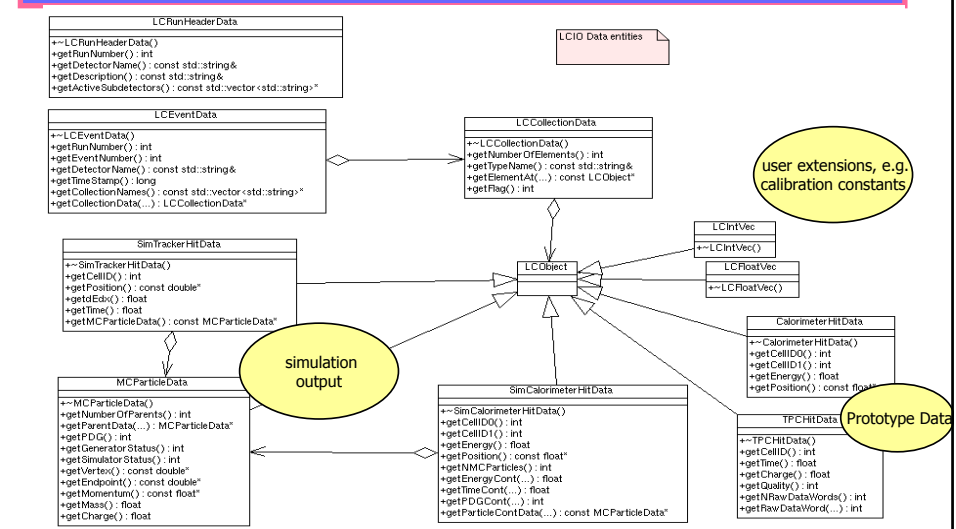
- need Java, C++ and f77 (!) implementation
- extendable data model for current and future simulation studies
- user code separated from concrete data format
 - -> want to be flexible for future decisions on persistency
- needed a.s.a.p.
 - > keep it simple (lightweight)



LCIO Datamodel

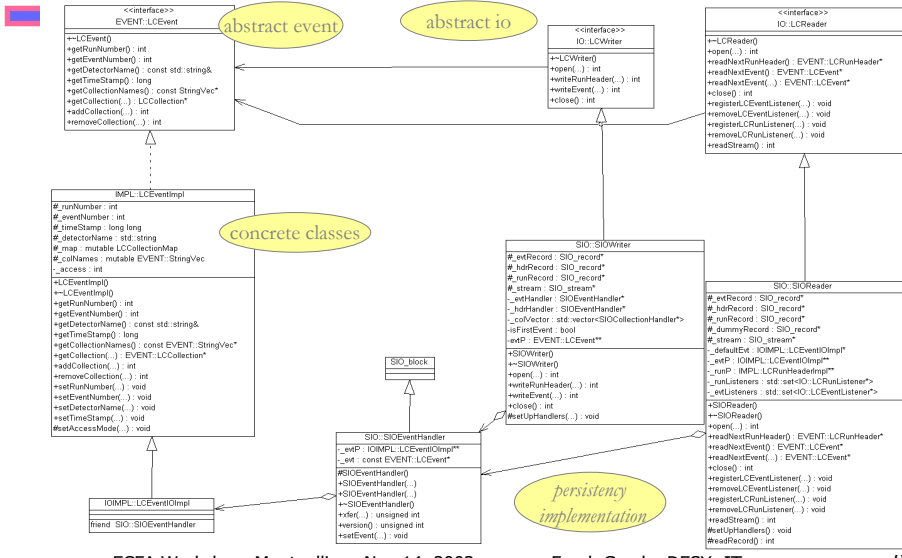


Data model - Design

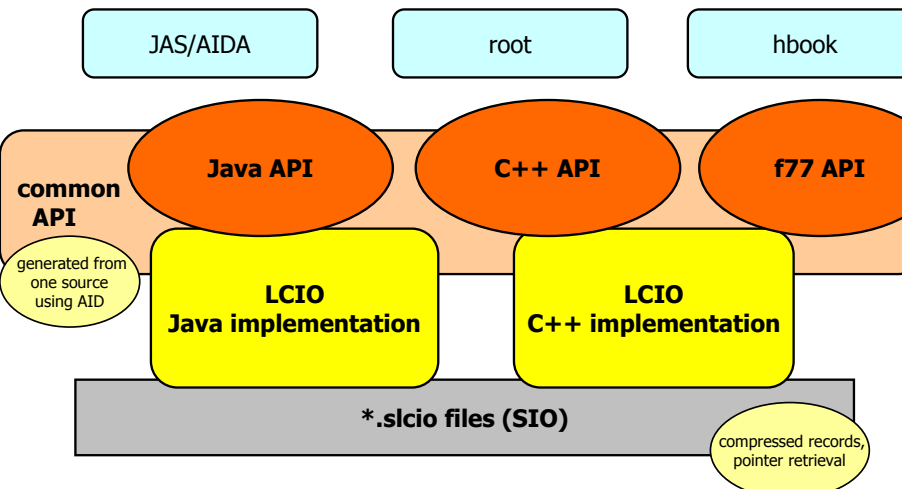




Implementation - Design



LCIO SW-Architecture





C++ and f77 example code

```
emacs@pcx3340.desy.de <7>
File Edit Options Buffers Tools C++ Help
// ---- event loop ----
const LCEvent* event ;
while( (event = lcRdr->readNextEvent()) != 0 ){

    int runNum = event->getRunNumber() ;
    int evtNum = event->getEventNumber() ;
    string detName = event->getDetectorName() ;

    std::cout << " run: " << runNum << std::endl ;
    std::cout << " evt: " << evtNum << std::endl ;
    std::cout << " det: " << detName << std::endl ;

}

//---- end event loop ----
ana.job.cc (C++ Abbrev)--L16-- 4%-----

emacs@pcx3340.desy.de <6>
File Edit Options Buffers Tools Fortran Help
c ---- event loop ----
do 10
    event = lrdreadnextevent( reader )
    if( event.eq.0 ) goto 11

    runnum = levtgetrunnumber( event )
    evtnum = levtgeteventnumber( event )
    detname = levtgetdetectorname( event )

    write(*,*) " run: ", runnum
    write(*,*) " evt: ", evtnum
    write(*,*) " det: ", detname

10 continue
11 continue
c ---- end event loop ----
ana.job.f
```

plus additional methods in f77 for user convenience, e.g. HEPEvt <-> LCIO conversion

ECFA Workshop, Montpellier - Nov 14, 2003 Frank Gaede, DESY -IT- 11



LCIO Status

- production version 1.0 released:
 - C++, Java, f77 complete for simulation data
 - and generator data (HEPEvt<->LCIO)
 - simple example code for all languages
 - 'real world' examples (JAS3, root, hbook)
 - documentation
 - users manual
 - API documentation HTML (javadoc, doxygen)
 - available for download via CVS
 - linux (gcc), windows (cygwin)
- schema evolution from now on (reading old files)
- API stable (only extensions)



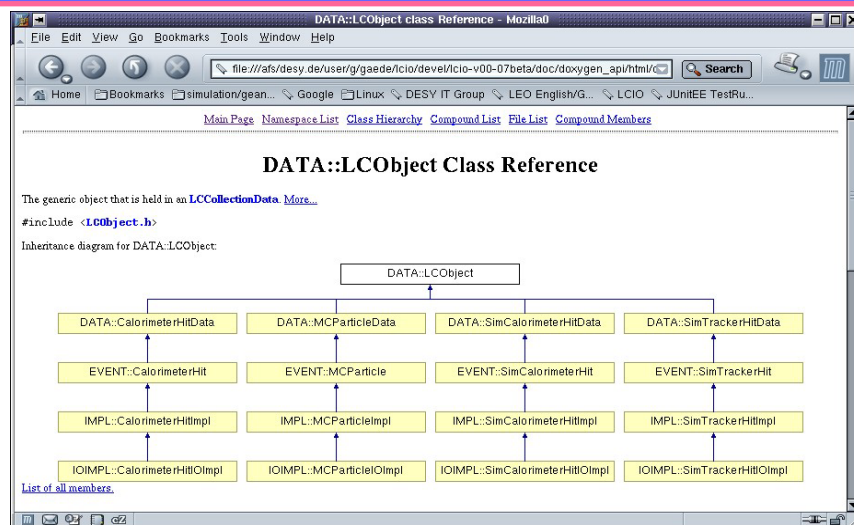
Javadoc example

The screenshot shows a Microsoft Internet Explorer browser window displaying the Javadoc for the `LCIReader` class. The page title is "LCIReader (LCIO API Documentation, Version v00-04) - Microsoft Internet Explorer". The address bar shows the URL: `http://www-k.desy.de/physics/projects/simsoft/lcio/api_java_v00-04/html/lcio/LCIReader.html`. The main content is a "Method Summary" table with the following entries:

Method	Description
<code>close ()</code>	Closes the output file/stream etc.
<code>open (String filename)</code>	Opens a file for reading (read-only)
<code>LCEvent readEvent (int runNumber, int evtNumber)</code>	Reads the specified event from file.
<code>LCEvent readNextEvent ()</code>	Reads the next event from the file.
<code>LCEvent readNextEvent (int accessMode)</code>	Same as above allowing to set the access mode (LCIO_READ_ONLY is default)
<code>LCRunHeader readNextRunHeader ()</code>	Reads the next run header from the file.
<code>readStream ()</code>	Reads the input stream and notifies registered listeners according to the object type found in the stream.
<code>void registerEventListener (LCEventListener l)</code>	Registers a listener for reading LCEvents from a stream.
<code>void registerCRunListener (LCRunListener l)</code>	Registers a listener for reading LCEvents/LCRunHeaders from a stream.
<code>void removeEventListener (LCEventListener l)</code>	Remove a listener for reading LCEvents from a stream.
<code>void removeCRunListener (LCRunListener l)</code>	Remove a listener for reading LCRunHeaders from a stream.



Doxygen example





LCIO Customers/Users

- Mokka simulation (v08-00)
 - update to v01-00 within ~1 week
- Brahms reconstruction
 - under development (H.Vogt)
-> to be release in ~2 weeks
- JAS3
 - provides convenient file browser
 - will have LCIO-WIRED plugin
-> generic event display !
- Calorimeter group (DESY)
 - will convert MiniCal raw data to LCIO files
 - to be used also for HCalPPT
- TPC groups (DESY & LBNL?)
 - will use LCIO for prototype
- other groups looking into using LCIO



JAS3 – LCIO file browser

N	Type	Status	Parent	PX	PY	PZ	Mass
0	2212	Document...	0	0	0	7000.0	0.93827
1	2212	Document...	0	0	0	-7000.0	0.93827
2	21	Document...	0	0.25815	-0.27900	6.5793	0
3	-3	Document...	1	-0.45454	-0.36117	-1802.7	0
4	4	Document...	2	-0.40964	-1.0530	2.2164	0
5	-3	Document...	3	-13.179	1.9646	-717.51	0
6	22	Document...	4,5	0.78672	0.69178	-4.4768	0
7	24	Document...	4,5	-14.375	0.21979	-710.81	80.667
8	22	Final State	6	0.78672	0.69178	-4.4768	0
9	24	Intermediate	7	-14.375	0.21979	-710.81	80.667
10	3224	Intermediate	1	0.16978	0.20640	-1483.5	1.3846
11	-4	Intermediate	2	1.0287	0.84333	2.4188	1.3500
12	2	Intermediate	0	0.080131	0.087964	0.31987	5.6000E-3
13	-3	Intermediate	9	-11.920	16.413	-260.20	0.19900
14	21	Intermediate	9	-9.7052	16.270	-246.29	0
15	21	Intermediate	9	-0.18941	-0.12814	-6.3494	0
16	21	Intermediate	9	-0.47022	-0.21941	-2.9564	0
17	21	Intermediate	9	0.41252	0.36534	-2.3612	0
18	21	Intermediate	9	-0.11239	-0.075933	0.055171	0
19	21	Intermediate	9	1.3372	-4.4404	-32.038	0
20	4	Intermediate	9	6.2717	-27.965	-160.67	1.3500
21	2	Intermediate	2	-3.5848	-3.3256	730.00	0
22	-2	Intermediate	3	3.5848	3.3256	-35.384	0
23	1	Intermediate	2	-2.7119	2.7973	2.4939	0

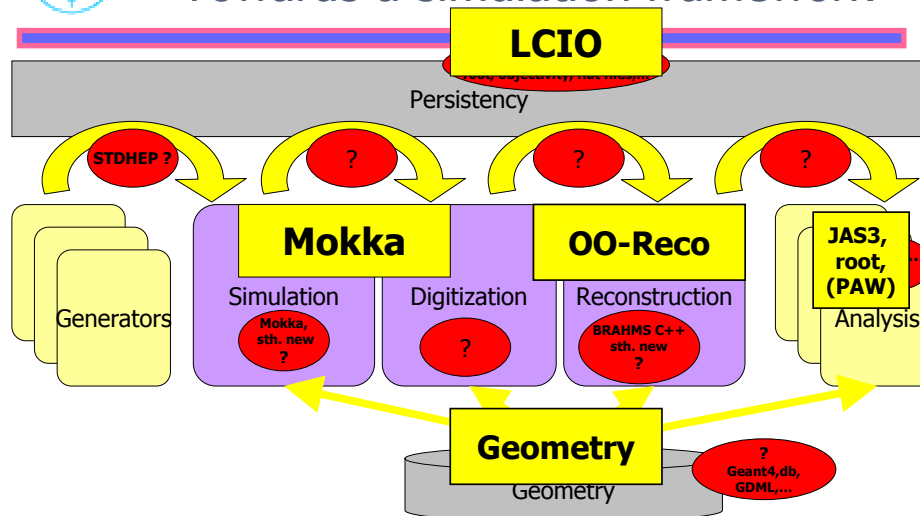


Future developments

- implement reconstruction data model
- add convenient methods
 - looping over MCParticles
 - analyzing parent/daughter relationships
 - ...
- add possibility to store more generic user data (calibration constants etc.)
- respond to user requests



Towards a simulation framework





Summary

- LCIO, a persistency framework for the LC:
 - Java, C++ and f77 user interface
 - Java and C++ implementation
 - data model for simulation and prototype data
 - reconstruction soon to follow
- production version released (1.0)
- used by several groups and tools
 - others invited to join !
- see LCIO homepage for more:

<http://www-it.desy.de/physics/projects/simsoft/lcio/index.html>